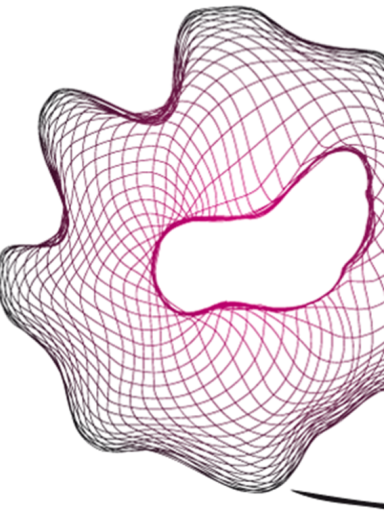
UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**



An FPGA based sensor fusion algorithm for IMU data processing

Pietro Pennestrì

MSc Embedded Systems
February 2022

Supervisors:
prof. dr. ir. Paul Havinga
dr. ir. Yanqiu Huang
dr.ir. N. Alachiotis

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

An FPGA based sensor fusion algorithm for IMU data processing

ABSTRACT

This study was motivated by the development of a smart bike accessory to evaluate onboard, according to internationally recognized standards, vibrational comfort and road condition using accelerometer sensor data. Since the norms require gravity filtered accelerometer data, it is necessary to compute the orientation of the Inertial Measurement Unit (IMU) platform.

Thus, the thesis theoretical focus was to estimate a body attitude using a 9-d.o.f IMU and provide an FPGA architecture for its implementation. As a novelty, it is herein proposed an improvement of the Madgwick algorithm, both in accuracy and speed. The first amelioration is achieved by adding a necessary condition between quaternions and their time derivatives in the computational flow. Since the algorithm requires many vectors normalization to compute the square root (SQRT) and inverse square root (ISQRT) of floating-point numbers, a customized architecture is proposed. This architecture implements a novel polynomial approximation based on the Chebyshev min-max criterion. A comparison of the performance of our SQRT/ISQRT floating-point IP core with vendor-provided IP cores confirms a smaller resource utilization and nearly four times better performance (smaller clock-cycle latency).

Quelli che s'innamorano di pratica, sanza scienza, son come 'l nocchiere, ch'entra in navilio sanza timone o bussola, che mai ha certezza dove si vada. Sempre la pratica dev'esser edificata sopra la bona teorica. La scienza è il capitano, e la pratica sono i soldati. Studia prima la scienza, e poi seguita la pratica nata da essa scienza.

Those who fall in love with practice without science are like the helmsman, who enters a ship without rudder or compass, who is never sure where he is going. Practice must always be built upon the good theory. Science is the captain, and practice is the soldier. Study science first, and then practice the science born of it.

Leonardo da Vinci (1452-1519),
Trattato della Pittura

ACKNOWLEDGEMENTS

I am thankful to Prof. Paul Havinga for proposing a thesis within the extensive research area of the Smart Bike Project. The topic was broadly defined, and autonomy of choice allowed. I was uncertain initially, but I am now really grateful for the absence of constraints. The task and responsibility to find a specific research path are challenging but, at the same time, greatly motivate the spirit of creativity and disclose a wealth of learning opportunities.

I wish to acknowledge the expert guidance and feedback of Dr Yanqiu Huang. These have been fundamental to outline the main features, functions and hardware selection of the device developed in the thesis. Furthermore, her support and encouragement were instrumental in exploring the theory of `IMU` sensor data fusion, an essential tool for the development of Internet of Things multi-sensor based intelligent devices.

My thanks also go to Dr Nikolaos Alachiotis for accepting to be a member of the Examination Committee and for contributions and counselling during the write up of the paper *A novel approximation scheme for floating-pointsquare root and inverse square root for FPGAs*.

Ai miei genitori per l'amore ed il supporto in ogni occasione.

Contents

Acknowledgements **4**

List of figures **iv**

List of tables **vii**

Acronyms **ix**

1 Introduction **1**
1.1 Background . . . 1
1.2 Motivation . . . 2
1.3 Problem Statement and Scientific Advancement . . . 2
1.4 Thesis organization . . . 3

2 Theoretical Background **5**
2.1 Sensor fusion theoretical background . . . 5
2.1.1 Review of spatial kinematics . . . 5
2.1.2 The Euler's theorem . . . 6
2.1.3 The Rodrigues' formula . . . 6
2.1.4 The Euler angles . . . 9
2.1.5 Angular speed and Euler angles time derivatives . . . 13
2.1.6 The Euler parameters . . . 14
2.1.7 Euler parameters consistency conditions . . . 14
2.1.8 Angular speed and coordinate transform time derivative . . . 16
2.1.9 Properties of Euler parameters time derivatives . . . 17
2.1.10 Quaternions . . . 19
2.2 Approximation Theory . . . 21
2.2.1 The min-max optimality criterion of Chebyshev . . . 22

3 Proposed Methods **25**
3.1 Classic Sensor Fusion Algorithms for IMU Data in Matrix Notation . . . 25
3.1.1 Summary of Mahony Algorithm . . . 25
3.1.2 Summary of Madgwick Algorithm . . . 26
3.2 Proposed Madgwick Algorithm Refinement . . . 27
3.3 Proposed Computation Scheme Features for ISQRT and SQRT . . . 29
3.3.1 Error analysis . . . 30

3.4 Polynomial approximation of $\frac{1}{\sqrt{x}}$ . . . 33
3.4.1 Proposed linear approximation of $\frac{1}{\sqrt{x}}$ . . . 33
3.4.2 Proposed quadratic approximations of $\frac{1}{\sqrt{x}}$ . . . 38
3.4.3 Proposed cubic approximation of $\frac{1}{\sqrt{x}}$ . . . 60
3.5 Polynomial approximation of $\sqrt{x}$ . . . 61
3.5.1 Proposed linear approximation of $\sqrt{x}$ . . . 61
3.5.2 Proposed quadratic approximation of $\sqrt{x}$ . . . 62
3.6 Summary of polynomial approximation accuracy for both ISQRT and SQRT . . . 70

4 Simulation Results and FPGA Implementation **73**
4.1 Sensor Fusion for IMU data: Numerical Tests . . . 73
4.2 FPGA implementation of the modified Madgwick sensor fusion algorithm . . . 73
4.2.1 Synthesis results . . . 81
4.3 FPGA Implementation of the proposed approximation for ISQRT and SQRT . . . 81
4.3.1 Exponent Computation . . . 86
4.3.2 Mantissa Computation . . . 87
4.3.3 Error Generation . . . 88
4.3.4 System Testing . . . 88
4.3.5 Synthesis Result . . . 97

5 Applications **101**
5.1 The Smart Bike Concept . . . 102
5.2 V-AUDIT: a vibration monitoring device . . . 104
5.2.1 Functional Features of V-AUDIT . . . 104
5.3 The Social Usefulness & Impact of V-AUDIT . . . 105
5.3.1 The Rider Point of View . . . 105
5.3.2 The Employer Point of View . . . 106
5.3.3 The City Administrator Point of View . . . 107
5.3.4 The Medical Point of View . . . 108
5.4 Bikers Comfort & Vibration Monitoring . . . 109
5.4.1 Literature review . . . 109
5.4.2 Whole Body Vibration: The `ISO 2631` Norm . . . 112
5.4.3 Hand-Arm Vibration: The `ISO 5349` Norm . . . 116
5.4.4 A Preliminary design of Digital Weighting Filters . . . 117
5.5 Road conditions assessment . . . 120
5.5.1 Previous studies . . . 120
5.5.2 International Roughness Index - `IRI` . . . 122
5.5.3 Proposed model for the computation of road roughness . . . 123
5.5.4 Numerical Validation . . . 125
5.6 Prototype Hardware Components . . . 125
5.6.1 System Requirements . . . 126
5.6.2 IMU Placement . . . 127

List of Figures

2.1 Proof of Rodrigues formula: Nomenclature . . . 7
2.2 The zxz Euler angles convention . . . 10
2.3 Rotation about z_1 . . . 11
2.4 Rotation about x_1 . . . 12
2.5 Rotation about z_2 . . . 13
2.6 W.R. Hamilton (1805-1865) . . . 19
2.7 Plaque on the Broom Bridge where Hamilton first wrote down the fundamental formula for quaternions multiplication . . . 19

3.1 `IEEE754` notation . . . 30
3.2 Overall computation scheme of the `Twente ISQRT SQRT` architecture . . . 31
3.3 Plot of error function $\varepsilon(x)$ . . . 34
3.4 Percent error $\varepsilon_{\%}(x)$ (see 3.22b) . . . 35
3.5 Error function (see (3.24a)) . . . 36
3.6 Error function (see (3.24b)) . . . 37
3.7 Comparison between $\varepsilon^{m2}_{\text{abs}}(x)$ and $\varepsilon^{m0}_{\text{abs}}(x)$ . . . 38
3.8 Comparison between $\varepsilon^{m2}_{\text{rel}}$ and $\varepsilon^{m0}_{\text{rel}}$ . . . 39
3.9 Comparison of relative error between Moroz *et al.* [57] and the proposed approximation $P_1^{\text{m1}}(x)$ (see Eq. (3.23)) . . . 40
3.10 Table from Parrilla et al. [58] where n is the number of bits addressing the table, and m is the number of bits representing the mantissa of the seed value. . . . 41
3.11 Error comparison between the magic constants proposed by Hasnat et al. [42] and the linearisation herein proposed in 3.27 . . . 42
3.12 Error function $\varepsilon(x)$ (see (3.31a)) . . . 43
3.13 Percent error $\varepsilon_{\%}$ (see (3.31b)) . . . 44
3.14 Error function $\varepsilon(x)$ (see (3.31a)) and its piecewise linearisation $\varepsilon_{\text{lin}}(x)$ (see (3.34a)) . . . 45
3.15 Error function $\varepsilon^{\text{m1}}(x)$ (see (3.34a)) . . . 46
3.16 Percent error for 2nd degree polynomial approximating $\dfrac{1}{\sqrt{x}}$ after first refinement . . . 47
3.17 Error functions, $\widetilde{\varepsilon}^{\,\text{m1}}_{[1,x_1]}(x)$, $\widetilde{\varepsilon}^{\,\text{m1}}_{(x_2,2)}(x)$ and $\varepsilon^{\text{m1b}}(x)$ . . . 49
3.18 . . . 50
3.19 Approximation domain division into 2 parts . . . 51
3.20 Error associate g_{int2temp} approximation . . . 52
3.21 Approximation error . . . 54
3.22 Plot of $\varepsilon^{\text{m2}}(x)$ (see (3.50b)) . . . 55

3.23 Plot of $\varepsilon_{\%}^{\mathrm{m2}}(x)$ (see (3.50c)) . . . 55
3.24 Plot of $\varepsilon^{\mathrm{m3}}(x)$ (see (3.53a)) . . . 57
3.25 Plot of $\varepsilon_{\%}^{\mathrm{m3}}(x)$ (see (3.53b)) . . . 58
3.26 Approximation domain ranges . . . 58
3.27 Plot of $\varepsilon^{\mathrm{m4}}(x)$ (see (3.55a)) . . . 59
3.28 Plot of $\varepsilon_{\%}^{\mathrm{m4}}(x)$ (see (3.55b)) . . . 60
3.29 Error for 3rd degree polynomial approximating $\frac{1}{\sqrt{x}}$ . . . 62
3.30 Percent error for 3rd degree polynomial approximating $\frac{1}{\sqrt{x}}$ . . . 63
3.31 Plot of $\varepsilon(x)$ (see (3.61a)) . . . 64
3.32 Plot of $\varepsilon_{\%}(x)$ (see (3.61b)) . . . 64
3.33 Plot of $\varepsilon^{\mathrm{m1}}(x)$ (see (3.63a)) . . . 65
3.34 Plot of $\varepsilon_{\%}^{\mathrm{m1}}(x)$ (see (3.63b)) . . . 65
3.35 Plot of $\varepsilon^{\mathrm{m0}}(x)$ (see (3.66a)) . . . 66
3.36 Plot of $\varepsilon_{\%}^{\mathrm{m0}}(x)$ (see (3.66b)) . . . 66
3.37 Plot of $\varepsilon^{\mathrm{m2}}(x)$ (see (3.68a)) . . . 67
3.38 Plot of $\varepsilon_{\%}^{\mathrm{m2}}(x)$ (see (3.68b)) . . . 68
3.39 Plot of $\varepsilon^{\mathrm{m3}}(x)$ (see (3.70a)) . . . 69
3.40 Plot of $\varepsilon_{\%}^{\mathrm{m3}}(x)$ (see (3.70b)) . . . 69
3.41 Plot of $\varepsilon^{\mathrm{m4}}(x)$ (see (3.72a)) . . . 70
3.42 Plots of $\varepsilon_{\%}^{\mathrm{m4}}(x)$ (see (3.72b)) . . . 71

4.1 IMU Data . . . 75
4.2 Error analysis for the Madgwick algorithm . . . 76
4.3 Error analysis for the Mahony algorithm . . . 77
4.4 Error analysis for the proposed algorithm (4 *refinement steps*) . . . 78
4.5 Sensor Fusion FPGA architecture . . . 79
4.6 Pinout of the Sensor Fusion block . . . 80
4.7 Sensor fusion simulation results . . . 82
4.8 Finite state Machine of the `constants_reg` component . . . 83
4.9 Finite state Machine of the `MatrixA` component . . . 84
4.10 Finite state Machine of the `quaternion_matrices` component . . . 85
4.11 Pinout of the proposed `Twente ISQRT & SQRT` architecture . . . 86
4.12 Proposed simplified schematic of the mantissa computation block (`polyroot` architecture). For clarity, some details have been omitted. . . . 89
4.13 ASM chart of the proposed control scheme . . . 91
4.14 Testing Phases of the `Twente ISQRT SQRT` . . . 92
4.15 Result of mantissa error analysis . . . 93
4.16 Result of mantissa error analysis (relative error). . . . 94
4.17 Result of mantissa error analysis (percent error). . . . 95
4.18 System error analysis for the square root and inverse square root cases . . . 96
4.19 System Architecture . . . 98
4.20 Exponent Architecture . . . 99

5.1 Gravity removal from accelerometer data for their processing according to international `ISO` norms . . . 102

5.2 The Smart Bike Concept . . . 103
5.3 A rider delivery worker . . . 105
5.4 Cyclist after a fall . . . 106
5.5 Potholes on a Public Road . . . 108
5.6 Cyclist with back pain . . . 108
5.7 Hand-Arm Vibration Syndrome . . . 109
5.8 Test on a paved street and OROS OR 36 acquisition system [35] . . . 110
5.9 Schematics of the seat shaker test bench, in the configuration with the human subject [37] 111
5.10 (a) Bicycle treadmill with two dowels attached to the belt; (b) Instrumented brake hood [38] 111
5.11 Databike of Taylor *et al.* [45] . . . 112
5.12 Experimental laboratory setup scheme of Doria *et al.* [49] . . . 113
5.13 Vibration Dose Value for Whole-Body Vibration assessment: $k_x = k_y = 1.4$, $k_z = 1$ (ISO 2631) . . . 114
5.14 ISO 2631 Cartesian axis orientation [18] . . . 114
5.15 Proposed Simulink model of the VDV considering accelerometer data along z axis only . . . 115
5.16 Vibration Dose Value Matlab computation. . . . 128
5.17 Computation Flow of the Vibration Total Value a_{hv}, ISO 5349 . . . 129
5.18 ISO 5349 Cartesian axis orientation [25] . . . 129
5.19 Weighting filter W_d . . . 130
5.20 Weighting filter W_k . . . 131
5.21 Qian and Dietmeier [43]: Positioning of accelerometers for measuring bike vertical displacements . . . 131
5.22 Installation of the smartphone to collect acceleration data according to Zang *et al.* [47] . . . 132
5.23 Measuring accelerations with smartphone for road roughness analysis [59] . . . 132
5.24 Example of roughness profile (adapted from [19]) . . . 133
5.25 The quarter-car model . . . 133
5.26 Road Profile Computation: Proposed dynamic model of the bicycle and biker . . . 134
5.27 Simulated Mass Acceleration . . . 134
5.28 Road Profile Comparison . . . 135
5.29 Qualitative comparison between ASIC, DSP, Microcontroller and FPGA . . . 135
5.30 FPGA Development Board Ecosystem . . . 136
5.31 Debug with logic analyzer . . . 136
5.32 Installation position of the IMU sensors . . . 137
5.33 IMU Shell . . . 138

List of Tables

2.1 Substitutions required to express Rodrigues' formula from vector to matrix notation 8
2.2 Coordinate transform, Angular speed and quaternion derivatives: Equivalence between quaternion and matrix expressions [22, 24] 21

3.1 Proposed method: Computation of maximum relative error (MRE) within the range [1 , 2), n: Polynomial degree , N: number of sub-intervals of $\log_2(N)$ amplitude 72

4.1 Comparison between AHRS algorithms 74
4.2 Resource Utilization 81
4.3 Max. Frequency 81
4.4 Scheduling of the polynomial mantissa evaluation 88
4.5 Resource comparison with the ALTFP_INV_SQRT IP 97
4.6 Max. Frequency 97

5.1 Vibration limits established by EU Directive 2002/44/EC [60] 107
5.2 Transfer function parameters of the weighting filters 119

Acronyms

AHRS Attitude and Heading Reference Systems.

ASIC Application Specific Integrated Circuit.

ASTM Americal Society Testing and Materials.

DSP Digital Signal Processor.

FPGA Field Programmable Gate Array.

I2C Inter-Integrated Circuit.

IMU Inertial Measurement Unit.

IoT Internet of Things.

IQL Information Quality Level.

ISO International Organization for Standardization.

ISQRT Inverse square root.

MARG Magnetic, Angular Rate, and Gravity sensor arrays.

MCU Microcontroller Unit.

PMS Pavement Management Systems.

RMS Root Mean Square.

SOPC System On a Programmable Chip.

SQRT Square root.

VDV Vibration Dose Value.

WBV Whole-Body Vibration.

Chapter 1

Introduction

Dimidium facti, qui coepit, habet.
Half is done when the beginning is done.

Quintus Horatius Flaccus

1.1 Background

We are all well aware that the need for low carbon emissions, imposed by legislation in many countries, requires a sustainable mobility based on a new energy-efficient, sensorized and connected class of vehicles. To meet the ambitious goal of the European Green Deal, it is estimated that, within the period 2021-2027, the European Regional Development Fund will allocate more than 1 billion euros for bicycle-related projects [1]. The digital systems installed on board of vehicles oversee already many important tasks such as assisted driving, safety devices, media entertainment, global positioning, wireless transmissions, energy consumption and powertrain control. Within 2030 it is forecasted that electronic systems will account for half of the total price of a new car[2]. This modern digitalization trend can be already observed in the new models of bikes available on the market. In these early phases, the digitalization of bikes is mainly limited to electric motor power control and to aftermarket accessories. However, replicating the trend observed for the automotive market, the development and installation on bikes of novel sensor-based electronic devices offer new possibilities. In fact, the growing trend is witnessed by the wide use in commercial advertisement of the term *Smart Bike*. A clear definition of such term cannot be found. Therefore, for my use, I coined the following tentative definition:

> *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.*

The early phases of my thesis work have been devoted to developing a novel low-cost device to collect and onboard process different sensors data. The great variety of sensors allows plenty of choices and

[1] https://www.ecf.com/funds-cycling/unlocking-cycle-investments/cycling-new-eu-budget-2021-2027-mff

[2] https://www.statista.com/statistics/277931/automotive-electronics-cost-as-a-share-of-total-car-cost-worldwide/

possibilities among the functions that the device may perform. A website dedicated to high performance bikes[3] was a source of inspiration. In particular, the vibrational comfort of different sport bike models was tested and measured by processing accelerometer sensors data according to ISO 2631 norm. It was striking that a norm issued to protect workers' health from the exposure to vibrations, was being used to compare sport bikes performance. Moreover, the ISO 2631 norm is officially recognized in the legislation of all European Union countries and embodied in the EU directive 2002/44/ES.

1.2 Motivation

The assessment of the bikers vibration using, for whole-body vibration ISO 2631 and ISO 5349 for hand-harm vibration, is an actively pursued research topic by many investigators. However, all the experimental tests were carried out in laboratories or, for limited time intervals, on the road with bulky and heavy collection data devices. All the data collected were post-processed and not immediately made available to the rider. That inspired the idea of a device able to continuously collect and process data from IMU sensors permanently installed on the bike. For our purposes, although limited in sampling frequency and accuracy, the 9 DoF IMU sensor based on MEMS offered a low-cost solution.

The website of the European Cyclists' Federation[4] was another source of inspiration. Many projects aim to develop and adopt the newest technology to capture large amounts of cycling data. Often smartphones apps combine GPS and accelerometer sensors data gathered on cyclist smartphone for different purposes. Recent scientific papers exploit the use of such data also for road conditions assessment.

To ensure data consistency and comparability, the device herein developed should process accelerometer sensors output according to recognized international norms, namely the ISO 2631, ISO 5349, for human vibrations, and the IRI (International Roughness Index), for road condition assessment. However, all the cited norms are based on the processing of gravity filtered acceleration data. This task, requires **the estimation of bike orientation** to transform the acceleration Cartesian components from the sensor to world system of reference.
These needs are shared by the bikers community and scientists. Prof. Chris Oliver (Royal College Surgeons of Edinburgh), in a recent interview at The Herald [48], called for:

> *A HIGH-TECH bicycle which measures cyclists' exposure to potentially harmful vibrations from uneven road surfaces could be used to check the safety of cycle lanes and other routes.*

1.3 Problem Statement and Scientific Advancement

The focus of the thesis is to develop an FPGA core to estimate, with a customized sensor fusion algorithm, the spatial orientation of an IMU platform. The thesis also discusses applying such development toward designing an innovative system to monitor the vibration dose value absorbed by riders (ISO 2631 and ISO 5349) and asses road conditions (IRI).The IMU spatial orientation is necessary to remove gravity from accelerometer data, as required by the mentioned applications.
The thesis offers the following novel contributions:

- The Madgwick [30, 32] and Mahony [23] sensor fusion algorithms have been improved. The quaternion notation is replaced with a matrix notation to reduce the number of operations scheduled in

[3] https://cyclingtips.com/2021/02/rough-ride-whole-body-vibrations-might-be-worse-for-you-than-you-think/

[4] https://www.ecf.com/

FPGA. Moreover, the accuracy was refined by introducing the norm E_m from a consistency condition of Euler parameters and their time derivatives. Finally, at least one iteration is added to minimize the norm.

- An FPGA architecture for the *modified* Madgwick algorithm is proposed.
- Both $\sqrt{x}$ and $\frac{1}{\sqrt{x}}$ are operations frequently used by sensor fusion algorithms. Because of the high latency of the `Altera ALTFP_INV_SQRT` IP core, a new algorithm to carry out such operations on shared architecture is proposed. This has led to an improvement in both resource usage and latency. The architecture was synthesized onto FPGA.

1.4 Thesis organization

The order of the chapters does not reflect the timeline of my work. The theoretical parts are ante-posed to the application description. In particular, the work is organized as follows:

- The second chapter deals with all the theoretical background of the novelty methods proposed, namely the modified Madgwick algorithm and the polynomial approximation of ISQRT and SQRT, based on Chebyshev min-max optimality criterion.
- The third chapter deals with the details of the proposed methods.
- The fourth chapter reports the numerical simulation results and FPGA architectures.
- The last chapter outlines some applications to the sensor fusion algorithm. In particular, it is described an onboard vibration monitoring device based on international standards to be adopted for the purpose of biker vibrational safety monitoring and road conditions assessment.

Chapter 2

Theoretical Background

Felix qui potuit rerum cognoscere causas.
Blessed is he who has been able to learn the causes.

Publius Vergilius Maro

2.1 Sensor fusion theoretical background

In this section some important spatial kinematic relationships, necessary as a theoretical background to fully understand the reasoning about the modification introduced in AHRS filters, will be deduced. In particular, the initial focus will be on how to express the orientation of a moving Cartesian system of coordinates (*i.e.* the one attached to the sensor) with respect to a fixed one (*i.e.* the world coordinate system).

There are different methods for expressing the spatial rotation or the orientation of two different coordinate systems. Many of these methods are reported by Rooney [8], Shuster [14] and in a NASA report [7].

For the properties about Euler parameters, the main bibiographical source is Nikravesh [10, 12].

The textbook of Kuipers [21] gives a modern introduction about quaternions. However, a classic introduction to quaternions is due to Klein [2]. The most detailed reference about the correspondence of kinematic equations both quaternion and matrix operations is the one of Schwab & Meijaard [24].

2.1.1 Review of spatial kinematics

Consider two right-hand orthonormal Cartesian reference systems with a common origin O:

- $O-XYZ$, fixed in the inertial system w;
- $O-xyz$, fixed in the moving body s.

The attitude or spatial orientation of a moving rigid body can be specified by locating the Cartesian reference fixed in the rigid body. It will be shown that the body orientation may be described in terms of a direction cosines matrix.

In particular, the attitude of the body is defined by a coordinate transform matrix ${}^{w}_{s}\mathbf{A}$ which maps the components of a vector ${}^{s}\mathbf{v}$, with components expressed in s, into those of a vector ${}^{w}\mathbf{v}$, expressed into w [8, 14]:

$$ {}^{w}\mathbf{v} = {}^{w}_{s}\mathbf{A}\,{}^{s}\mathbf{v} \tag{2.1} $$

2.1.2 The Euler's theorem

The original statement of the Euler's theorem is as follows[1]:

> *When a sphere is moved around its centre, it is always possible to find a diameter whose direction in the displaced position is the same as in the initial position.*

In other words, given two Cartesian reference systems, with the origin in common, their axes can be overlapped with a single finite rotation about an axis $\mathbf{u}$ (*finite rotation axis*). The points on $\mathbf{u}$ do not change their positions.

Since the body motion is rigid

$$ \|\mathbf{Ar}\| = \left\|\mathbf{r}'\right\| = \|\mathbf{r}\| \tag{2.2} $$

where

- $\mathbf{A}$ is a coordinate tranform or rotation matrix;
- $\mathbf{r}$ and $\mathbf{r}'$ are the initial and final positions of a vector attached to the moving body, respectively.

By definition of eigenvalue

$$ \mathbf{Ar} = \lambda\mathbf{r} \tag{2.3} $$

$$ \|\mathbf{Ar}\| = |\lambda|\,\|\mathbf{r}\| \tag{2.4} $$

The comparison of Eqs. (2.2) and (2.4) yelds

$$ \lambda = 1 \tag{2.5} $$

Therefore, it exists a vector $\mathbf{u}$ such that

$$ \mathbf{Au} = \mathbf{u} \tag{2.6} $$

where $\mathbf{u}$ is the unit vector (eigenvector of $\mathbf{A}$) representing the rotation axis.

2.1.3 The Rodrigues' formula

Let us consider the rotation of vector $\mathbf{r} = \overrightarrow{OR}$ of an angle θ around axis $\mathbf{u}$. The new position of the vector is $\mathbf{r}' = \overrightarrow{OR}_1$.

With reference to Figure 2.1, a plane orthogonal to $\mathbf{u}$ containing the points P, R and R_1 is introduced, where P is the intersection of such a plane with $\mathbf{u}$. From the geometry of Figure 2.1 follows:

$$ \mathbf{r}' = \overrightarrow{OP} + \overrightarrow{PQ} + \overrightarrow{QR}_1 $$

[1] L. Euler, *Novi Commentarii Academiae Scientiarum Petropolitanae* **20**, 1776, pp. 189-207 (E478), `http://vmm.math.uci.edu/euler/interlinear.htm`

> *Quomodocunque sphaera circa centrum suum conuertatur, semper assignari potest diameter, cuius directio in situ translato conueniat cum situ initiali.*

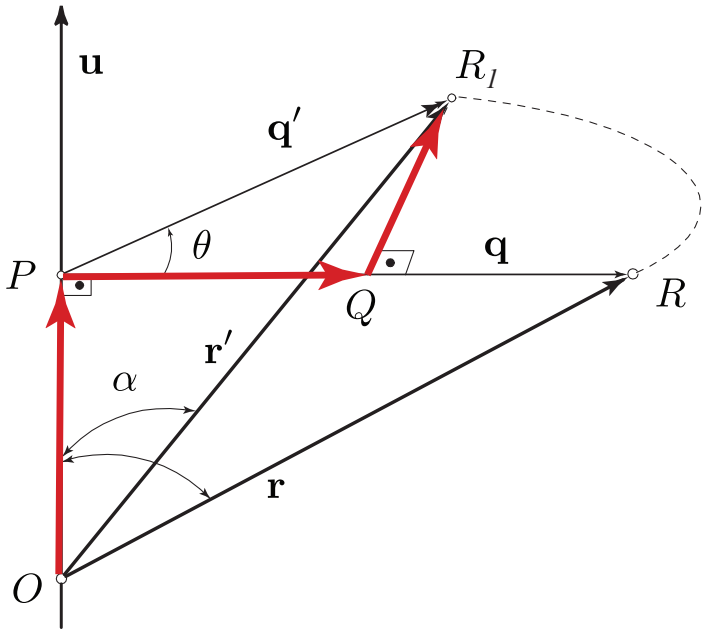


Figure 2.1: Proof of Rodrigues formula: Nomenclature

where Q is the foot of perpendicular of R_1 to PR.

We observe that the $\mathbf{r}'$ can be interpreted as the sum of three orthogonal vectors:

$$\mathbf{r}' = m_1\mathbf{u} + m_2\mathbf{q} + m_3\mathbf{u}\times\mathbf{q}\,, \tag{2.7}$$

where the coefficients m_1, m_2 ed m_3 are unknowns,

$$\mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^\intercal$$

is the unit vector of rotation axis and θ the angle of rotation.

The unknown coefficients are deduced as follows:

- Coefficient m_1

  The scalar product of $\mathbf{u}$ for both members of (2.7) yields

$$m_1 = \mathbf{r}'\cdot\mathbf{u} = \mathbf{r}\cdot\mathbf{u} \tag{2.8}$$

- Coefficient m_2

  The scalar product of $\mathbf{q}$ for both members of (2.7) yields

$$\mathbf{r}'\cdot\mathbf{q} = m_2\|\mathbf{q}\|^2 \tag{2.9}$$

  After substitution of

$$\mathbf{r}' = \overrightarrow{OP} + \overrightarrow{PR_1} = \left(\mathbf{r}'\cdot\mathbf{u}\right)\mathbf{u} + \mathbf{q}' \tag{2.10}$$

Table 2.1: Substitutions required to express Rodrigues' formula from vector to matrix notation

Vector notation	Matrix notation
$(\mathbf{r}\cdot\mathbf{u})\,\mathbf{u}$	$(\mathbf{u}^{\mathsf{T}}\mathbf{r})\,\mathbf{u} = (\mathbf{u}\mathbf{u}^{\mathsf{T}})\,\mathbf{r}$
$[\mathbf{r}-(\mathbf{r}\cdot\mathbf{u})\,\mathbf{u}]$	$[\mathbf{I}-\mathbf{u}\mathbf{u}^{\mathsf{T}}]\,\mathbf{r}$
$(\mathbf{u}\times\mathbf{r})$	$\widetilde{\mathbf{U}}\mathbf{r}$

[a] The matrix

$$\widetilde{\mathbf{U}} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$$

is the skew-symmetric matrix associated to $\mathbf{u}$ and $\mathbf{I}$ the identity matrix.

into (2.9) one obtains

$$\mathbf{r}'\cdot\mathbf{q} = \cancel{(\mathbf{r}'\cdot\mathbf{u})\,\mathbf{u}\cdot\mathbf{q}} + \mathbf{q}'\cdot\mathbf{q} = m_2\|\mathbf{q}\|^2$$

or

$$m_2 = \frac{\|\mathbf{q}'\|^2}{\|\mathbf{q}\|^2}\cos\theta = \cos\theta \tag{2.11}$$

- Coefficient m_3

  The scalar product of $\mathbf{u}\times\mathbf{q}$ for both members of (2.7) yields

  $$\mathbf{r}'\cdot(\mathbf{u}\times\mathbf{q}) = m_3\,(\mathbf{u}\times\mathbf{q})\cdot(\mathbf{u}\times\mathbf{q}) = m_3\mathbf{q}\|^2 \tag{2.12}$$

  After substitution of (2.10) into the previous equation one obtains

  $$\mathbf{q}'\cdot(\mathbf{u}\times\mathbf{q}) = m_3\mathbf{q}\|^2$$

  or

  $$\mathbf{q}'\cdot\mathbf{q} = \mathbf{q}'\|\mathbf{q}\|\cos\left(\frac{\pi}{2}-\theta\right) = m_3\mathbf{q}\|^2$$

  $$m_3 = \sin\theta \tag{2.13}$$

Since

$$\mathbf{q} = \overrightarrow{OR} - \overrightarrow{OP} = \mathbf{r} - \mathbf{r}\cdot(\mathbf{r}\cdot\mathbf{u})\cdot\mathbf{u} \tag{2.14}$$

the substitution of (2.8), (2.11), eqrefeq:m3 into (2.7), gives the **Rodrigues' formula** in vector notation:

$$\mathbf{r}' = \cos\theta\mathbf{r} + (1-\cos\theta)\,(\mathbf{r}\cdot\mathbf{u})\,\mathbf{u} + \sin\theta\mathbf{u}\times\mathbf{r} \tag{2.15}$$

The Rodrigues' formula can be expressed in matrix notation to obtain a general form of rotation matrix $\mathbf{R}(\mathbf{u},\theta)$ such that

$$\mathbf{r}' = \mathbf{R}\mathbf{r} \tag{2.16}$$

From equation (2.15), taking into account the entries of the Table 2.1, follows the general form of the rotation matrix:

$$\mathbf{R} = \mathbf{I}\cos\theta + (1-\cos\theta)\mathbf{u}\mathbf{u}^{\mathsf{T}} + \widetilde{\mathbf{U}}\sin\theta \tag{2.17}$$

where

$$\widetilde{\mathbf{U}} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \tag{2.18}$$

The rotation matrix $\mathbf{R}$ coincides with the coordinate transform matrix ${}^{w}_{s}\mathbf{A}$ from the moving reference system s to the fixed reference system w:

$$\mathbf{R} = {}^{w}_{s}\mathbf{A}$$

After expanding (2.17), the transform matrix can be expressed as follows:

$${}^{w}_{s}\mathbf{A} = \begin{bmatrix} u_x^2 V_\theta + c_\theta & u_x u_y V_\theta - u_z s_\theta & u_x u_z V_\theta + u_y s_\theta \\ u_x u_y V_\theta + u_z s_\theta & u_y^2 V_\theta + c_\theta & u_y u_z V_\theta - u_x s_\theta \\ u_x u_z V_\theta - u_y s_\theta & u_y u_z V_\theta + u_x s_\theta & u_z^2 V_\theta + c_\theta \end{bmatrix} \tag{2.19}$$

where $V_\theta = 1-\cos\theta$, $s_\theta = \sin\theta$, $c_\theta = \cos\theta$.

This is the **most general case of coordinate transform matrix**.

Its elements depend on **three independent parameters** (namely the Euler angles).

- The columns of ${}^{w}_{s}\mathbf{A}$ represent the unit vectors of the moving reference axes of the moving system s, expressed in the reference system w (direction cosines matrix).
- Since ${}^{w}_{s}\mathbf{A}$ is orthonormal:
  - $\det\left({}^{w}_{s}\mathbf{A}\right) = +1$
  - the transpose coincides with its inverse, the following equality holds:

$${}^{s}_{w}\mathbf{A} = {}^{w}_{s}\mathbf{A}^{\mathsf{T}} \tag{2.20}$$

  - $\lambda = 1$ must be one of its eigenvalues.

This last property can be used to prove the Euler's theorem.

2.1.4 The Euler angles

Consider two positions A and B of the moving body Cartesian reference systems. In the first one the axes of moving and inertial Cartesian reference frames are aligned. The Euler angles are the angles of rotation, imposed to the moving reference around its own axes, required to move it from position A to position B.

There are three axes so there should be $3^3 = 27$ sequences of Euler angles. However, there is no point in rotating around the same axis twice in succession because the same result can be obtained by rotating once by the sum of the angles, so there are only $3 \cdot 2 \cdot 2 = 12$ different Euler angle sequences [7]:

xyx	xyz	xzx	xzy
yxy	yxz	yzx	yzy
zxy	zxz	zyx	zyz

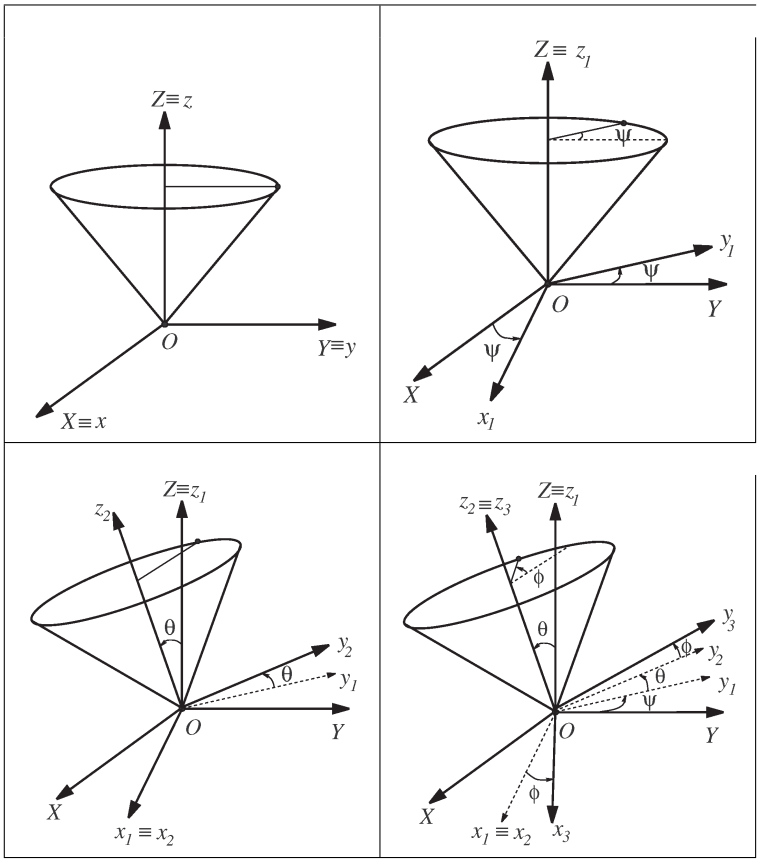


Figure 2.2: The zxz Euler angles convention

In this section the so called Euler angles zxz convention, shown in Figure 2.2, will be discussed.

- Rotation about z_1 (see Figure 2.3).

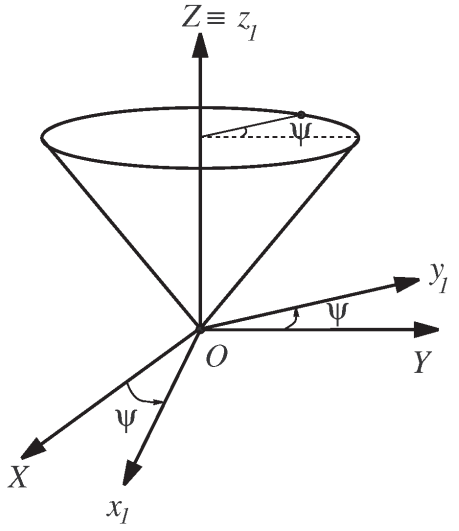


Figure 2.3: Rotation about z_1

${}^1_0\mathbf{A}$ is the **components transform matrix** from Cartesian system $O-XYZ$ to system $o-x_1y_1z_1$:

$$ {}^1_0\mathbf{A} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ \text{ù} & -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.21} $$

- Rotation about x_1 (see Figure 2.4).

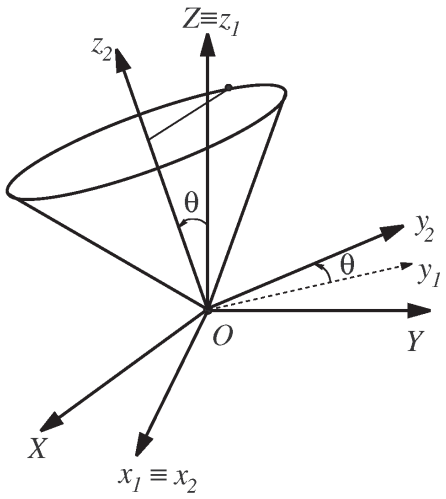


Figure 2.4: Rotation about x_1

${}^2_1\mathbf{A}$ is the **components transform matrix** from Cartesian system $O-x_1y_1z_1$ to system $o-x_2y_2z_2$:

$$ {}^2_1\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{2.22} $$

${}^2_1\mathbf{A}$ is the **components transform matrix** from Cartesian system $O-x_2y_2z_2$ to system $o-x_3y_3z_3$:

- Rotation about z_2 (see Figure 2.4).

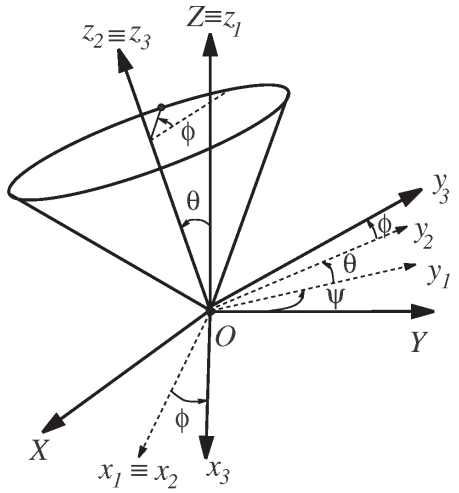


Figure 2.5: Rotation about z_2

$$ {}^{3}_{2}\mathbf{A} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ \grave{u} & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.23} $$

Hence, the transform matrix from fixed (**world**, $O-XYZ$) to moving local (**sensor**, $o_3 - x_3 y_3 z_3$) reference frame is:

$$ {}^{s}_{w}\mathbf{A} = {}^{3}_{2}\mathbf{A}\,{}^{2}_{1}\mathbf{A}\,{}^{1}_{0}\mathbf{A} = \begin{bmatrix} c_\phi c_\psi - s_\phi c_\theta s_\psi & c_\phi s_\psi + s_\phi c_\theta c_\psi & s_\phi s_\theta \\ -s_\phi c_\psi - c_\phi c_\theta s_\psi & -s_\phi s_\psi + c_\phi c_\theta c_\psi & c_\phi s_\theta \\ s_\theta s_\psi & -s_\theta c_\psi & c_\theta \end{bmatrix} \tag{2.24} $$

$c_\phi = \cos\phi$, $s_\phi = \sin\phi$,...,

2.1.5 Angular speed and Euler angles time derivatives

The angular speed of the resultant motion is obtained adding the angular speeds of all the previous three motions. Therefore, the angular speed with components expressed in the sensor frame $s \to 3$ is:

$$ {}^{3}\boldsymbol{\omega} = {}^{3}_{2}\mathbf{A}\,{}^{2}_{1}\mathbf{A} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + {}^{3}_{2}\mathbf{A} \begin{bmatrix} \dot{\theta} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\phi} \end{bmatrix} \tag{2.25} $$

The expansion of the previous sum gives the body angular speed with components expressed in the local system as function of Euler angles derivatives:

$$ {}^{s}\boldsymbol{\omega} = \begin{bmatrix} \sin\theta\cos\phi & \cos\phi & 0 \\ \sin\theta\cos\phi & -\sin\phi & 0 \\ \cos\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} \tag{2.26} $$

2.1.6 The Euler parameters

One of the most meaningful method for representing the orientation of a moving system is based on the use of **quaternion components** or **Euler parameters**.

$$\mathbf{p} = \begin{bmatrix} e_0 & e_1 & e_2 & e_3 \end{bmatrix}^\mathsf{T}$$

where

$$e_0 = \cos\frac{\theta}{2} \tag{2.27a}$$

$$e_1 = u_x \sin\frac{\theta}{2} \tag{2.27b}$$

$$e_2 = u_y \sin\frac{\theta}{2} \tag{2.27c}$$

$$e_3 = u_z \sin\frac{\theta}{2} \tag{2.27d}$$

The substitution of Euler parameters into transform matrix (2.19) yields:

$${}^w_s\mathbf{A} = 2\begin{bmatrix} e_0^2 + e_1^2 - \frac{1}{2} & e_1e_2 - e_0e_3 & e_1e_3 + e_0e_2 \\ e_1e_2 + e_0e_3 & e_0^2 + e_2^2 - \frac{1}{2} & e_2e_3 - e_0e_1 \\ e_1e_3 - e_0e_2 & e_2e_3 + e_0e_1 & e_0^2 + e_3^2 - \frac{1}{2} \end{bmatrix} \tag{2.28}$$

Introduced the matrices

$$\mathbf{E} = \begin{bmatrix} -e_1 & e_0 & -e_3 & e_2 \\ -e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{bmatrix} \tag{2.29}$$

$$\mathbf{G} = \begin{bmatrix} -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{bmatrix} \tag{2.30}$$

the matrix transform can be factored as follows:

$${}^w_s\mathbf{A} = \mathbf{E}\mathbf{G}^\mathsf{T} \tag{2.31}$$

2.1.7 Euler parameters consistency conditions

If we introduce the vectors [10, 12]

$$\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^\mathsf{T} \tag{2.32}$$

$$\mathbf{p} = \begin{bmatrix} e_0 \\ \mathbf{e} \end{bmatrix} \tag{2.33}$$

then the matrices can be expressed in a partitioned form

$$\mathbf{E} = \left[\begin{array}{c|c} -\mathbf{e} & \widetilde{\mathbf{e}} + e_0\mathbf{I} \end{array}\right] \tag{2.34}$$

$$\mathbf{G} = \left[\begin{array}{c|c} -\mathbf{e} & -\widetilde{\mathbf{e}} + e_0\mathbf{I} \end{array}\right] \tag{2.35}$$

where

$$\widetilde{\mathbf{e}} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{2.36}$$

is the antisymmetric matrix associated to vector $\mathbf{e}$ and $\mathbf{I}$ the unit matrix.

Taken into account the equalities:

$$\widetilde{\mathbf{e}}\mathbf{e} = \mathbf{0} \tag{2.37}$$

$$e_0^2 = 1 - \mathbf{e}^\top\mathbf{e} \tag{2.38}$$

$$\begin{aligned} \widetilde{\mathbf{e}}\,\widetilde{\mathbf{e}} &= \mathbf{e}\mathbf{e}^\top - \left(\mathbf{e}\mathbf{e}^\top\right)\mathbf{I} \\ &= \mathbf{e}\mathbf{e}^\top - \left(1 - e_0^2\right)\mathbf{I} \end{aligned} \tag{2.39}$$

The following properties can be verified:

$$\begin{aligned} \mathbf{E}\mathbf{p} &= \left[\begin{array}{c|c} -\mathbf{e} & \widetilde{\mathbf{e}} + e_0\mathbf{I} \end{array}\right] \begin{bmatrix} e_0 \\ \mathbf{e} \end{bmatrix} \\ &= -e_0\mathbf{e} + \widetilde{\mathbf{e}}\mathbf{e} + e_0\mathbf{e} = \mathbf{0} \end{aligned} \tag{2.40a}$$

or

$$\mathbf{E}\mathbf{p} = \mathbf{0} \tag{2.40b}$$

Similarly:

$$\mathbf{G}\mathbf{p} = \mathbf{0} \tag{2.40c}$$

$$\mathbf{E}\mathbf{E}^\top = \mathbf{G}\mathbf{G}^\top = \mathbf{I} \tag{2.40d}$$

$$\mathbf{E}^\top\mathbf{E} = \mathbf{G}^\top\mathbf{G} = \mathbf{I} - \mathbf{e}\mathbf{e}^\top \tag{2.40e}$$

Since

$$\mathbf{p}^\top\mathbf{p} = 1 \tag{2.41}$$

then[2]

$$\dot{\mathbf{p}}^\top\mathbf{p} = \mathbf{p}^\top\dot{\mathbf{p}} = 0 \tag{2.42}$$

In fact

$$\begin{aligned} &\dot{\mathbf{p}}^\top\mathbf{p} = \\ &= \frac{1}{2}\frac{\mathrm{d}\theta}{\mathrm{d}t}\left[-\sin\frac{\theta}{2}\;\; u_x\cos\frac{\theta}{2}\;\; u_y\cos\frac{\theta}{2}\;\; u_z\cos\frac{\theta}{2}\right]\left[\cos\frac{\theta}{2}\;\; u_x\sin\frac{\theta}{2}\;\; u_y\sin\frac{\theta}{2}\;\; u_z\sin\frac{\theta}{2}\right]^\top \\ &= \frac{1}{2}\frac{\mathrm{d}\theta}{\mathrm{d}t}\left(-\cos\frac{\theta}{2}\sin\frac{\theta}{2} + \cos\frac{\theta}{2}\sin\frac{\theta}{2}\right) = 0 \end{aligned}$$

[2] This is the consistency condition between Euler parameters and their derivatives missing from the Madgwick's sensor fusion algorithm. The herein proposed algorithm modification includes it in the computational flow aimed to improve the sensor orientation accuracy.

From $\mathbf{Ep} = \mathbf{0}$ and $\mathbf{Gp} = \mathbf{0}$, after time differentiation, follow, respectively:

$$\mathbf{E}\dot{\mathbf{p}} = -\dot{\mathbf{E}}\mathbf{p} \tag{2.43}$$

$$\mathbf{G}\dot{\mathbf{p}} = -\dot{\mathbf{G}}\mathbf{p} \tag{2.44}$$

2.1.8 Angular speed and coordinate transform time derivative

The general transform equation

$$^{w}\mathbf{s} = {}^{w}_{s}\mathbf{A}\,{}^{s}\mathbf{s} \tag{2.45}$$

can be differentiated w.r.t. time

$$^{w}\dot{\mathbf{s}} = {}^{w}_{s}\dot{\mathbf{A}}\,{}^{s}\mathbf{s} + {}^{w}_{s}\mathbf{A}\,{}^{s}\dot{\mathbf{s}} \tag{2.46}$$

For a rigid body motion ${}^{s}\dot{\mathbf{s}} = 0$ and the previous one reduces to

$$^{w}\dot{\mathbf{s}} = {}^{w}_{s}\dot{\mathbf{A}}\,{}^{s}\mathbf{s} \tag{2.47}$$

The Poisson equation is

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{s} = \vec{\omega} \times \vec{s} \tag{2.48}$$

or, in matrix notation

$$^{w}\dot{\mathbf{s}} = {}^{w}\widetilde{\boldsymbol{\omega}}\,{}^{w}\mathbf{s} \tag{2.49}$$

The comparison of the previous one with

$$^{w}\dot{\mathbf{s}} = {}^{w}_{s}\dot{\mathbf{A}}\,{}^{s}\mathbf{s} = {}^{w}_{s}\dot{\mathbf{A}}\,{}^{s}_{w}\mathbf{A}\,{}^{w}\mathbf{s} \tag{2.50}$$

yields

$$^{w}\widetilde{\boldsymbol{\omega}} = {}^{w}_{s}\dot{\mathbf{A}}\,{}^{s}_{w}\mathbf{A} = {}^{w}_{s}\dot{\mathbf{A}}\,{}^{w}_{s}\mathbf{A}^{\mathsf{T}} \tag{2.51}$$

In conclusion, the following relationship holds:

$$^{w}\widetilde{\boldsymbol{\omega}} = {}^{w}_{s}\dot{\mathbf{A}}\,{}^{w}_{s}\mathbf{A}^{\mathsf{T}} \tag{2.52}$$

In a similar manner, one can demonstrate that:

$$^{s}\widetilde{\boldsymbol{\omega}} = {}^{w}_{s}\mathbf{A}^{\mathsf{T}}\,{}^{w}_{s}\dot{\mathbf{A}} \tag{2.53}$$

To extract ${}^{w}_{s}\dot{\mathbf{A}}$, one can post-multiply both terms by ${}^{w}_{s}\mathbf{A}$

$$^{w}\widetilde{\boldsymbol{\omega}}\,{}^{w}_{s}\mathbf{A} = {}^{w}_{s}\dot{\mathbf{A}}\cancel{{}^{w}_{s}\mathbf{A}^{\mathsf{T}}\,{}^{w}_{s}\mathbf{A}} \tag{2.54}$$

or

$$^{w}_{s}\dot{\mathbf{A}} = {}^{w}\widetilde{\boldsymbol{\omega}}\,{}^{w}_{s}\mathbf{A} \tag{2.55}$$

2.1.9 Properties of Euler parameters time derivatives

In the following, we will demonstrate two important equalities regarding Euler parameters and their derivatives:

$$\widetilde{\mathbf{E}\dot{\mathbf{p}}} = -\mathbf{E}\dot{\mathbf{E}}^{\mathsf{T}} \tag{2.56}$$

$$\widetilde{\mathbf{G}\dot{\mathbf{p}}} = -\mathbf{G}\dot{\mathbf{G}}^{\mathsf{T}} \tag{2.57}$$

For this purpose, the following general matrix identities must be taken into account:

$$\widetilde{\widetilde{\mathbf{a}}\mathbf{b}} = \mathbf{b}\mathbf{a}^{\mathsf{T}} - \mathbf{a}\mathbf{b}^{\mathsf{T}} = \tilde{\mathbf{a}}\,\tilde{\mathbf{b}} - \tilde{\mathbf{b}}\,\tilde{\mathbf{a}} \tag{2.58}$$

$$\tilde{\mathbf{a}}\,\tilde{\mathbf{b}} = \mathbf{b}\mathbf{a}^{\mathsf{T}} - \mathbf{a}\mathbf{b}^{\mathsf{T}} \tag{2.59}$$

$$\begin{aligned}
\widetilde{\mathbf{E}\dot{\mathbf{p}}} &= -\dot{e}_0\tilde{\mathbf{e}} + \widetilde{\widetilde{\dot{\mathbf{e}}}} + e_0\tilde{\dot{\mathbf{e}}} \\
&= -\dot{e}_0\tilde{\mathbf{e}} + \tilde{\mathbf{e}}\,\tilde{\dot{\mathbf{e}}} - \tilde{\dot{\mathbf{e}}}\tilde{\mathbf{e}} + e_0\tilde{\dot{\mathbf{e}}} \\
&= -\dot{e}_0\tilde{\mathbf{e}} + \tilde{\mathbf{e}}\,\tilde{\dot{\mathbf{e}}} - \mathbf{e}\dot{\mathbf{e}}^{\mathsf{T}} + \left(\dot{\mathbf{e}}^{\mathsf{T}}\mathbf{e}\right)\mathbf{I} + e_0\tilde{\dot{\mathbf{e}}} \\
&= -\dot{e}_0\tilde{\mathbf{e}} + \tilde{\mathbf{e}}\,\tilde{\dot{\mathbf{e}}} - \mathbf{e}\dot{\mathbf{e}}^{\mathsf{T}} - e_0\dot{e}_0\mathbf{I} + e_0\tilde{\dot{\mathbf{e}}} \\
&= \left[\begin{array}{c|c} \mathbf{e} & -\tilde{\mathbf{e}} - e_0\mathbf{I} \end{array}\right] \left[\begin{array}{c} -\dot{\mathbf{e}}^{\mathsf{T}} \\ -\tilde{\dot{\mathbf{e}}} + \dot{e}_0\mathbf{I} \end{array}\right] \\
&= -\mathbf{E}\dot{\mathbf{E}}^{\mathsf{T}}
\end{aligned} \tag{2.60}$$

$$\boxed{\widetilde{\mathbf{E}\dot{\mathbf{p}}} = -\mathbf{E}\dot{\mathbf{E}}^{\mathsf{T}}} \tag{2.61}$$

$$\begin{aligned}
\widetilde{\mathbf{G}\dot{\mathbf{p}}} &= -\dot{e}_0\tilde{\mathbf{e}} - \widetilde{\widetilde{\dot{\mathbf{e}}}} + e_0\tilde{\dot{\mathbf{e}}} \\
&= -\dot{e}_0\tilde{\mathbf{e}} - \tilde{\mathbf{e}}\,\tilde{\dot{\mathbf{e}}} + \tilde{\dot{\mathbf{e}}}\tilde{\mathbf{e}} + e_0\tilde{\dot{\mathbf{e}}} \\
&= -\dot{e}_0\tilde{\mathbf{e}} - \tilde{\mathbf{e}}\,\tilde{\dot{\mathbf{e}}} + \mathbf{e}\dot{\mathbf{e}}^{\mathsf{T}} - \left(\dot{\mathbf{e}}^{\mathsf{T}}\mathbf{e}\right)\mathbf{I} + e_0\tilde{\dot{\mathbf{e}}} \\
&= -\dot{e}_0\tilde{\mathbf{e}} - \tilde{\mathbf{e}}\,\tilde{\dot{\mathbf{e}}} + \mathbf{e}\dot{\mathbf{e}}^{\mathsf{T}} + e_0\dot{e}_0\mathbf{I} + e_0\tilde{\dot{\mathbf{e}}} \\
&= \left[\begin{array}{c|c} -\mathbf{e} & -\tilde{\mathbf{e}} + e_0\mathbf{I} \end{array}\right] \left[\begin{array}{c} -\dot{\mathbf{e}}^{\mathsf{T}} \\ \tilde{\dot{\mathbf{e}}} + \dot{e}_0\mathbf{I} \end{array}\right] \\
&= \mathbf{G}\dot{\mathbf{G}}^{\mathsf{T}}
\end{aligned} \tag{2.62}$$

$$\boxed{\widetilde{\mathbf{G}\dot{\mathbf{p}}} = \mathbf{G}\dot{\mathbf{G}}^{\mathsf{T}}} \tag{2.63}$$

$$\begin{aligned}
\mathbf{E}\dot{\mathbf{G}}^{\mathsf{T}} &= \left[\begin{array}{c|c} -\mathbf{e} & \tilde{\mathbf{e}} + e_0\mathbf{I} \end{array}\right] \left[\begin{array}{c} -\dot{\mathbf{e}}^{\mathsf{T}} \\ \tilde{\dot{\mathbf{e}}} + \dot{e}_0\mathbf{I} \end{array}\right] \\
&= \mathbf{e}\dot{\mathbf{e}}^{\mathsf{T}} + \tilde{\mathbf{e}}\tilde{\dot{\mathbf{e}}} + e_0\tilde{\mathbf{e}} + e_0\dot{e}_0\mathbf{I} + e_0\tilde{\dot{\mathbf{e}}}
\end{aligned} \tag{2.64}$$

Moreover, considered that

$$\mathbf{e}\dot{\mathbf{e}}^{\mathsf{T}} = \dot{\mathbf{e}}^{\mathsf{T}}$$

$$\widetilde{\mathbf{e}}\,\widetilde{\dot{\mathbf{e}}} = \widetilde{\dot{\mathbf{e}}}\,\widetilde{\mathbf{e}}$$

one has

$$\dot{\mathbf{E}}\mathbf{G}^{\mathsf{T}} = \mathbf{e}\dot{\mathbf{e}}^{\mathsf{T}} + \widetilde{\mathbf{e}}\widetilde{\dot{\mathbf{e}}} + e_0\widetilde{\dot{\mathbf{e}}} + e_0\dot{e}_0\mathbf{I} + e_0\widetilde{\dot{\mathbf{e}}} \tag{2.65}$$

thus, after comparison, one can conclude:

$$\boxed{\mathbf{E}\dot{\mathbf{G}}^{\mathsf{T}} = \dot{\mathbf{E}}\mathbf{G}^{\mathsf{T}}} \tag{2.66}$$

It is convenient to express the angular speed in terms of Euler parameters time derivatives.
The time derivative of ${}^{w}_{s}\mathbf{A} = \mathbf{E}\mathbf{G}^{\mathsf{T}}$, taking into account (2.66), is

$${}^{w}_{s}\dot{\mathbf{A}} = \dot{\mathbf{E}}\mathbf{G}^{\mathsf{T}} + \mathbf{E}\dot{\mathbf{G}}^{\mathsf{T}} = 2\dot{\mathbf{E}}\mathbf{G}^{\mathsf{T}} = 2\mathbf{E}\dot{\mathbf{G}}^{\mathsf{T}} \tag{2.67}$$

We can substitute (2.67) into (2.52)

$${}^{w}\widetilde{\boldsymbol{\omega}} = 2\dot{\mathbf{E}}\mathbf{G}^{\mathsf{T}}\mathbf{G}\mathbf{E}^{\mathsf{T}} \tag{2.68}$$

or, taking into account (2.40e) and (2.40b),

$$\begin{aligned} {}^{w}\widetilde{\boldsymbol{\omega}} &= 2\dot{\mathbf{E}}\mathbf{E}^{\mathsf{T}} - 2\dot{\mathbf{E}}\mathbf{p}\mathbf{p}^{\mathsf{T}}\mathbf{E}^{\mathsf{T}} \\ &= 2\dot{\mathbf{E}}\mathbf{E}^{\mathsf{T}} \end{aligned} \tag{2.69}$$

Finally, taking into account (2.61) $\widetilde{\mathbf{E}\dot{\mathbf{p}}} = -\mathbf{E}\dot{\mathbf{E}}^{\mathsf{T}}$, the previous one can be rewritten in the form

$$\boxed{{}^{w}\boldsymbol{\omega} = 2\mathbf{E}\dot{\mathbf{p}}} \tag{2.70}$$

A similar equation can be obtained for the angular speed ${}^{s}\boldsymbol{\omega}$.
In fact, the substitution of (2.67) into (2.53) yelds:

$$\begin{aligned} {}^{s}\widetilde{\boldsymbol{\omega}} &= \mathbf{G}\mathbf{E}^{\mathsf{T}}\mathbf{E}\dot{\mathbf{G}}^{\mathsf{T}} \\ 2\mathbf{G}\left(\mathbf{I} - \mathbf{p}\mathbf{p}^{\mathsf{T}}\right)\dot{\mathbf{G}}^{\mathsf{T}} &= 2\mathbf{G}\dot{\mathbf{G}}^{\mathsf{T}} \end{aligned} \tag{2.71}$$

Taking into account (2.63), the previous one can be rewritten in the form

$$\boxed{{}^{s}\boldsymbol{\omega} = 2\mathbf{G}\dot{\mathbf{p}}} \tag{2.72}$$

Since $\mathbf{p}^{\mathsf{T}}\dot{\mathbf{p}} = e_0\dot{e}_0 + e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 = 0$, some sources report (2.70) and (2.72) as follows

$$\begin{bmatrix} 0 \\ {}^{w}\omega_x \\ {}^{w}\omega_y \\ {}^{w}\omega_z \end{bmatrix} = 2\begin{bmatrix} e_0 & e_1 & e_2 & e_3 \\ -e_1 & e_0 & -e_3 & e_2 \\ -e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{bmatrix}\begin{bmatrix} \dot{e}_0 \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} \tag{2.73}$$

$$\begin{bmatrix} 0 \\ {}^s\omega_x \\ {}^s\omega_y \\ {}^s\omega_z \end{bmatrix} = 2 \begin{bmatrix} e_0 & e_1 & e_2 & e_3 \\ -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{bmatrix} \begin{bmatrix} \dot{e}_0 \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} \tag{2.74}$$

The ${}^s\boldsymbol{\omega} = 2\mathbf{G}\dot{\mathbf{p}}$ can be rewritten in the form

$$\mathbf{G}^\top {}^s\boldsymbol{\omega} = 2\mathbf{G}^\top\mathbf{G}\dot{\mathbf{p}} \tag{2.75}$$

Since $\mathbf{G}^\top\mathbf{G} = \mathbf{I} - \mathbf{p}\mathbf{p}^\top$ and $\mathbf{p}^\top\dot{\mathbf{p}} = 0$, the previous formula can be expressed in the form:

$$\dot{\mathbf{p}} = \frac{1}{2}\mathbf{G}^\top {}^s\boldsymbol{\omega} \tag{2.76}$$

In a similar manner, one obtains:

$$\dot{\mathbf{p}} = \frac{1}{2}\mathbf{E}^\top {}^w\boldsymbol{\omega} \tag{2.77}$$

2.1.10 Quaternions

The quaternions, introduced by the Irish mathematician W.R. Hamilton, is a number system extending the complex numbers [2, 21].

Figure 2.6: W.R. Hamilton (1805-1865)

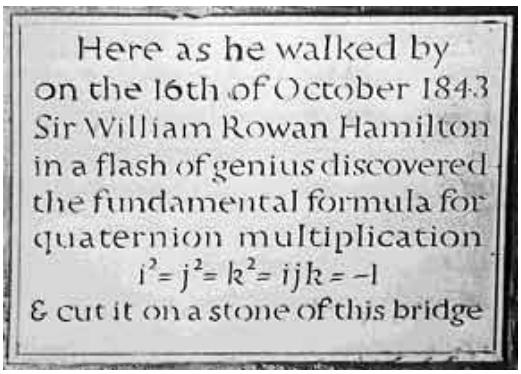


Figure 2.7: Plaque on the Broom Bridge where Hamilton first wrote down the fundamental formula for quaternions multiplication

A common way for expressing quaternions is by adding the real component with the three imaginary units:

$$\underline{\mathbf{e}} = \left(e_0 + e_1 \cdot i + e_2 \cdot j + e_3 \cdot k\right) \tag{2.78}$$

with $e_0, e_1, e_2, e_3 \in \mathbb{R}$ and the imaginary units having the following multiplication properties[3]:

$$\begin{aligned} i^2 &= -1 \\ j^2 &= -1 \\ k^2 &= -1 \\ i \cdot j \cdot k &= -1 \\ i \cdot j = -j \cdot i &= k \\ j \cdot k = -k \cdot j &= i \\ k \cdot i = -i \cdot k &= j \end{aligned}$$

Alternatively, a quaternion can be expressed distinguishing the real part e_0 and the vector part $\mathbf{e} = \begin{Bmatrix} e_1 & e_2 & e_3 \end{Bmatrix}^\mathsf{T}$ or

$$\underline{\mathbf{e}} = (e_0, \mathbf{e}) \tag{2.79}$$

Therefore, a vector $\mathbf{v}$ can be expressed as a quaternion $\underline{\mathbf{v}}$ with zero real part:

$$\underline{\mathbf{v}} = (0, \mathbf{v}) \tag{2.80}$$

In the following, the algebraic definition of the main quaternion operations is listed:

$$\underline{\mathbf{q}} = \left(q_0, \mathbf{q}\right) \tag{2.81}$$

- Sum

$$\underline{\mathbf{q}}_a \pm \underline{\mathbf{q}}_b = \left(q_{0a} \pm q_{0b} \pm \mathbf{q}_a \pm \mathbf{q}_b\right) \tag{2.82}$$

- Conjugate

$$\underline{\mathbf{q}}^* = \left(q_0, -\mathbf{q}\right) \tag{2.83}$$

- Quaternion product

$$\begin{aligned} \underline{\mathbf{q}}_b \otimes \underline{\mathbf{q}}_a = &\left(q_{0a}q_{0b} - q_{1a}q_{1b} - q_{2a}q_{2b} - q_{3a}q_{3b}\right) + \\ &+ \left(q_{0a}q_{1b} + q_{1a}q_{0b} + q_{2a}q_{3b} - q_{3a}q_{2b}\right) \cdot i + \\ &+ \left(q_{0a}q_{2b} - q_{1a}q_{3b} + q_{2a}q_{0a} + q_{3a}q_{1b}\right) \cdot j + \\ &+ \left(q_{0a}q_{3b} + q_{1a}q_{2b} - q_{2a}q_{1b} + q_{3a}q_{0b}\right) \cdot k \end{aligned} \tag{2.84}$$

- Module

$$\left|\underline{\mathbf{q}}\right| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{2.85}$$

- Division

$$\underline{\mathbf{q}}^{-1} = \underline{\mathbf{q}}^* \frac{1}{\left|\underline{\mathbf{q}}\right|^2} \tag{2.86}$$

[3] The Figure 2.7 represents the stone plaque placed to commemorate the place where Hamilton had the *eureka moment* about quaternion multiplication.

Table 2.2: Coordinate transform, Angular speed and quaternion derivatives: Equivalence between quaternion and matrix expressions [22, 24]

Quaternion notation	Matrix notation
${}^{s}\underline{\mathbf{v}} = {}_{w}^{s}\underline{\mathbf{q}} \otimes {}^{w}\underline{\mathbf{v}} \otimes {}_{w}^{s}\underline{\mathbf{q}}^{*}$	${}^{s}\mathbf{v} = {}_{w}^{s}\mathbf{A}\,{}^{w}\mathbf{v}$
${}^{s}\underline{\boldsymbol{\omega}} = 2\,{}_{w}^{s}\underline{\mathbf{q}}^{*} \otimes {}_{w}^{s}\dot{\underline{\mathbf{q}}}$	${}^{s}\boldsymbol{\omega} = 2\mathbf{G}\dot{\underline{\mathbf{q}}}$
${}^{w}\underline{\boldsymbol{\omega}} = 2\,{}_{w}^{s}\dot{\underline{\mathbf{q}}} \otimes {}_{w}^{s}\underline{\mathbf{q}}^{*}$	${}^{w}\boldsymbol{\omega} = 2\mathbf{E}\dot{\underline{\mathbf{q}}}$
${}_{w}^{s}\dot{\underline{\mathbf{q}}} = \frac{1}{2}\,{}^{w}\underline{\boldsymbol{\omega}} \otimes {}_{w}^{s}\underline{\mathbf{q}}$	$\dot{\underline{\mathbf{q}}} = \frac{1}{2}\mathbf{E}^{\top}\,{}^{w}\boldsymbol{\omega}$
${}_{w}^{s}\dot{\underline{\mathbf{q}}} = \frac{1}{2}\,{}_{w}^{s}\underline{\mathbf{q}} \otimes {}^{s}\underline{\boldsymbol{\omega}}$	$\dot{\underline{\mathbf{q}}} = \frac{1}{2}\mathbf{G}^{\top}\,{}^{s}\boldsymbol{\omega}$

The orientation described by quaternion ${}_{w}^{s}\underline{\mathbf{q}}$ can be also represented by the transform matrix ${}_{w}^{s}\mathbf{A}$.

The coordinate transform of the same vector, represented by ${}^{w}\mathbf{v}$ and ${}^{s}\mathbf{v}$ in two coordinate frame w and s, respectively, can be computed both in matrix and quaternion notation as follows:

- **Quaternion notation**

$$ {}^{s}\underline{\mathbf{v}} = {}_{w}^{s}\underline{\mathbf{q}} \otimes {}^{w}\underline{\mathbf{v}} \otimes {}_{w}^{s}\underline{\mathbf{q}}^{*} \tag{2.87} $$

- **Matrix notation**

$$ {}^{s}\mathbf{v} = {}_{w}^{s}\mathbf{A}\,{}^{w}\mathbf{v} \tag{2.88} $$

More generally, the entries of Table 2.2 summarize the equivalence between some relevant kinematic formulas expressed both in quaternion and matrix notation. These equivalences are fundamental for the translation of Madgwick algorithm into matrix notation.

2.2 Approximation Theory

The solution of many mathematical and technological problems rely upon the approximation of a real continuos function $f(x)$ with a simpler function $F(P,x)$, where P stands for a fixed finite number of parameters $p_0, p_1, \ldots, p_n$.

According to Rice [4], there is no scientific method of determining which of the many functions available lead to the *most* efficient approximation. For this reason, experience, intuition and type of application herein play a fundamental role.

Beside the choice of functions, another important item is the measure of approximation goodness. For this purpose, a distance function or *norm* $\varepsilon(x)$ needs to be introduced to measure the approximation.

In synthesis, the approximation problem can be stated as follows [4]:

Let $f(x)$ be a given real-valued continuous function defined on a set X, and let $F(p,x)$ be a real-valued approximating function, depending continuously on $x \in X$ and on n parameters P. Given the distance function ε, determine a set of parameters P^* such that

$$\varepsilon\left(F\left(P^*,x\right),f(x)\right) \leq \varepsilon\left(F\left(P,x\right),f(x)\right) \tag{2.89}$$

Once the problem is formulated, the following phases for its solution needs to be distinguished[4]:

1. The choice of approximating function and the distance function.
2. The existence of the solution.
3. The uniqueness of a solution.
4. The characteristic and other special properties of the solution.
5. The computation of the solution.

Considered the practical purpose of the present work, herein only the first and last items are of interest. In particular, our treatment will be limited to an application of the Chebyshev min-max norm toward the approximation of $\sqrt{x}$ and $\dfrac{1}{\sqrt{x}}$.

For a rigorous and extensive theoretical background and mathematical properties of Chebyshev theory of approximation, the book of Remez [3] is recommended. Other interesting bibliographical sources are those of Mayans [26], for an elementary proof of Chebyshev equioscillation theorem, and by Golub *et al.* [5, 6], for computer programming of min-max approximations.

2.2.1 The min-max optimality criterion of Chebyshev

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

$$F_n(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n \tag{2.90}$$

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

$$\varepsilon\left(x\right) = \max_{x_0 \leq x \leq x_{n+1}} \left|f(x) - F_n(x)\right| \tag{2.91}$$

It can be shown that $F_n(x)$ is the Chebyshev approximation of a differentiable[4] function $f(x)$ on $[x_0,x_{n+1}]$ if and only if there exists a set of points

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

[4] Recent proofs (*e.g.* [26]) of the Chebyshev equioscillation theorem require continuity only. However, the original theorem statement from Chebyshev requires also differentiability of $f(x)$. This is witnessed by the following excerpt from the original Chebyshev paper [1]:

> Soit $f(x)$ une fonction donnée, U un polynome du degré n avec des coefficients arbitraires. Si l'on choisit ces coefficients de manière à ce que la différence $f(x)-U$, depuis $x=a-h$, jusq'à $x=a+h$, reste dans les limites les plus rapprochées de 0, la différence $f(x)-U$ jouira, comme on le sait, de cette propriété:
>
> Parmi les valeurs le plus grandes et les plus petites de la différence $f(x)-U$ entre les limites $x=a-h$, $x=a+h$, on trouve au moins $n+2$ fois la même valeur numérique.
>
> Les valeurs que $f(x)-U$ prend pour $x=a-h$ and $x=a+h$ son considérées comme *maximum* ou *minimum*.
>
> D'après cela on trouve facilement les équations que les coefficients de U doivent vérifier. Si nous convenons de dénoter par L la valeur numérique commune des $n+2$ *maxima* ou *minima* de $f(x)-U$ qui doivent avoir lieu entre

such that:

$$\varepsilon(x_i) = (-1)^i L \qquad (i = 0, 1, \ldots, n+1) \tag{2.95a}$$

$$\left.\frac{\mathrm{d}\varepsilon(x)}{\mathrm{d}x}\right|_{x=x_j} = 0 \qquad (j = 1, 2, \ldots, n) \tag{2.95b}$$

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

These conditions ensure that the error function $\varepsilon(x)$, within the approximation interval, has $n+2$ maxima and minima with alternating sign, and n stationary points, respectively.

The previous $2n+2$ conditions (2.95) are those required by the so-called Chebyshev's equioscillation theorem [1, 3, 26]. In particular, the resulting equations form a non linear system whose solution yields: L, $x_1, x_2, \ldots, x_n$, p_0, p_1, …, p_n.

Thus, the solution of the non linear system provides the user not only with the coefficients of the optimal approximating polynomial, but also with the value L of maximum error and the abscissae where the maxima and minima of $\varepsilon(x)$ occur.

---

les limites $x = a - h$, $x = a + h$, l'équation

$$[f(x) - U]^2 - L^2 = 0 \tag{2.92}$$

doit avoir $n+2$ racines comprises entre $a-h$ et $a+h$, et toutes ces racines doivent vérifier l'équation

$$\frac{\mathrm{d}}{\mathrm{d}x}[f(x) - U] = 0 \tag{2.93}$$

qui est la condition du *maximum* et du *minimum*, ou bien se réduire aux valeurs $a-h$, $a+h$; en un mot, les $n+2$ racines de l'equation (2.92), comprises entre $a-h$, $a+h$ doivent vérifier celle-ci

$$(x - a + h)(x - a - h)\frac{\mathrm{d}}{\mathrm{d}x}[f(x) - U] = 0 \tag{2.94}$$

Cela nous donne un nombre suffisant d'équations pour trouver les $n+1$ coefficients du polynome U et la valeur inconnu L; car chaque des $n+2$ racines communes aux équations (2.92) et (2.94) suppose une équation entre les coefficients et la quantité L, ce qui fait en total $n+2$ équations.

Chapter 3

Proposed Methods

.... et per hoc difficilis etiam ipsius intellectus et doctrina, sed tanto amplius effectus et completio gloriosa. Propter quod cum subtiliori advertentia attende et considera que secuntur.
...the comprehension and theory are difficult, but mind blowing is the effect and execution. For this reason, study carefully the things that will be explained in the following.

G. de Dondi (*Tractatus Astrarii*)

The theory presented in the previous Chapter will be now used for the support and development of:

- the proposed modified version of the Madgwick algorithm;
- the theoretical study of ISQRT, SQRT polynomial approximations and the proposal of a novel computational approach.

3.1 Classic Sensor Fusion Algorithms for IMU Data in Matrix Notation

3.1.1 Summary of Mahony Algorithm

The main computational steps of Mahony sensor fusion algorithm [**mahony2005**, 27] are summarized as follows:

1. **Data:** ${}^{s}\mathbf{a}_{i+1}$, ${}^{s}\mathbf{h}_{i+1}$, ${}^{s}\boldsymbol{\omega}_{i+1}$, $\underline{\mathbf{q}}_{i}$
2. **Constants:** ${}^{w}\mathbf{g} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^{\mathsf{T}}$, ${}^{w}\mathbf{b} = \begin{bmatrix} b_x & 0 & b_y \end{bmatrix}^{\mathsf{T}}$, K_b, K_g, Δt
3. **Compute:** $\mathbf{G}$, $\mathbf{E}$, $\mathbf{A} = \mathbf{E}\mathbf{G}^{\mathsf{T}}$

4. **Normalize:** ${}^s\mathbf{a}_{i+1} \rightarrow \dfrac{{}^s\mathbf{a}_{i+1}}{\left\|{}^s\mathbf{a}_{i+1}\right\|}$, ${}^s\mathbf{h}_{i+1} \rightarrow \dfrac{{}^s\mathbf{h}_{i+1}}{\left\|{}^s\mathbf{h}_{i+1}\right\|}$

5. **Compute:** ${}^s\mathbf{v}_a = \mathbf{A}\,{}^w\mathbf{g}$, ${}^s\mathbf{v}_h = \mathbf{A}\,{}^w\mathbf{b}$

6. **Compute:** $\boldsymbol{\varepsilon} = {}^s\mathbf{v}_a \times {}^s\mathbf{a} + {}^s\mathbf{v}_h \times {}^s\mathbf{h}$ (Heading error)

7. **Compute:** $\boldsymbol{\omega}_b(t_{i+1}) = K_b\boldsymbol{\varepsilon}$, $\boldsymbol{\omega}_g(t_{i+1}) = K_g\boldsymbol{\varepsilon}\Delta t + \boldsymbol{\omega}_g(t_i)$

8. **Compute:** ${}^s\hat{\boldsymbol{\omega}}(t_{i+1}) = {}^s\boldsymbol{\omega}(t_{i+1}) + \boldsymbol{\omega}_b(t_{i+1}) + \boldsymbol{\omega}_g(t_{i+1})$ (Corrected gyroscope angular rate)

9. **Compute:** $\dot{\underline{\mathbf{q}}}_\omega(t_{i+1}) = \dfrac{1}{2}\mathbf{G}\,{}^s\hat{\boldsymbol{\omega}}(t_{i+1})$

10. $\underline{\mathbf{q}}(t_{i+1}) = \underline{\mathbf{q}}(t_i) + \dot{\underline{\mathbf{q}}}_\omega(t_{i+1})\Delta t$

11. **Normalize** $\underline{\mathbf{q}}(t_{i+1})$

3.1.2 Summary of Madgwick Algorithm

The main computational steps of Madgwick sensor fusion algorithm [30] are summarized as follows:

1. **Data:** ${}^s\mathbf{a}_{i+1}$, ${}^s\mathbf{h}_{i+1}$, ${}^s\boldsymbol{\omega}_{i+1}$, $\underline{\mathbf{q}}_i$

2. **Constants:** ${}^w\mathbf{g} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^\top$, ${}^w\mathbf{b} = \begin{bmatrix} b_x & 0 & b_y \end{bmatrix}^\top$, β, Δt

3. **Normalize:** ${}^s\mathbf{a}_{i+1} \leftarrow \dfrac{{}^s\mathbf{a}_{i+1}}{\left\|{}^s\mathbf{a}_{i+1}\right\|}$, ${}^s\mathbf{h}_{i+1} \leftarrow \dfrac{{}^s\mathbf{h}_{i+1}}{\left\|{}^s\mathbf{h}_{i+1}\right\|}$

4. **Compute:** $\mathbf{G}$, $\mathbf{E}$, $\mathbf{A} = \mathbf{E}\mathbf{G}^\top$

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

6. **Form:** $\mathbf{f} = \begin{bmatrix} \mathbf{f}_a \\ \mathbf{f}_h \end{bmatrix}$ and $\mathbf{J} = \begin{bmatrix} \mathbf{J}_a \\ \mathbf{J}_h \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \mathbf{f}_a}{\partial \mathbf{q}} \\ \dfrac{\partial \mathbf{f}_h}{\partial \mathbf{q}} \end{bmatrix}$, $\nabla\mathbf{f} = \mathbf{J}^\top\mathbf{f}$

7. **Compute:** $\dot{\underline{\mathbf{q}}}_{\omega_{i+1}} = \dfrac{1}{2}\mathbf{G}^\top\,{}^s\boldsymbol{\omega}_{i+1}$

8. **Compute:** $\dot{\underline{\mathbf{q}}}_{est_{i+1}} = \left(\dot{\underline{\mathbf{q}}}_{\omega_{i+1}} - \beta\dfrac{\nabla\mathbf{f}}{\|\nabla\mathbf{f}\|}\right)$

9. **Compute:** $\underline{\mathbf{q}}_{i+1} = \underline{\mathbf{q}}_i + \dot{\underline{\mathbf{q}}}_{est_{i+1}}\Delta t$

10. **Normalize** $\underline{\mathbf{q}}_{i+1}$

3.2 Proposed Madgwick Algorithm Refinement

In this section are described three different types of Madgwick algorithm improvements:

1. The computation of **A**, **f** and **J** elements have several common terms. Therefore, it is worthwhile to compute them once for all. This reduces the amount of algebraic operation to be scheduled into the target architecture.
2. The condition $\mathbf{f}_a = \mathbf{A}\,{}^w\mathbf{g} - {}^s\mathbf{a}_{i+1} = 0$ requires that the modules of ${}^s\mathbf{a}_{i+1}$ and $\mathbf{g}$ be the same. This condition is fulfilled only when the sensor has a slow motion through quasi-static or still positions. Therefore, in our implementation, the condition is included in the computation flow only when the module of $\mathbf{a}_{i+1}$ and $\mathbf{g}$ differ of a prescribed threshold value.
3. The accuracy is improved by including in the computation flow the consistency condition expressed by (3.1).

The solution obtained at *step* 10 of the Madgwick algorithm, can be further improved requiring the minimization of the norm:

$$E_m = \dot{\underline{\mathbf{q}}}^{\top}\underline{\mathbf{q}} \tag{3.1}$$

$$F = E_m^2 \tag{3.2}$$

At least one step iteration, based on the gradient descent method, gives the *refined* value of $\underline{\mathbf{q}}_{i+1}$:

$$\underline{\mathbf{q}}_{i+1}^{r} = \underline{\mathbf{q}}_{i+1} - 2\dot{\underline{\mathbf{q}}}_{i+1}\left(\dot{\underline{\mathbf{q}}}_{i+1}^{\top}\underline{\mathbf{q}}_{i+1}\right)^3 \tag{3.3}$$

where

$$\dot{\underline{\mathbf{q}}}_{i+1} = \frac{1}{2}\mathbf{G}^{\top}\,{}^s\boldsymbol{\omega}_{i+1} \tag{3.4}$$

and $\mathbf{G}$ is evaluated as function considering the elements of $\underline{\mathbf{q}}_{i+1}$. The described refinement can be also applied to the Mahony algorithm.

1. **Data:** ${}^s\mathbf{a}_{i+1}$, ${}^s\mathbf{h}_{i+1}$, ${}^s\boldsymbol{\omega}_{i+1}$, $\underline{\mathbf{q}}_i$
2. **Constants:**
   ${}^w\mathbf{g} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^{\top}$, β, Δt, a_{th}, n
3. **if** $\left|\left|{}^s\mathbf{a}_{i+1}\right|\right| \leq a_{th}$ **then** a =TRUE **else** a =FALSE
4. **Normalize:** ${}^s\mathbf{a}_{i+1} \leftarrow \dfrac{{}^s\mathbf{a}_{i+1}}{\left\|{}^s\mathbf{a}_{i+1}\right\|}$, ${}^s\mathbf{h}_{i+1} \leftarrow \dfrac{{}^s\mathbf{h}_{i+1}}{\left\|{}^s\mathbf{h}_{i+1}\right\|}$
5. **Form** matrices $\mathbf{G}, \mathbf{E}$ and $\mathbf{A} = \mathbf{E}\mathbf{G}^{\top}$
6. **Let**
   $${}^w\mathbf{h} = \mathbf{A}\,{}^s\mathbf{h}$$

7. **Let**

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

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

8. **Let**

$$^{w}\mathbf{b} = \begin{bmatrix} b0 & 0 & b1 \end{bmatrix}$$

9. **if** $a ==$ 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

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

10. **Let**

$$\mathbf{K} = \begin{bmatrix} -e_2 & e_3 & -e_0 & e_1 \\ e_1 & e_0 & e_3 & e_2 \\ 0 & -2e_1 & -2e_2 & 0 \end{bmatrix} \tag{3.5}$$

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & -2e_2 & -2e_3 \\ -e_3 & e_2 & e_1 & -e_0 \\ e_2 & e_3 & e_0 & e_1 \end{bmatrix} \tag{3.6}$$

11. **if** $a ==$ TRUE **then**

$$\mathbf{f}^{\mathsf{T}}\mathbf{J} = 2\left(\mathbf{f}_a^{\mathsf{T}} + b_1\mathbf{f}_h^{\mathsf{T}}\right)\mathbf{K} + 2b_0\mathbf{f}_a^{\mathsf{T}}\mathbf{L} \tag{3.7}$$

else

$$\mathbf{f}^{\mathsf{T}}\mathbf{J} = 2b_1\mathbf{f}_h^{\mathsf{T}}\mathbf{K} + 2b_0\mathbf{f}_a^{\mathsf{T}}\mathbf{L} \tag{3.8}$$

12. **Compute**

$$\Delta\underline{\mathbf{q}} = \frac{\mathbf{J}^{\mathsf{T}}\mathbf{f}}{||\mathbf{J}^{\mathsf{T}}\mathbf{f}||}$$

13. **Compute:** $\dot{\underline{\mathbf{q}}}_{\omega_{i+1}} = \frac{1}{2}\mathbf{G}^{\mathsf{T}}\,{}^{s}\boldsymbol{\omega}_{i+1}$

14. **Compute:** $\dot{\underline{\mathbf{q}}}_{est_{i+1}} = \left(\dot{\underline{\mathbf{q}}}_{\omega_{i+1}} - \beta\Delta\underline{\mathbf{q}}\right)$

15. **Compute:** $\underline{\mathbf{q}}_{i+1} = \underline{\mathbf{q}}_i + \dot{\underline{\mathbf{q}}}_{est_{i+1}}\Delta t$

16. **Normalize** $\underline{\mathbf{q}}_{i+1}$

17. E_m **Refinement Iteration Steps**

for $i = 1 : n$

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

3.3 Proposed Computation Scheme Features for ISQRT and SQRT

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` notation summarized in Figure 3.1 [11]:

$$x = (-1)^s 2^{e_u} \cdot m \tag{3.9}$$

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

- **isqrt even exponent**

$$\frac{1}{\sqrt{x}} = 2^{\frac{-e_u}{2}} \cdot \frac{1}{\sqrt{m}} \tag{3.10}$$

- **isqrt odd exponent**

$$\frac{1}{\sqrt{x}} = 2^{\frac{-e_u+1}{2}} \cdot \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{m}} \tag{3.11}$$

- **sqrt even exponent**

$$\sqrt{x} = 2^{\frac{e_u}{2}} \cdot \sqrt{m} \tag{3.12}$$

- **sqrt odd exponent**

$$\sqrt{x} = 2^{\frac{e_u-1}{2}} \cdot \sqrt{2} \cdot \sqrt{m} \tag{3.13}$$

On these premises, the challenging task is limited to the computation of either the inverse square root or the square root of the mantissa. Hence, the proposed computation scheme is aimed toward a polynomial approximation of the two previous operations in the range $[1\ ,\ 2)$. The overall computation scheme is depicted in Figure 3.2.

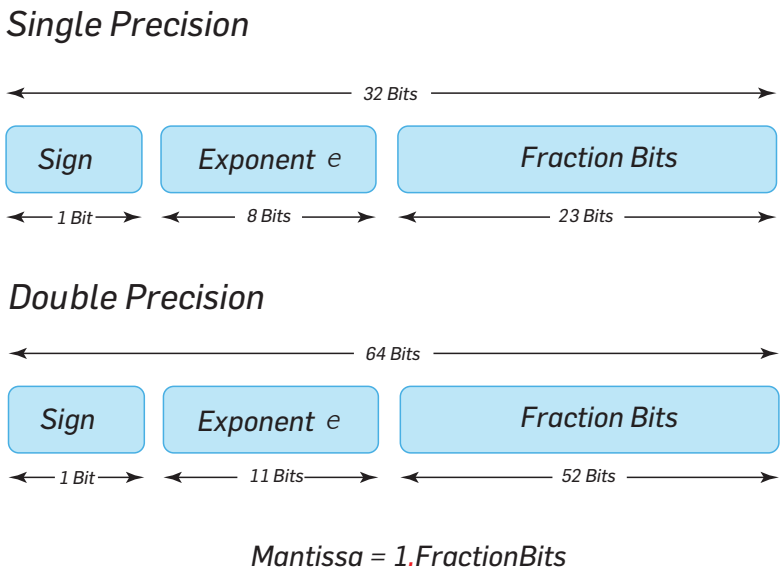


Figure 3.1: `IEEE754` notation

3.3.1 Error analysis

In this section an error analysis of the proposed computation scheme depicted in Figure 3.2 is carried out for an input $x = 2^{e_u} \cdot m$.
Let $P_{(n,\sqrt{m})}(m)$ and $P_{\left(n,\frac{1}{\sqrt{m}}\right)}(m)$ be the n^{th} degree polynomial approximations within the range $[1\ ,\ 2)$ of $\sqrt{m}$ and $\dfrac{1}{\sqrt{m}}$, respectively.
Two cases can be enumerated [1]:

1. e_u is odd

$$\sqrt{x} = \sqrt{2} \cdot 2^{\frac{e_u - 1}{2}} \cdot \sqrt{m} \tag{3.14a}$$

$$\frac{1}{\sqrt{x}} = \frac{1}{\sqrt{2}} \cdot 2^{\frac{1 - e_u}{2}} \cdot \frac{1}{\sqrt{m}} \tag{3.14b}$$

with the proposed method the two functions are evaluated as follows:

$$\sqrt{x} \approx \sqrt{2} \cdot 2^{\frac{e_u - 1}{2}} \cdot P_{(n,\sqrt{m})}(m) \tag{3.15a}$$

$$\frac{1}{\sqrt{x}} \approx \frac{1}{\sqrt{2}} \cdot 2^{\frac{1 - e_u}{2}} \cdot P_{\left(n,\frac{1}{\sqrt{m}}\right)}(m) \tag{3.15b}$$

The relative errors are computed according to:

[1] The case $e_u = 0$ is trivial and will be not discussed.

For the proposed algorithm the input data are
IEEE 754 single precision floating point numbers

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

31	30 … 23	22 … 0
Sign Bit s	Exponent e expressed in excess 127 $e = e_u + 127$	Mantissa $m = 1.fractionBits$ 23 fractional bits + 1 hidden integer bit

Square root and inverse square root functions are
approximated with n^{th} degree polynomial in the range [1,2)

$$P_n(x) = p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n$$

The sets $\mathbb{A}$ and $\mathbb{B}$ are used to compute the square root and inverse square operations, respectively.

$$\mathbb{A} := \left\{ p_0^{\text{sqrt}}, p_1^{\text{sqrt}}, \cdots, p_n^{\text{sqrt}} \right\}$$

$$\mathbb{B} := \left\{ p_0^{\text{isqrt}}, p_1^{\text{isqrt}}, \cdots, p_n^{\text{isqrt}} \right\}$$

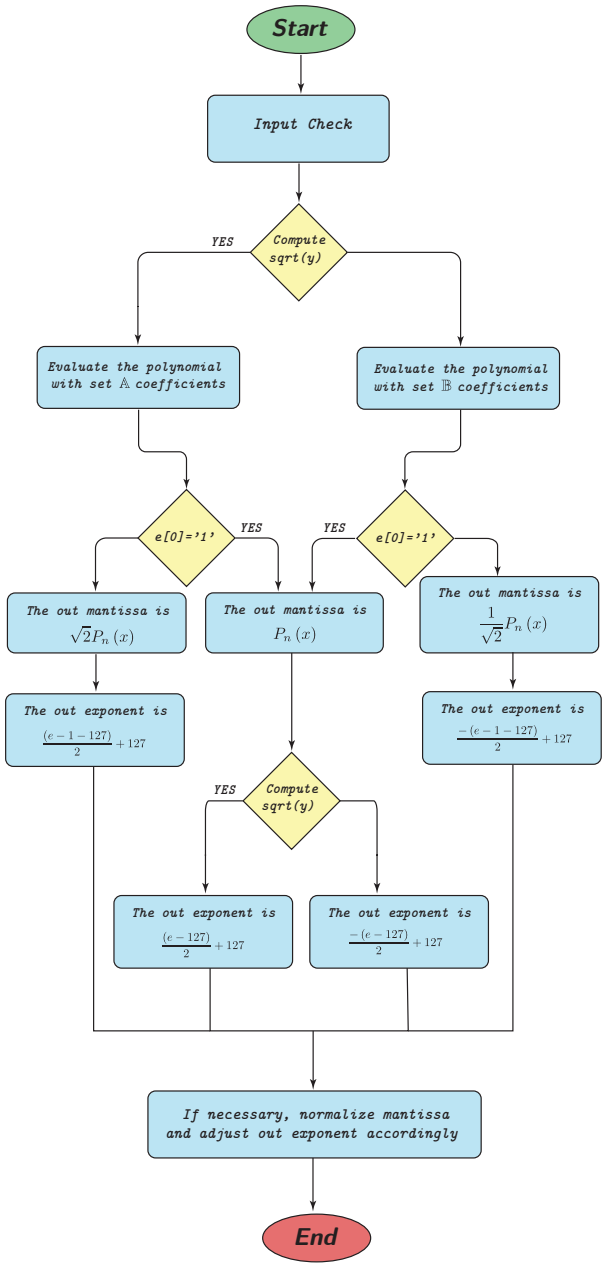


Figure 3.2: Overall computation scheme of the `Twente ISQRT SQRT` architecture

$$\varepsilon^{\text{odd}}_{\sqrt{x}} = \frac{\sqrt{2}\cdot 2^{\frac{e_u-1}{2}}\cdot P_{(n,\sqrt{m})}(m) - \left(\sqrt{2}\cdot 2^{\frac{e_u-1}{2}}\cdot\sqrt{m}\right)}{\sqrt{2}\cdot 2^{\frac{e_u-1}{2}}\cdot\sqrt{m}}$$
$$= \frac{P_{(n,\sqrt{m})}(x) - \sqrt{m}}{\sqrt{m}} \tag{3.16a}$$

$$\varepsilon^{\text{odd}}_{\frac{1}{\sqrt{x}}} = \frac{\frac{1}{\sqrt{2}}\cdot 2^{\frac{1-e_u}{2}}\cdot P_{\left(n,\frac{1}{\sqrt{m}}\right)}(m) - \left(\frac{1}{\sqrt{2}}\cdot 2^{\frac{1-e_u}{2}}\cdot\frac{1}{\sqrt{m}}\right)}{\frac{1}{\sqrt{2}}\cdot 2^{\frac{1-e_u}{2}}\cdot\frac{1}{\sqrt{m}}}$$
$$= \frac{P_{\left(n,\frac{1}{\sqrt{m}}\right)}(m) - \frac{1}{\sqrt{m}}}{\frac{1}{\sqrt{m}}} \tag{3.16b}$$

2. e_u is even

$$\sqrt{x} = 2^{\frac{e_u}{2}}\cdot\sqrt{m} \tag{3.17a}$$
$$\frac{1}{\sqrt{x}} = 2^{\frac{-e_u}{2}}\cdot\frac{1}{\sqrt{m}} \tag{3.17b}$$

The relative errors are computed as:

$$\varepsilon^{\text{even}}_{\sqrt{x}} = \frac{2^{\frac{e_u}{2}}\cdot P_{(n,\sqrt{m})}(m) - \left(2^{\frac{e_u}{2}}\cdot\sqrt{m}\right)}{2^{\frac{e_u}{2}}\cdot\sqrt{m}}$$
$$= \frac{P_{(n,\sqrt{m})}(m) - \sqrt{m}}{\sqrt{m}} \tag{3.18a}$$

$$\varepsilon^{\text{even}}_{\frac{1}{\sqrt{x}}} = \frac{2^{\frac{-e_u}{2}}\cdot P_{\left(n,\frac{1}{\sqrt{m}}\right)}(m) - \left(2^{\frac{-e_u}{2}}\cdot\frac{1}{\sqrt{m}}\right)}{2^{\frac{-e_u}{2}}\cdot\frac{1}{\sqrt{m}}}$$
$$= \frac{P_{\left(n,\frac{1}{\sqrt{m}}\right)}(m) - \frac{1}{\sqrt{m}}}{\frac{1}{\sqrt{m}}} \tag{3.18b}$$

In conclusion, with the proposed computation scheme, the relative error is meaningfully represented by the mantissa value only. The relative error can be related to the ulp (unit in last place) by means of the of the following corollary demonstrated in [41]:

Let x_1 and x_2 be two floating point number.
If $x_1 = x_2\cdot(1+\varepsilon_{\text{rel}})$, with $|\varepsilon_{\text{rel}}| < \frac{1}{2^{N+1}-1}$, then $x_1 \approx x_2$, within $\frac{1}{2}$ ulp of x_2.

3.4 Polynomial approximation of $\frac{1}{\sqrt{x}}$

3.4.1 Proposed linear approximation of $\frac{1}{\sqrt{x}}$

In this section some possibilities of approximately express $\frac{1}{\sqrt{x}}$ with the first degree polynomial

$$P_1(x) = p_1 \cdot x + p_0 \,. \tag{3.19}$$

are explored.

Approximation over a single interval

The polynomial coefficients are computed by solving the following non linear system obtained from (2.95):

$$p_1 x + p_0 - \frac{1}{\sqrt{x}} = 0 \tag{3.20a}$$

$$p_1 + p_0 - 1 + L = 0 \tag{3.20b}$$

$$2p_1 + p_0 - \frac{\sqrt{2}}{2} + L = 0 \tag{3.20c}$$

$$p_1 x_1 + p_0 - \frac{1}{\sqrt{x_1}} - L = 0 \tag{3.20d}$$

$$p_1 + \frac{1}{2x_1^{1.5}} = 0 \tag{3.20e}$$

- **Max Absolute Error** $0.1890715724 \cdot 10^{-1}$ ($L = 0.1890715724 \cdot 10^{-1}$)
- **Abscissae of error function stationary point** :
  $x_1 = 1.428369138$
- **Approximating polynomial**

$$P_1^{\mathrm{m0}}(x) = -0.2928932190x + 1.273986062 \tag{3.21}$$

$$\varepsilon(x) \equiv P_1^{\mathrm{m0}}(x) - \frac{1}{\sqrt{x}} \tag{3.22a}$$

$$\varepsilon_{\%}(x) \equiv 100 \cdot \frac{\varepsilon(x)}{\frac{1}{\sqrt{x}}} \tag{3.22b}$$

- **Error Plots** see Figures 3.3, 3.4.

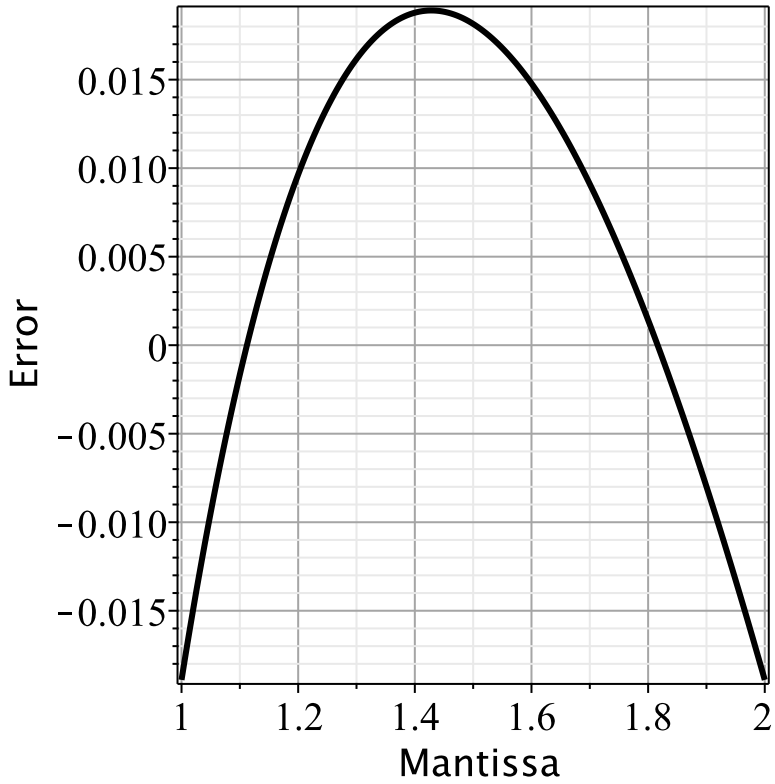


Figure 3.3: Plot of error function $\varepsilon(x)$
(see 3.22a)

Approximation over two intervals

To further reduce the error of the polynomial (3.21), it is possible to apply again the Chebyshev criterion within $[1, x_1]$ and $(x_1, 2)$. This leads to the following approximation:

$$P_1^{\mathrm{m1}}(x) = \begin{cases} -0.3811682860\,x + 1.375720238 \text{ if } x \in [1, x_1] \\ -0.2267415871\,x + 1.156506125 \text{ if } x \in (x_1, 2) \end{cases} \tag{3.23}$$

The maximum absolute error within $[1, x_1]$ is $0.5448047491 \cdot 10^{-2}$ and occurs at $x_2 = 1.198309025$, while within $(x_1, 2)$ is $0.4083830056 \cdot 10^{-2}$ and occurs at $x_3 = 1.694178690$
The error functions associated with the approximation in (3.23) are (see Figures 3.5 and 3.6):

$$\varepsilon^{m1}(x) \equiv P_1^{\mathrm{m1}}(x) - \frac{1}{\sqrt{x}} \tag{3.24a}$$

$$\varepsilon_{\%}^{m1}(x) \equiv \frac{\varepsilon^{m1}(x)}{\frac{1}{\sqrt{x}}} \tag{3.24b}$$

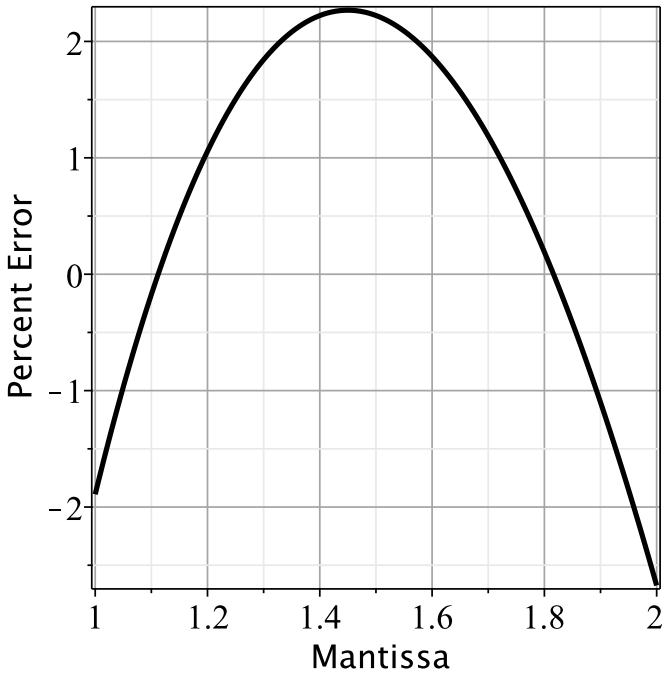


Figure 3.4: Percent error $\varepsilon_{\%}(x)$ (see 3.22b)

The least squares solution

A linearisation of $\frac{1}{\sqrt{x}}$ within the interval [1 , 2) is obtained with the `polyfit`. The routine computes the following least squares polynomial:

$$P_1^{\mathrm{m2}}(x) = -0.2843 \cdot x + 1.2549 \tag{3.25}$$

To compare the accuracy of (3.21) and (3.25), let us define the following error functions:

$$\varepsilon_{\mathrm{abs}}^{m2}(x) \equiv \left| P_1^{\mathrm{m2}}(x) - \frac{1}{\sqrt{x}} \right| \tag{3.26a}$$

$$\varepsilon_{\mathrm{rel}}^{m2}(x) \equiv \frac{\varepsilon^{m2}(x)}{\frac{1}{\sqrt{x}}} \tag{3.26b}$$

$$\varepsilon_{\mathrm{abs}}^{m0}(x) \equiv \left| P_1^{\mathrm{m0}}(x) - \frac{1}{\sqrt{x}} \right| \tag{3.26c}$$

$$\varepsilon_{\mathrm{rel}}^{m0}(x) \equiv \frac{\varepsilon_{\mathrm{abs}}(x)}{\frac{1}{\sqrt{x}}} \tag{3.26d}$$

The plots of absolute and relative error functions, are reported in Figures 3.7 and 3.8, respectively.

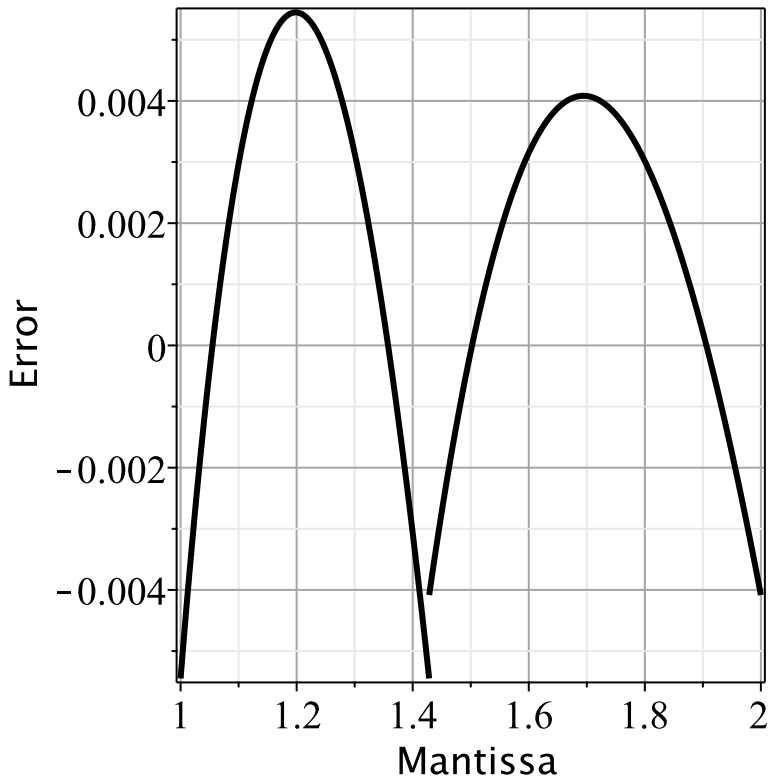


Figure 3.5: Error function (see (3.24a))

From this figures, it can be observed that, within $x \in [1.15, 180]$, the solution based on the least squares criterion is more accurate than the one Chebyshev criterion. However, the Chebyshev solution produces the best *worst-case* scenario over the entire approximation range.

Comparisons with recent contributions

In this section, the proposed approach is compared with those recently published in journal papers.

Comparison with Moroz et al. [57]

The improvement of our solution with respect to the one of Moroz et al. [57] is highlighted by the comparison of the plots of Figure 3.9 .
The error associated with the herein proposed approximation (see Figure 3.9) is two order of magnitude less than the one of Moroz *et al.* [57].

Comparison with Parrilla et al. [58]

The table shown in Figure 3.10, depicts the relative initial guess approximation error, computed with the method proposed by Parrilla et al. [58]. Their best worst case proposed relative error is 0.0333. This value

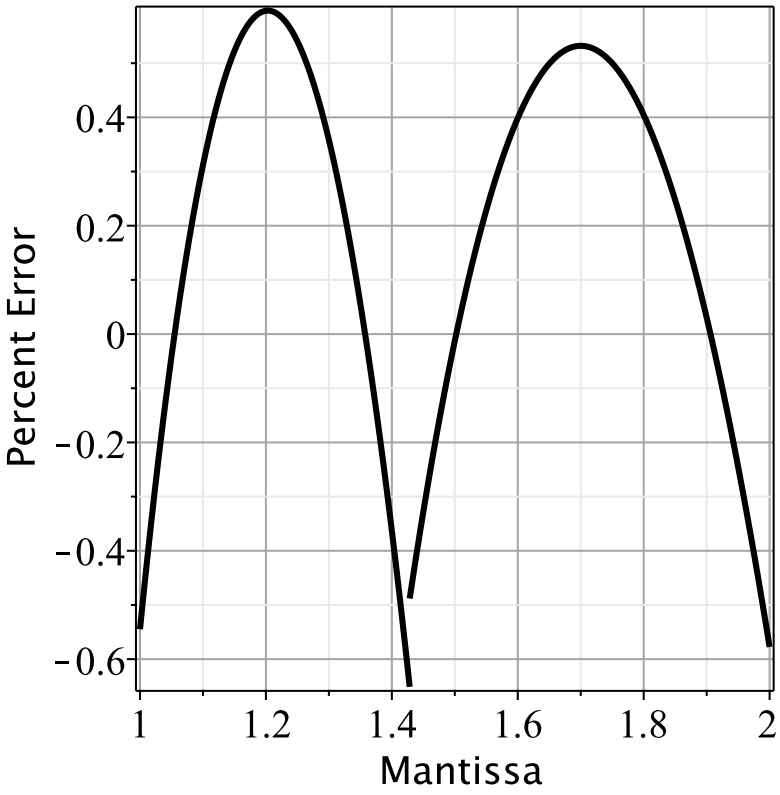


Figure 3.6: Error function (see (3.24b))

is one order of magnitude greater than the error due to the approximation herein proposed.

Comparison with Hasnat et al. [42]

The error comparison between Hasnat et al [42] and the seven intervals linearisation herein adopted in (3.27), is depicted in Figure 3.11.
The approximation proposed by (3.27) has been deduced by applying the Chebyshev min-max criterion within the same intervals proposed in [42].

$$P_1^{\mathrm{m2}}(x) = \begin{cases} 1.50971358702244 - 0.500000000000000 \cdot x \\ 1.34377614452159 - 0.358217995563320 \cdot x \\ 1.28092920145555 - 0.310455290827024 \cdot x \\ 1.22617183547953 - 0.272442417019179 \cdot x \\ 1.17789649241698 - 0.241599852241761 \cdot x \\ 1.13491593196515 - 0.216167221275863 \cdot x \\ 1.09632697261222 - 0.194903334931431 \cdot x \end{cases} \tag{3.27}$$

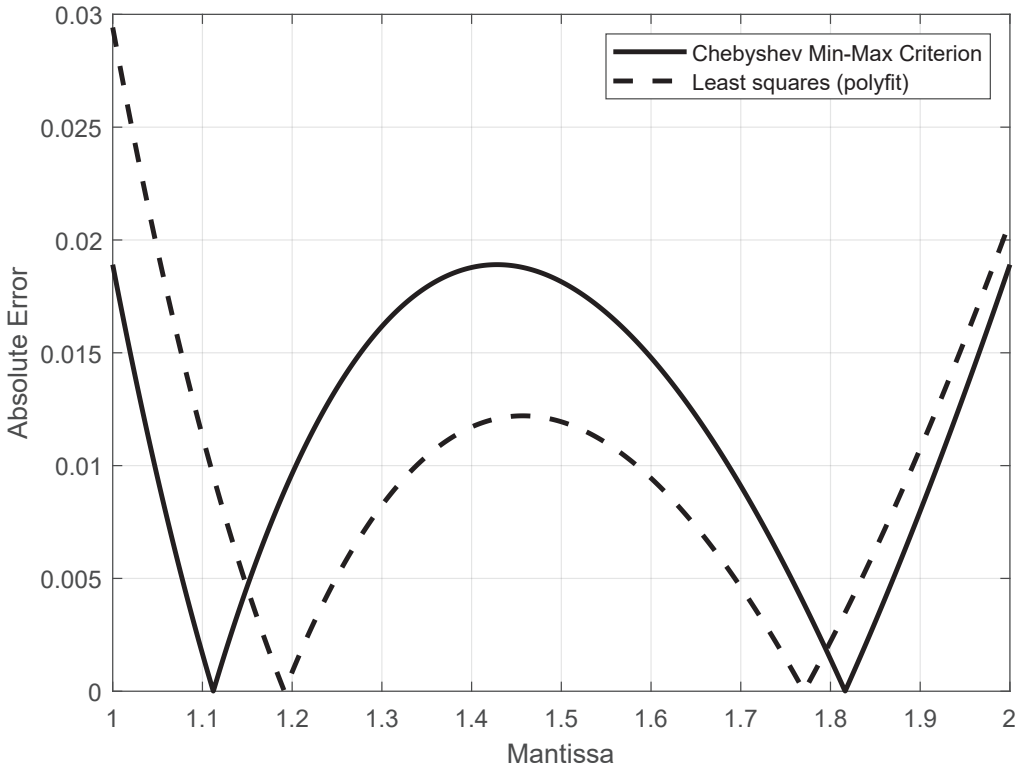


Figure 3.7: Comparison between $\varepsilon_{\text{abs}}^{m2}(x)$ and $\varepsilon_{\text{abs}}^{m0}(x)$

3.4.2 Proposed quadratic approximations of $\frac{1}{\sqrt{x}}$

In this section some possibilities of approximately express $\frac{1}{\sqrt{x}}$ with the second degree polynomial

$$P_1(x) = p_2 \cdot x^2 + p_1 \cdot x + p_0 \,. \tag{3.28}$$

are explored.

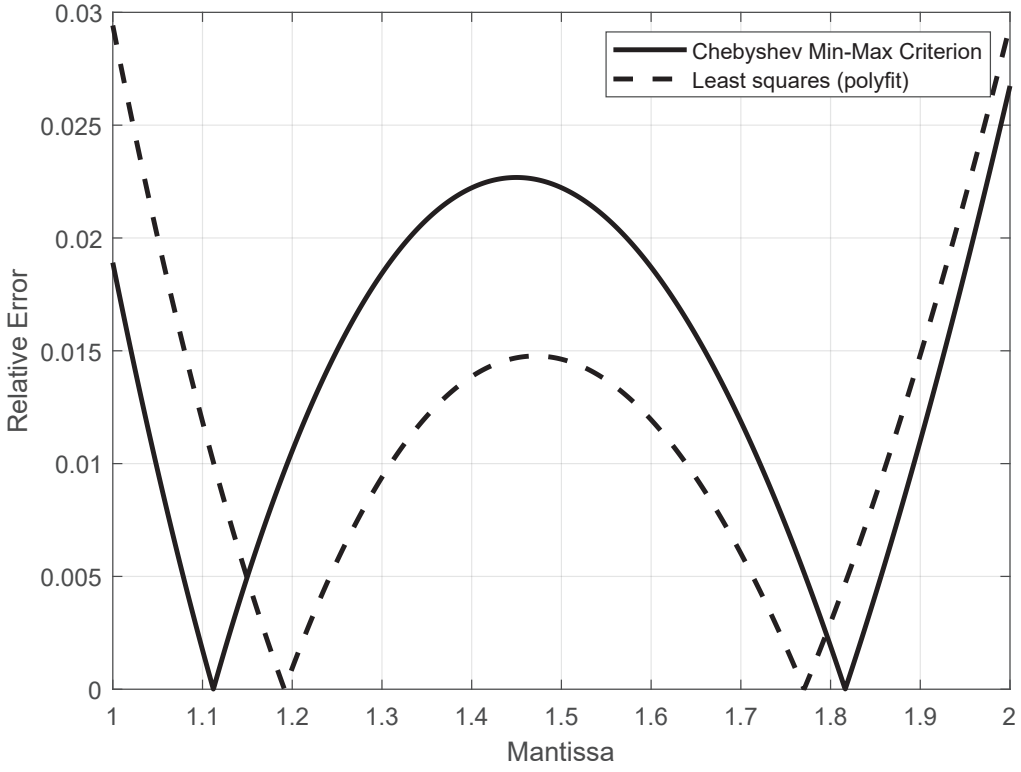


Figure 3.8: Comparison between $\varepsilon_{\text{rel}}^{m2}$ and $\varepsilon_{\text{rel}}^{m0}$

Approximation over a single interval

The polynomial coefficients are computed by solving the following non linear system obtained from (2.95):

$$p_2 + p_1 + p_0 - 1 - L = 0 \tag{3.29a}$$

$$p_2 x_1^2 + p_1 x_1 + p_0 - \frac{1}{\sqrt{x_1}} + L = 0 \tag{3.29b}$$

$$p_2 x_2^2 + 2p_1 + p_0 - \frac{1}{\sqrt{x_2}} - L = 0 \tag{3.29c}$$

$$4p_2 + 2p_1 + p_0 - \frac{\sqrt{2}}{2} + L = 0 \tag{3.29d}$$

$$2p_2 x_1 + p_1 + \frac{1}{2x_1^{1.5}} = 0 \tag{3.29e}$$

$$2p_2 x_2 + p_1 + \frac{1}{2x_2^{1.5}} = 0 \tag{3.29f}$$

The solution of the system is:

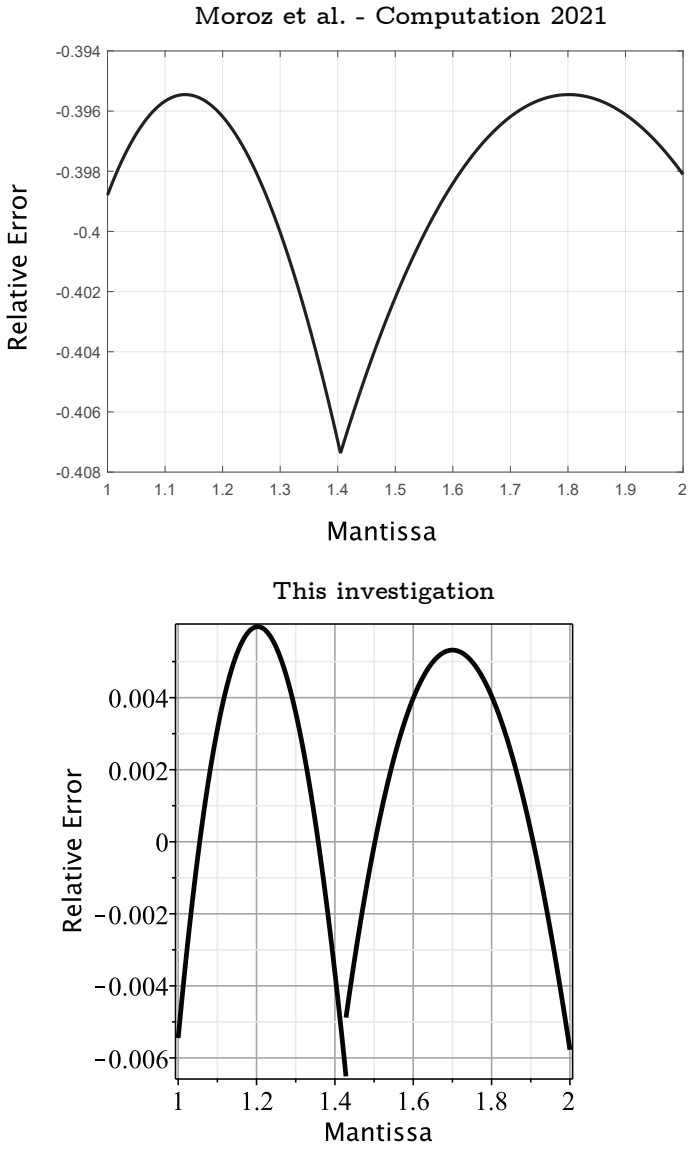


Figure 3.9: Comparison of relative error between Moroz *et al.* [57] and the proposed approximation $P_1^{\mathrm{m1}}(x)$ (see Eq. (3.23))

TABLE V
MEAN ABSOLUTE ERROR (MAE), MAXIMUM ABSOLUTE ERROR (MAXAE), MEAN RELATIVE ERROR (MRE), AND MAXIMUM RELATIVE ERROR (MAXRE) WHEN COMPUTING $(1/\sqrt{x})_{seed}$ FOR DIFFERENT VALUES OF n AND m

n	m	MAE	MAXAE	MRE	MAXRE	$nbits$
4	3	0.0854	0.3750	0.0873	0.3750	36
4	4	0.0277	0.1017	0.0287	0.0798	48
4	5	0.0189	0.1017	0.0192	0.0719	60
4	6	0.0165	0.1017	0.0160	0.0719	72
4	7	0.0150	0.0861	0.0147	0.0609	84
5	3	0.0660	0.375	0.0678	0.375	72
5	4	0.0272	0.1017	0.0279	0.0719	96
5	5	0.0148	0.0833	0.0148	0.0625	120
5	6	0.0101	0.0548	0.0102	0.0391	144
5	7	0.0079	0.047	0.0078	0.0333	168

Figure 3.10: Table from Parrilla et al. [58] where n is the number of bits addressing the table, and m is the number of bits representing the mantissa of the seed value.

- **Max Absolute Error**: $0.2710674567 \cdot 10^{-2}$ ($L = -0.2710674567 \cdot 10^{-2}$)
- **Abscissae of error function stationary points**:
  $x_1 = 1.212269880$, $x_2 = 1.712589631$
- **Approximating polynomial**

$$P_2^{\mathrm{m0}}(x) = 0.1514108797000x^2 - 0.7417045090000x + 1.5875829550000 \tag{3.30}$$

$$\varepsilon(x) \equiv P_2^{\mathrm{m0}}(x) - \frac{1}{\sqrt{x}} \tag{3.31a}$$

$$\varepsilon_{\%}(x) \equiv 100 \cdot \frac{\varepsilon(x)}{\frac{1}{\sqrt{x}}} \tag{3.31b}$$

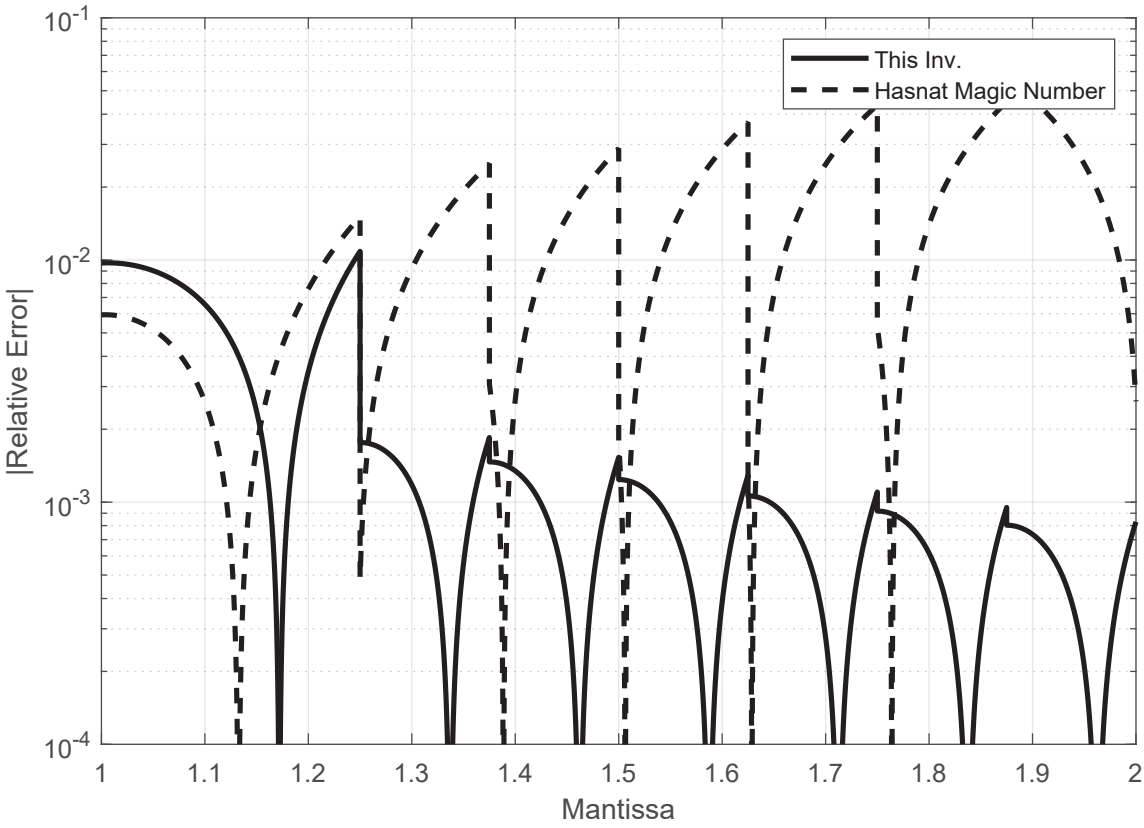


Figure 3.11: Error comparison between the magic constants proposed by Hasnat et al. [42] and the linearisation herein proposed in 3.27

- **Error Plots** see Figures 3.12, 3.13

Modus operandi I: solution improvement by means of error function linearisation

To refine the approximation given by (3.30), a piecewise linearisation ε_{lin} of the error function ε has been applied, as visually depicted in Figure 3.14. For this purpose the approximation domain is divided into 3 parts. The straight line equations assumed as error function approximation are:

$$\varepsilon_{\text{lin}}(x) = \begin{cases} 0.0255398888200 \cdot x - 0.0282505633900, & \text{if } x \in [1, x_1] \\ -0.0108357687700 \cdot x + 0.0158465506800, & \text{if } x \in (x_1\ ,\ x_2] \\ 0.0188627472000 \cdot x - 0.0350148198500, & \text{if } x \in (x_2, 2) \end{cases} \tag{3.32}$$

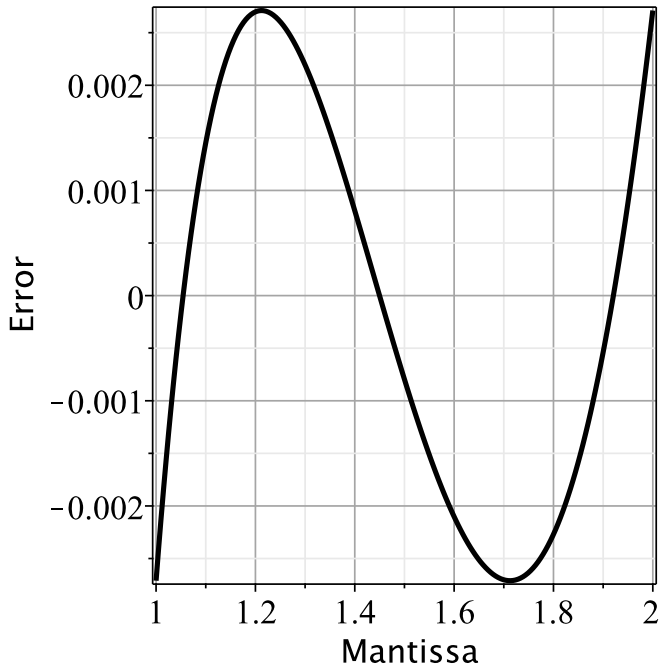


Figure 3.12: Error function $\varepsilon(x)$ (see (3.31a))

Consequently, the approximating polynomial is rewritten as:

$$
\begin{aligned}
{P_2}^{\mathrm{m1}}(x) &\equiv P_2^{\mathrm{m0}}(x) - \varepsilon_{\mathrm{lin}}(x) \\
&= \begin{cases} 0.1514108797000 \cdot x^2 - 0.7672443978000 \cdot x + 1.6158335180000, & \text{if } x \in [1, x_1] \\ 0.1514108797000 \cdot x^2 - 0.7308687402000 \cdot x + 1.5717364040000, & \text{if } x \in (x_1\ ,\ x_2] \\ 0.1514108797000 \cdot x^2 - 0.7605672562000 \cdot x + 1.6225977750000, & \text{if } x \in (x_2, 2) \end{cases}
\end{aligned} \tag{3.33}
$$

The error function $\varepsilon^{\mathrm{m1}}(x)$ and the percent $\varepsilon_{\%}^{\mathrm{m1}}(x)$ error associated with ${P_2}^{\mathrm{m1}}(x)$ approximation are shown in Figures 3.15 and 3.16, respectively.

$$
\varepsilon^{\mathrm{m1}}(x) \equiv {P_2}^{\mathrm{m1}}(x) - \frac{1}{\sqrt{x}} \tag{3.34a}
$$

$$
\varepsilon_{\%}^{\mathrm{m1}}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m1}}(x)}{\frac{1}{\sqrt{x}}} \tag{3.34b}
$$

Due to linear approximation, the coefficient of the second degree term in equation (3.33) does not change over the three intervals. This results into a reduced coefficients LUT size.
To further refine the approximation provided by ${P_2}^{\mathrm{m1}}(x)$, the Chebyshev criterion has be again applied to express the error function $\varepsilon^{\mathrm{m1}}(x)$ (see Figure 3.15) within the ranges $[1, x_1]$ and $(x_2, 2)$.

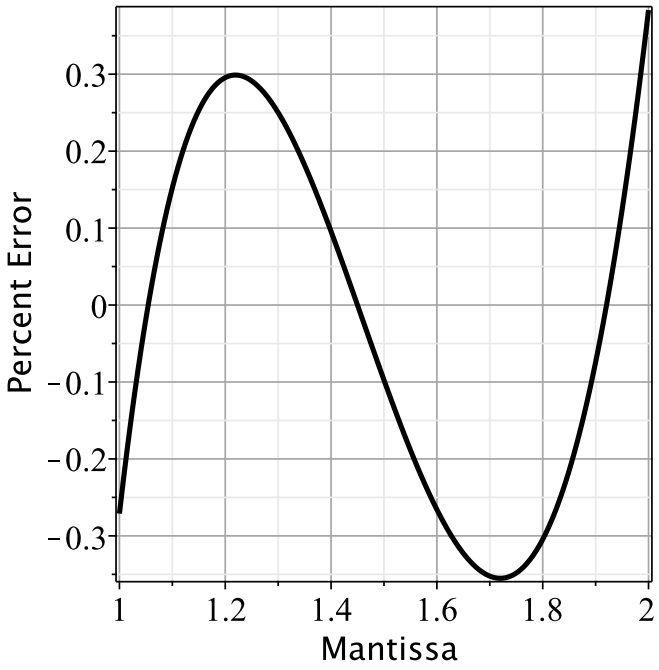


Figure 3.13: Percent error $\varepsilon_{\%}$ (see (3.31b))

The coefficients of the second degree polynomial used to approximate the $\varepsilon^{\mathrm{m1}}(x)$ in the range $[1, x_1]$ are

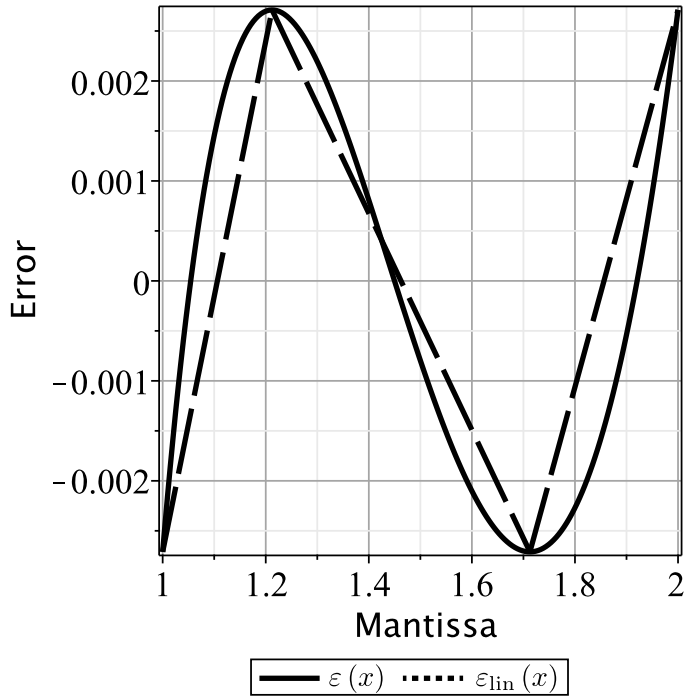


Figure 3.14: Error function $\varepsilon(x)$ (see (3.31a)) and its piecewise linearisation $\varepsilon_{\text{lin}}(x)$ (see (3.34a))

obtained by solving the following non linear system:

$$p_2^{\text{r2p1}} + p_1^{\text{r2p1}} + p_0^{\text{r2p1}} + 10^{-10} - L_{\text{r2p1}} = 0 \tag{3.35a}$$

$$p_2^{\text{r2p1}} \cdot x_{\text{1r2p1}}^2 + p_1^{\text{r2p1}} \cdot x_{\text{1r2p1}} + p_0^{\text{r2p1}} - 0.1514108797000 \cdot x_{\text{1r2p1}}^2 + 0.7672443978000 \cdot x_{\text{1r2p1}} - 1.6158335180000 + \frac{1}{\sqrt{x_{\text{1r2p1}}}} + L_{\text{r2p1}} = 0 \tag{3.35b}$$

$$p_2^{\text{r2p1}} \cdot x_{\text{2r2p1}}^2 + p_1^{\text{r2p1}} \cdot x_{\text{2r2p1}} + p_0^{\text{r2p1}} - 0.1514108797000 \cdot x_{\text{2r2p1}}^2 + 0.7672443978000 \cdot x_{\text{2r2p1}} - 1.6158335180000 + \frac{1}{\sqrt{x_{\text{2r2p1}}}} - L_{\text{r2p1}} = 0 \tag{3.35c}$$

$$1.4695982620000 \cdot p_2^{\text{r2p1}} + 1.2122698800000 \cdot p_1^{\text{r2p1}} + p_0^{\text{r2p1}} + 3.0000000000000 \cdot 10^{-10} + L_{\text{r2p1}} = 0 \tag{3.35d}$$

$$2p_2^{\text{r2p1}} x_{\text{1r2p1}} + p_1^{\text{r2p1}} - 0.3028217594000 \cdot x_{\text{1r2p1}} + 0.7672443978000 - \frac{1}{2\left(x_{\text{1r2p1}}\right)^{1.5}} = 0 \tag{3.35e}$$

$$2p_2^{\text{r2p1}} x_{\text{2r2p1}} + p_1^{\text{r2p1}} - 0.3028217594000 \cdot x_{\text{2r2p1}} + 0.7672443978000 - \frac{1}{2\left(x_{\text{2r2p1}}\right)^{1.5}} = 0 \tag{3.35f}$$

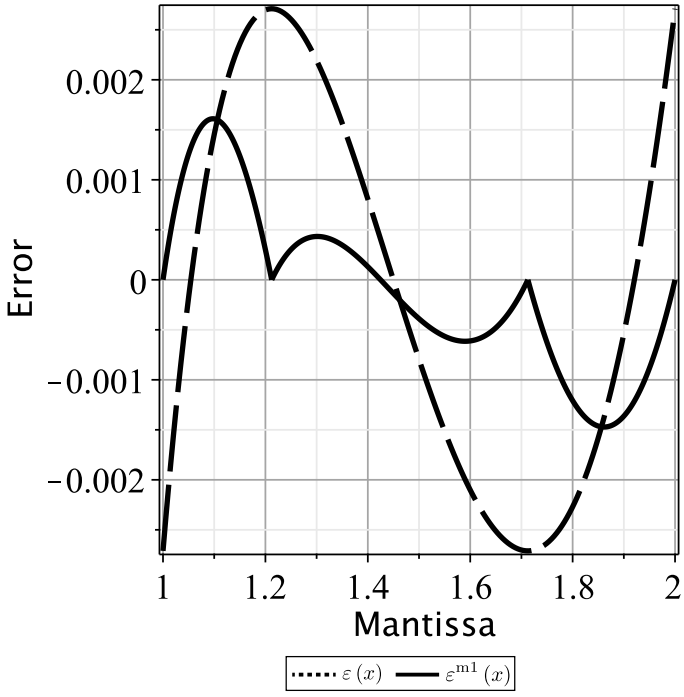


Figure 3.15: Error function $\varepsilon^{\mathrm{m1}}(x)$ (see (3.34a))

The error unction f_{ref1} is approximated in the range $[1, x_1]$ by means of

$$\begin{aligned}\widetilde{\varepsilon}^{\mathrm{m1}}_{[1,x_1]}(x) &= p_2^{\mathrm{r2p1}} \cdot x^2 + p_1^{\mathrm{r2p1}} \cdot x + p_0^{\mathrm{r2p1}} \\ &= -0.1424694987 \cdot x^2 + 0.3145559533 \cdot x - 0.1720201176 \end{aligned} \tag{3.36}$$

Using this approximation, the maximum absolute error $\left|L_{\mathrm{r2p1}}\right|$ computed within the range $[1, x_1]$ is 0.6633715316e-4 and will occur at $x_{1\mathrm{r2p1}} = 1.0508317500000$ and $x_{2\mathrm{r2p1}} = 1.1569720380000$.

The coefficients of the second degree polynomial used to approximate the $\varepsilon^{\mathrm{m1}}(x)$, computed within the

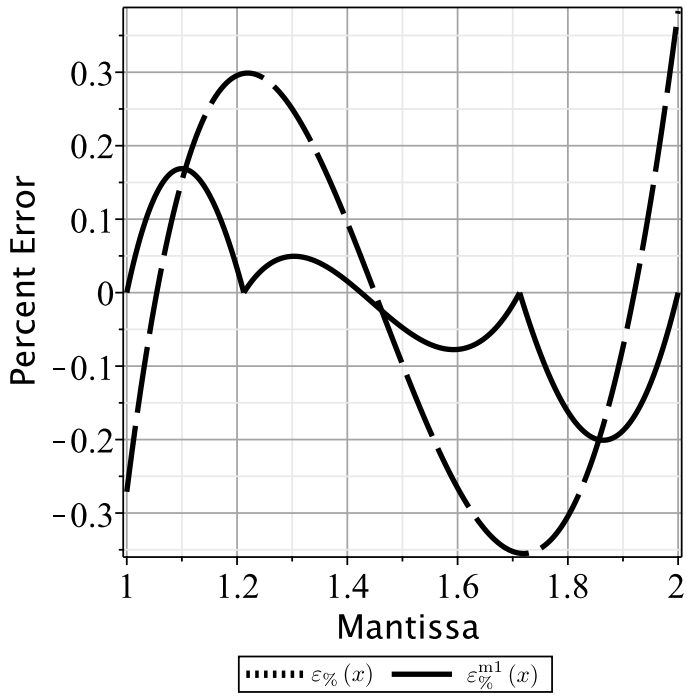


Figure 3.16: Percent error for 2nd degree polynomial approximating $\frac{1}{\sqrt{x}}$ after first refinement

range $(x_2, 2)$, are obtained by solving the following non linear system obtained from (2.95)::

$$2.9329632440000 \cdot p_2^{\mathrm{r2p3}} + 1.7125896310000 \cdot p_1^{\mathrm{r2p3}} + p_0^{\mathrm{r2p3}} - 2.0000000000000 \cdot 10^{-10} - L_{\mathrm{r2p3}} = 0 \tag{3.37a}$$

$$p_2^{\mathrm{r2p3}} x_{1\mathrm{r2p3}}^2 + p_1^{\mathrm{r2p3}} x_{1\mathrm{r2p3}} + p_0^{\mathrm{r2p3}} - 0.1514108797000 \cdot x_{1\mathrm{r2p3}}^2 + 0.7605672562000 \cdot x_{1\mathrm{r2p3}} - 1.6225977750000 + \frac{1}{\sqrt{x_{1\mathrm{r2p3}}}} + L_{\mathrm{r2p3}} = 0 \tag{3.37b}$$

$$p_2^{\mathrm{r2p3}} x_{2\mathrm{r2p3}}^2 + p_1^{\mathrm{r2p3}} x_{2\mathrm{r2p3}} + p_0^{\mathrm{r2p3}} - 0.1514108797000 \cdot x_{2\mathrm{r2p3}}^2 + 0.7605672562000 \cdot x_{2\mathrm{r2p3}} - 1.6225977750000 + \frac{1}{\sqrt{x_{2\mathrm{r2p3}}}} - L_{\mathrm{r2p3}} = 0 \tag{3.37c}$$

$$3.9999995240000 \cdot p_2^{\mathrm{r2p3}} + 1.9999998807907 \cdot p_1^{\mathrm{r2p3}} + \cdot p_0^{\mathrm{r2p3}} + 2.7000000000000 \cdot 10^{-9} + L_{\mathrm{r2p3}} = 0 \tag{3.37d}$$

$$2 \cdot p_2^{\mathrm{r2p3}} x_{1\mathrm{r2p3}} + p_1^{\mathrm{r2p3}} - 0.3028217594000 \cdot x_{1\mathrm{r2p3}} + 0.7605672562000 - \frac{1}{2 \cdot \left(x_{1\mathrm{r2p3}}\right)^{1.5}} = 0 \tag{3.37e}$$

$$2 \cdot p_2^{\mathrm{r2p3}} x_{2\mathrm{r2p3}} + p_1^{\mathrm{r2p3}} - 0.3028217594000 \cdot x_{2\mathrm{r2p3}} + 0.7605672562000 - \frac{1}{2 \cdot \left(x_{2\mathrm{r2p3}}\right)^{1.5}} = 0 \tag{3.37f}$$

The error function $\varepsilon^{\mathrm{m1}}(x)$ is approximated in the range $[x_2\ ,\ 2)$ by means of

$$\begin{aligned}\widetilde{\varepsilon}^{\,\mathrm{m1}}_{(x_2,2)}(x) &= p_2^{\mathrm{r2p3}}\cdot x^2 + p_1^{\mathrm{r2p3}}\cdot x + p_0^{\mathrm{r2p3}}\\ &= 0.0710966017200\cdot x^2 - 0.2641389477000\cdot x + 0.2438646959000\end{aligned} \tag{3.38}$$

Using this approximation, the maximum absolute error $\left|L_{\mathrm{r2p3}}\right|$, computed within the range $[x_2\ ,\ 2)$, is and will occur at $x_{1\mathrm{r2p3}} = 1.7820023390000$
and $x_{2\mathrm{r2p3}} = 1.9257121440000$.

In conclusion, the overall error function is defined as follows:

$$\widetilde{\varepsilon}^{\,\mathrm{m1}}(x) = \begin{cases}\widetilde{\varepsilon}^{\,\mathrm{m1}}_{[1,x_1]}(x)\,\text{for } x\in[1,x_1]\\ 0\,\text{for } x\in(x_1\ ,\ x_2]\\ \widetilde{\varepsilon}^{\,\mathrm{m1}}_{[x_2\ ,2)}(x)\,\text{for } x\in(x_2,2)\end{cases} \tag{3.39}$$

and the current *modus operandi* leads to the following approximating polynomial

$$\begin{aligned}P_2{}^{\mathrm{m1b}}(x) &\equiv P_2^{\mathrm{m0}}(x) - \varepsilon_{\mathrm{lin}}(x) - \widetilde{\varepsilon}^{\,\mathrm{m1}}(x)\\ &= P_2{}^{\mathrm{m1}}(x) - \widetilde{\varepsilon}^{\,\mathrm{m1}}(x)\\ &= \begin{cases}0.2938803784000\cdot x^2 - 1.0818003510000\cdot x + 1.7878536360000, & \text{if } x\in[1,x_1]\\ 0.1514108797000\cdot x^2 - 0.7308687402000\cdot x + 1.5717364040000, & \text{if } x\in(x_1\ ,\ x_2]\\ 0.0803142779800\cdot x^2 - 0.4964283085000\cdot x + 1.3787330790000, & \text{if } x\in(x_2,2)\end{cases}\end{aligned} \tag{3.40}$$

$$\varepsilon^{\mathrm{m1b}}(x) = P_2{}^{\mathrm{m1b}}(x) - \frac{1}{\sqrt{x}} \tag{3.41a}$$

$$\varepsilon^{\mathrm{m1b}}_{\%}(x) = 100\cdot\frac{\varepsilon^{\mathrm{m1b}}(x)}{\frac{1}{\sqrt{x}}} \tag{3.41b}$$

Solution refinement by means of optimal approximation domain division: two intervals case

Herein, a different strategy is adopted for the division of the approximation domain. In particular with reference to Figure 3.19, this last one is split in two parts.
The division point x_3 is located where the second derivative of the error function $\varepsilon(x)$ has the zero value. Thus, $\frac{1}{\sqrt{x}}$ will be separately approximated within $[1,x_3]$ and $(x_3,2)$ where $x_3 = 1.4373074411960$. The analytical steps leading to the final polynomial approximation are detailed in the followings:

I Step Using the Chebyshev min-max criterion, $\frac{1}{\sqrt{x}}$ is approximated within $[1,x_3]$ by means of the following second degree polynomial:

$$\begin{aligned}&P_2{}^{\mathrm{m2}}_{[1,x_3]}(x) = p_{2e0}x^2 + p_{1e0}x + p_{0e0}\\ &= 1.7156661790000 - 0.9517382129000\cdot x\\ &+ 0.2356472637000\cdot x^2\ \text{for } x\in[1,x_3]\end{aligned} \tag{3.42a}$$

$$\varepsilon^{\mathrm{m2}}_{[1,x_3]}(x) \equiv P_2{}^{\mathrm{m2}}_{[1,x_3]}(x) - \frac{1}{\sqrt{x}} \tag{3.42b}$$

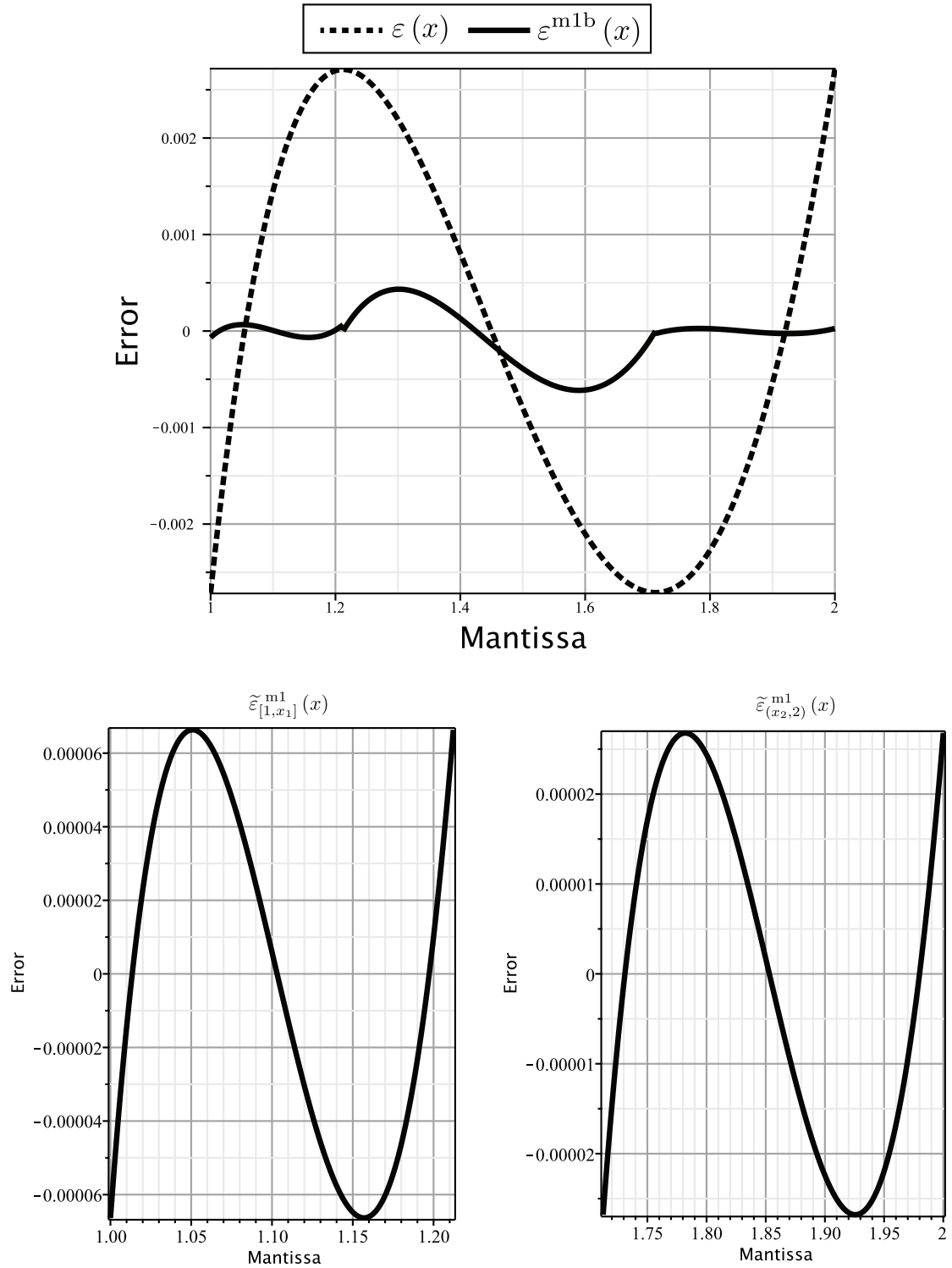


Figure 3.17: Error functions, $\widetilde{\varepsilon}^{\mathrm{m1}}_{[1,x_1]}(x)$, $\widetilde{\varepsilon}^{\mathrm{m1}}_{(x_2,2)}(x)$ and $\varepsilon^{\mathrm{m1b}}(x)$

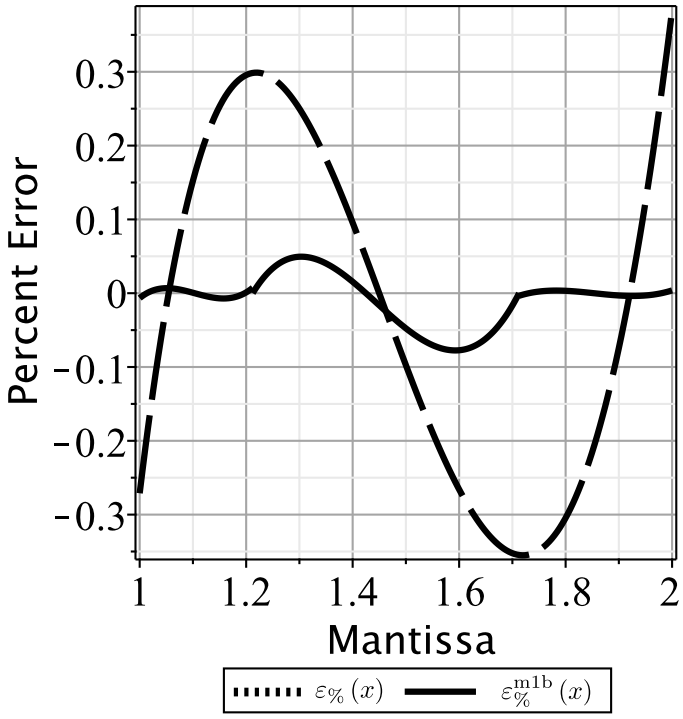


Figure 3.18:

whose coefficients have been obtained solving the following non linear system obtained from (2.95):

$$p_{2e0} + p_{1e0} + p_{0e0} - 1 - L_{e0} = 0 \tag{3.43a}$$

$$p_{2e0}x_{1e0}^2 + p_{1e0}x_{1e0} + p_{0e0} - \frac{1}{\sqrt{x_{1e0}}} + L_{e0} = 0 \tag{3.43b}$$

$$p_{2e0}x_{2e0}^2 + p_{1e0}x_{2e0} + p_{0e0} - \frac{1}{\sqrt{x_{2e0}}} - L_{e0} = 0 \tag{3.43c}$$

$$p_{2e0}x_3^2 + p_{1e0}x_3 + p0e0 - \frac{1}{\sqrt{x_3}} + L_{e0} = 0 \tag{3.43d}$$

$$2p_{2e0}x_{1e0} + p_{1e0} + \frac{1}{2x_{1e0}^{1.5}} = 0 \tag{3.43e}$$

$$2p_{2e0}x_{2e0} + p_{1e0} + \frac{1}{2x_{2e0}^{1.5}} = 0 \tag{3.43f}$$

The maximum absolute error associated with the approximation of (3.42b) is $0.4247701 \cdot 10^{-7}$ ($L_{e0} = -0.4247701 \cdot 10^{-7}$) and occurs at $x_{1e0} = 1.1006528$ and $x_{2e0} = 1.3193455$

II Step Using the Chebyshev min-max criterion, $\frac{1}{\sqrt{x}}$ is approximated within $(x_3, 2)$ by means of the fol-

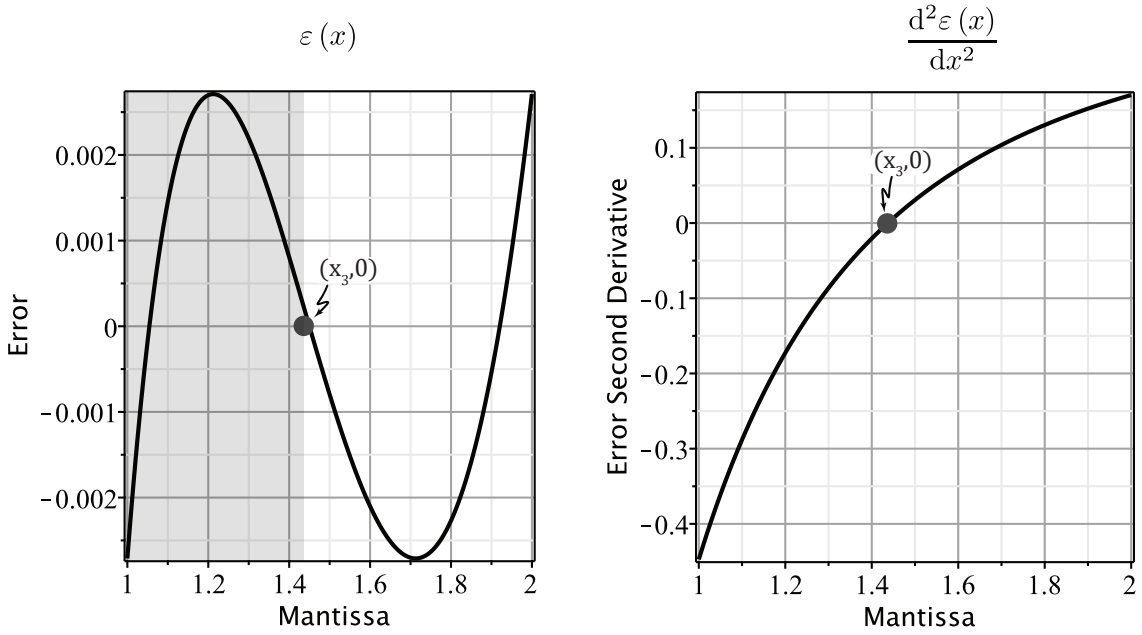


Figure 3.19: Approximation domain division into 2 parts

lowing second degree polynomial:

$$P_{2_{(x_3,2)}}^{\mathrm{m2}}(x) = p_{2e1}x^2 + p_{1e1}x + p_{0e1}$$
$$= 1.2036534 - 0.3113311x + 0.0336398x^2 \; x \in (x_3, 2) \tag{3.44a}$$
$$\varepsilon_{(x_3,2)}^{\mathrm{m2}}(x) \equiv P_{2_{(x_3,2)}}^{\mathrm{m2}}(x) - \frac{1}{\sqrt{x}} \tag{3.44b}$$

whose coefficients have been obtained solving the following non linear system obtained from

(2.95):

$$p_{2e1}x_3^2 + p_{1e1}x_3 + p_{0e1} - \frac{1}{\sqrt{x_3}} - L_{e1} = 0 \tag{3.45a}$$

$$p_{2e1}x_{1e1}^2 + p_{1e1}x_{1e1} + p_{0e1} - \frac{1}{\sqrt{x_{1e1}}} + L_{e1} = 0 \tag{3.45b}$$

$$p_{2e1}x_{2e1}^2 + p_{1e1}x_{2e1} + p_{0e1} - \frac{1}{\sqrt{x_{2e1}}} - L_{e1} = 0 \tag{3.45c}$$

$$4p_{2e1} + 2p_{1e1} + p_{0e1} - \frac{\sqrt{2}}{2} + L_{e1} = 0 \tag{3.45d}$$

$$2p_{2e1}x_{1e1} + p_{1e1} + \frac{1}{2x_{1e1}^{1.5}} = 0 \tag{3.45e}$$

$$2p_{2e1}x_{2e1} + p_{1e1} + \frac{1}{2x_{2e1}^{1.5}} = 0 \tag{3.45f}$$

Since both $x_{1e1} = 2.0000756$ and $x_{2e1} = 3.4852205$ (abscissae of error function stationary points) fall outside the range $(x_3, 2)$, as observed in Figure 3.20, the approximation proposed in (3.44a) is not feasible for our purposes. Therefore, an alternative strategy is adopted.

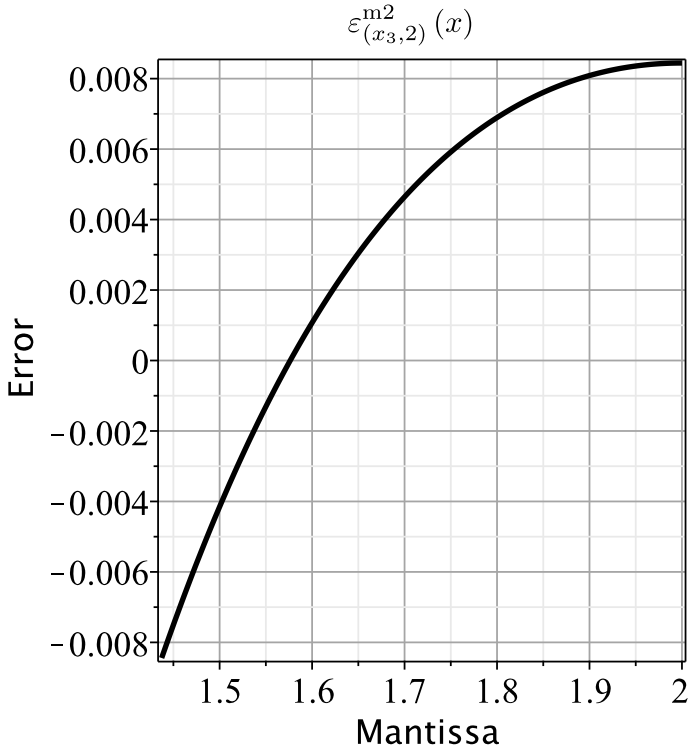


Figure 3.20: Error associate g_{int2temp} approximation

III Step Using the Chebyshev min-max criterion, $\varepsilon^{\mathrm{m2}}_{(x_3,2)}(x)$ is now approximated within $(x_3,2)$ by means of the following second degree polynomial:

$$\begin{aligned}&\widetilde{\varepsilon}^{\,\mathrm{m2}}_{(x_3,2)}(x) = p_{0e2} + p_{1e2}\cdot x + p_{2e2}\cdot x^2\\&-0.2385798933000 + 0.2546282768000\cdot x - 0.0656257091800\cdot x^2\end{aligned} \tag{3.46}$$

Where the coefficients of $\widetilde{\varepsilon}^{\,\mathrm{m2}}_{(x_3,2)}(x)$ have been computed by solving the following non linear system obtained from (2.95):

$$\begin{aligned}&p_{2e2}x_3^2 + p_{1e2}x_3 + p_{0e2} - p_{0e1} - p_{1e1}x_3 - p_{2e1}x_3^2\\&+\frac{1}{\sqrt{x3}} - L_{e2} = 0\end{aligned} \tag{3.47a}$$

$$\begin{aligned}&p_{2e2}x_{1e2}^2 + p_{1e2}x_{1e2} + p_{0e2} - p_{0e1} - p_{1e1}x_{1e2} - p_{2e1}x_{1e2}^2\\&+\frac{1}{\sqrt{x_{1e2}}} + L_{e2} = 0\end{aligned} \tag{3.47b}$$

$$\begin{aligned}&p_{2e2}x_{2e2}^2 + p_{1e2}x_{2e2} + p_{0e2} - p_{0e1} - p_{1e1}x_{2e2} - p_{2e1}x_{2e2}^2\\&+\frac{1}{\sqrt{x_{2e2}}} - L_{e2} = 0\end{aligned} \tag{3.47c}$$

$$4p_{2e2} + 2p_{1e2} + p_{0e2} - p_{0e1} - 2p_{1e1} - 4p_{2e1} + \frac{\sqrt{2}}{2} + L_{e2} = 0 \tag{3.47d}$$

$$2p_{2e2}x_{1e2} + p_{1e2} - p_{1e1} - 2p_{2e1}x_{1e2} - \frac{1}{2x_{1e2}^{1.5}} = 0 \tag{3.47e}$$

$$2p_{2e2}x_{2e2} + p_{1e2} - p_{1e1} - 2p_{2e1}x_{2e2} - \frac{1}{2x_{2e2}^{1.5}} = 0 \tag{3.47f}$$

The maximum absolute error associated with the $\widetilde{\varepsilon}^{\,\mathrm{m2}}_{(x_3,2)}(x)$ approximation is $0.2698990756\cdot 10^{-3}$ $(= 0.2698990756\cdot 10^{-3})$ and occurs at $x_{1e2} = 1.5678139670$ and $x_{2e2} = 1.8492020070$

IV Step By means of (3.46) it is possible to express $\dfrac{1}{\sqrt{x}}$ within $(x_3,2)$ as:

$$\begin{aligned}&P_2{}^{\mathrm{m2b}}_{(x_3,2)}(x) = P_2{}^{\mathrm{m2}}_{(x_3,2)}(x) - \widetilde{\varepsilon}^{\,\mathrm{m2}}_{(x_3,2)}(x)\\&= 1.4422332850000 - 0.5659594057000\cdot x + 0.0992655515500\cdot x^2\end{aligned} \tag{3.48}$$

In conclusion, the approximation of $\dfrac{1}{\sqrt{x}}$ over the [1 , 2) is expressed as:

$$P_2{}^{\mathrm{m2}}(x) = \begin{cases} P_2{}^{\mathrm{m2}}_{[1,x_3]}(x) \text{ if } x\in[1,x_3] \\ P_2{}^{\mathrm{m2b}}_{(x_3,2)}(x) \text{ if } x\in(x_3,2)\end{cases} \tag{3.49}$$

To evaluate the accuracy of the $P_2{}^{\mathrm{m2}}$ solution, the following error functions are introduced:

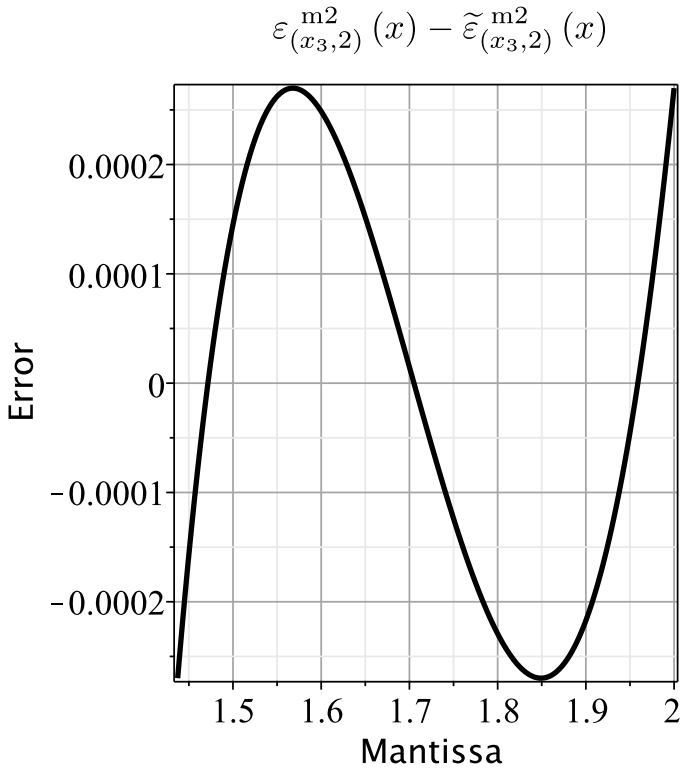


Figure 3.21: Approximation error

$$\varepsilon^{\mathrm{m2b}}_{(x_3,2)}(x) \equiv P_2{}^{\mathrm{m2b}}_{(x_3,2)}(x) - \frac{1}{\sqrt{x}} \text{ for } x \in (x_3, 2) \tag{3.50a}$$

$$\varepsilon^{\mathrm{m2}}(x) \equiv \begin{cases} \varepsilon^{\mathrm{m2}}_{[1,x_3]}(x) \text{ for } x \in [1, x_3] \\ \varepsilon^{\mathrm{m2b}}_{(x_3,2)}(x) \text{ for } x \in (x_3, 2) \end{cases}$$

$$= P_2{}^{\mathrm{m2}}(x) - \frac{1}{\sqrt{x}} \tag{3.50b}$$

$$\varepsilon^{\mathrm{m2}}_{\%}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m2}}(x)}{\frac{1}{\sqrt{x}}} \tag{3.50c}$$

The plots of $\varepsilon^{\mathrm{m2}}(x)$ and $\varepsilon^{\mathrm{m2}}_{\%}(x)$ are respectively shown in Figures 3.22 and 3.23.

Solution refinement by means of optimal approximation domain division: four intervals case

In this section the approximation domain is split in the following four intervals:

- $[1\ ,\ x_4)$

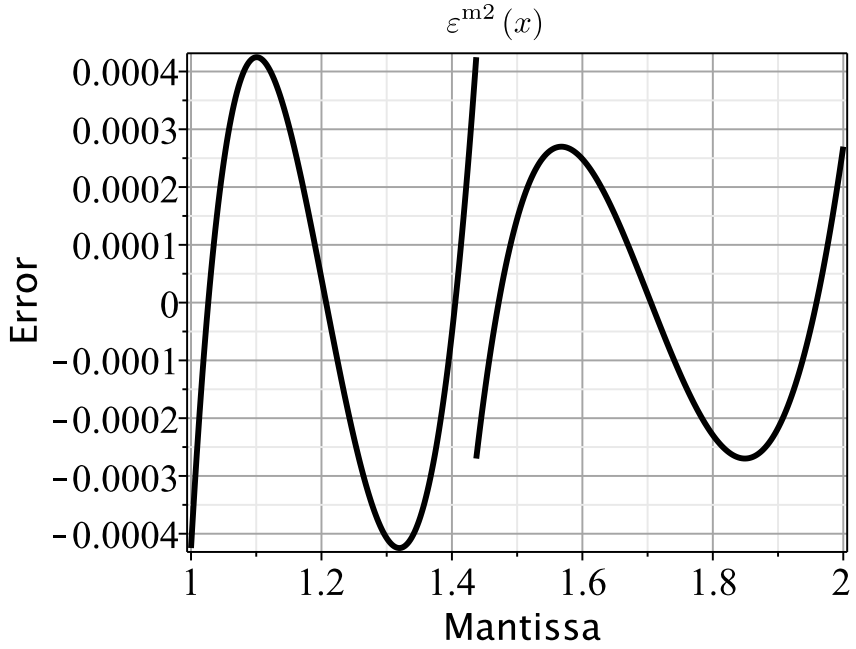


Figure 3.22: Plot of $\varepsilon^{\mathrm{m2}}(x)$ (see (3.50b))

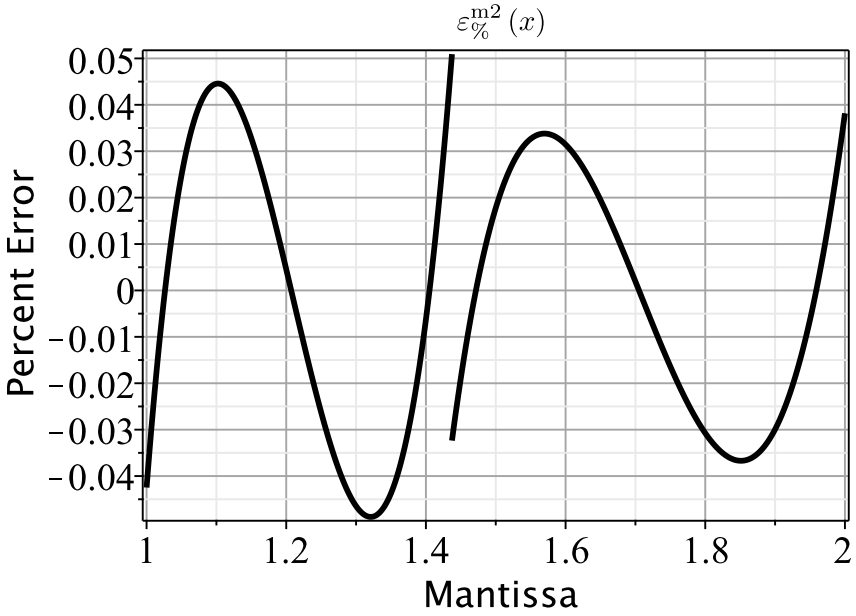


Figure 3.23: Plot of $\varepsilon_{\%}^{\mathrm{m2}}(x)$ (see (3.50c))

- $[x_4\ ,\ x_3)$
- $[x_3\ ,\ x_5)$
- $[x_5\ ,\ 2)$

where $x_3 = 1.4373074411960$ has been computed in the previous sections, while $x_4 = 1.2042247440$ and $x_5 = 1.7017395140$ are obtained by solving equations (3.51a) and (3.51b), respectively.

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\varepsilon^{\mathrm{m2}}_{[1,x_3]}(x) = 0 \tag{3.51a}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\varepsilon^{\mathrm{m2b}}_{(x_3,2)}(x) = 0 \tag{3.51b}$$

The Chebyshev criterion has been applied within the subintervals $[1\ ,\ x_4)$, $[x_4\ ,\ x_3)$, $[x_3\ ,\ x_5)$ and $[x_5\ ,\ 2)$ with the following result:

$$P_2{}^{\mathrm{m3}}(x) = \begin{cases} 1.7907676030000 - 1.0872271400000 \cdot x + 0.2963997462000 \cdot x^2 \text{ if } x \in [1\ ,\ x_4) \\ 1.6354291120000 - 0.8282541110000 \cdot x + 0.1883906567000 \cdot x^2 \text{ if } x \in [x_4\ ,\ x_3) \\ 1.4999205600000 - 0.6390428680000 \cdot x + 0.1223012800000 \cdot x^2 \text{ if } x \in [x_3\ ,\ x_5) \\ 1.3809636440000 - 0.4987944640000 \cdot x + 0.0809406000000 \cdot x^2 \text{ if } x \in [x_5\ ,\ 2) \end{cases} \tag{3.52}$$

To evaluate the accuracy of the approximation $P_2{}^{\mathrm{m3}}(x)$ the following error functions are defined:

$$\varepsilon^{\mathrm{m3}}(x) \equiv P_2{}^{\mathrm{m3}}(x) - \frac{1}{\sqrt{x}} \tag{3.53a}$$

$$\varepsilon^{\mathrm{m3}}_{\%}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m3}}(x)}{\frac{1}{\sqrt{x}}} \tag{3.53b}$$

Figures 3.53a and 3.53b depict the plots of $\varepsilon^{\mathrm{m3}}(x)$ and $\varepsilon^{\mathrm{m3}}_{\%}(x)$, respectively.

Solution refinement: approximation domain division in eight intervals

In this section the approximation domain is split in the following eight intervals of equal width:

- [1 , 1.125)
- [1.125 , 1.250)
- [1.250 , 1.375)
- [1.375 , 1.500)
- [1.500 , 1.625)
- [1.625 , 1.750)
- [1.750 , 1.875)
- [1.875 , 2.000)

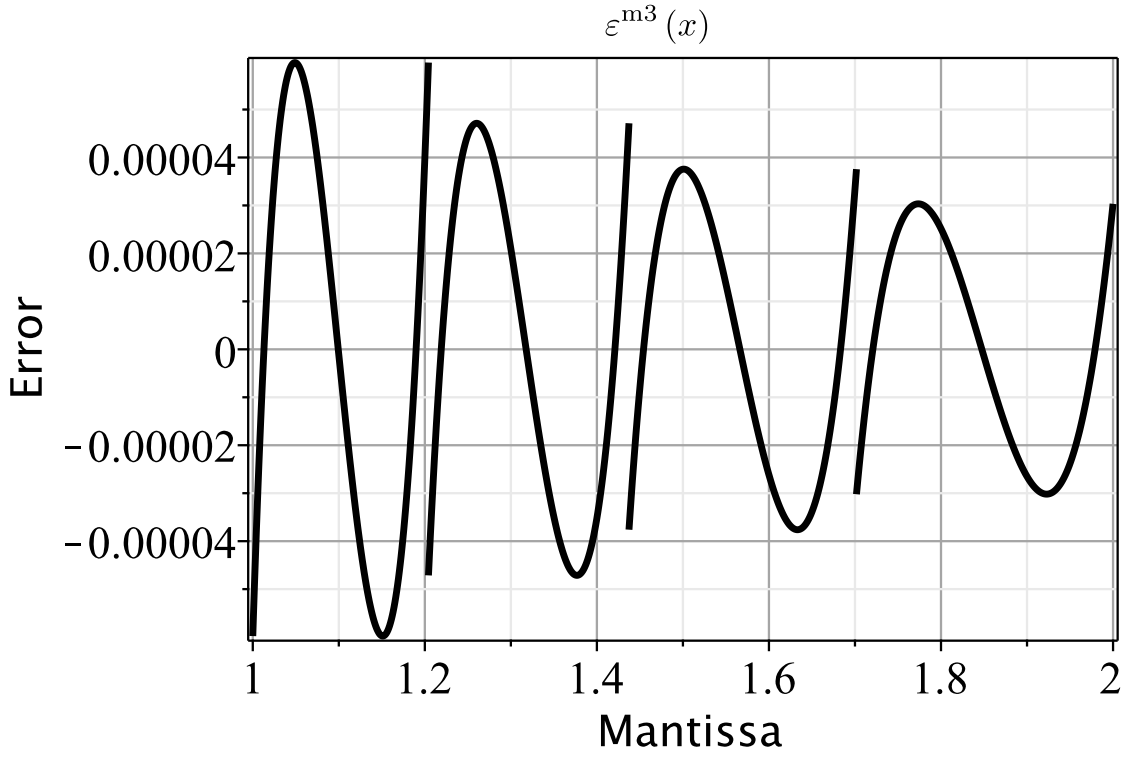


Figure 3.24: Plot of $\varepsilon^{\mathrm{m3}}(x)$ (see (3.53a))

As shown in Figure 3.26, this domain division bears the advantage that only the mantissa most 3 significant fractional bits are required to identify the interval. The approximating polynomial, are obtained again by applying the Chebyshev criterion in the selected interval ranges. The result is summarized in (3.54).

$$P_2{}^{\mathrm{m4}}(x) = \begin{cases} 0.323281455440467 \cdot x^2 - 1.14425291279838 \cdot x + 1.82095596689838 \text{ if } x \in [1\ ,\ 1.125) \\ 0.244649955973864 \cdot x^2 - 0.967930660701772 \cdot x + 1.72208544750509 \text{ if } x \in [1.125\ ,\ 1.250) \\ 0.190406394559703 \cdot x^2 - 0.832693255473729 \cdot x + 1.63777638541345 \text{ if } x \in [1.250\ ,\ 1.375) \\ 0.151621424591140 \cdot x^2 - 0.726275989198446 \cdot x + 1.56476772704897 \text{ if } x \in [1.375\ ,\ 1.500) \\ 0.123059465297923 \cdot x^2 - 0.640753029022596 \cdot x + 1.50073831985172 \text{ if } x \in [1.500\ ,\ 1.625) \\ 0.101499793039017 \cdot x^2 - 0.570797589645627 \cdot x + 1.44398467222478 \text{ if } x \in [1.625\ ,\ 1.750) \\ 0.0848803531284505 \cdot x^2 - 0.512710777288178 \cdot x + 1.39322434205260 \text{ if } x \in [1.750\ ,\ 1.875) \\ 0.0718354962811628 \cdot x^2 - 0.463852063024561 \cdot x + 1.34747080850443 \text{ if } x \in [1.875\ ,\ 2.000) \end{cases} \tag{3.54}$$

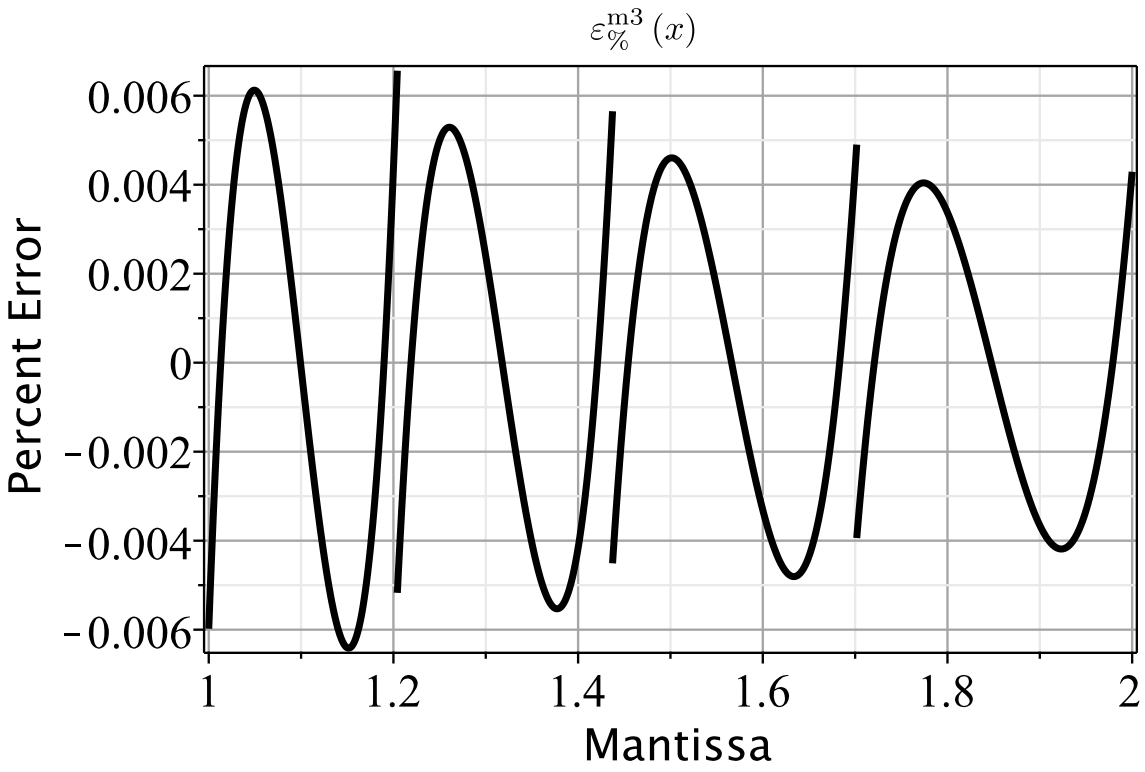


Figure 3.25: Plot of $\varepsilon_{\%}^{\mathrm{m3}}(x)$ (see (3.53b))

Interval 1:	1.000000000000000000000	to	1.000111111111111111111
Interval 2:	1.001000000000000000000	to	1.001111111111111111111
Interval 3:	1.010000000000000000000	to	1.010111111111111111111
Interval 4:	1.011000000000000000000	to	1.011111111111111111111
Interval 5:	1.100000000000000000000	to	1.100111111111111111111
Interval 6:	1.101000000000000000000	to	1.101111111111111111111
Interval 7:	1.110000000000000000000	to	1.110111111111111111111
Interval 8:	1.111000000000000000000	to	1.111111111111111111111

The interval to which the input mantissa belongs to is infered by reading the 3 most significant fractional bits

Input mantissa: 1.xxx xxxxxxxxxxxxxxxxxxxxx

Figure 3.26: Approximation domain ranges

To evaluate the accuracy of the approximation ${P_2}^{\mathrm{m4}}(x)$ the following error functions are defined:

$$\varepsilon^{\mathrm{m4}}(x) \equiv {P_2}^{\mathrm{m4}}(x) - \frac{1}{\sqrt{x}} \tag{3.55a}$$

$$\varepsilon_{\%}^{\mathrm{m4}}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m4}}(x)}{\frac{1}{\sqrt{x}}} \tag{3.55b}$$

The plots of the functions $\varepsilon^{\mathrm{m4}}(x)$ and $\varepsilon_{\%}^{\mathrm{m4}}(x)$ are depicted in Figures 3.27 and 3.28, respectively.

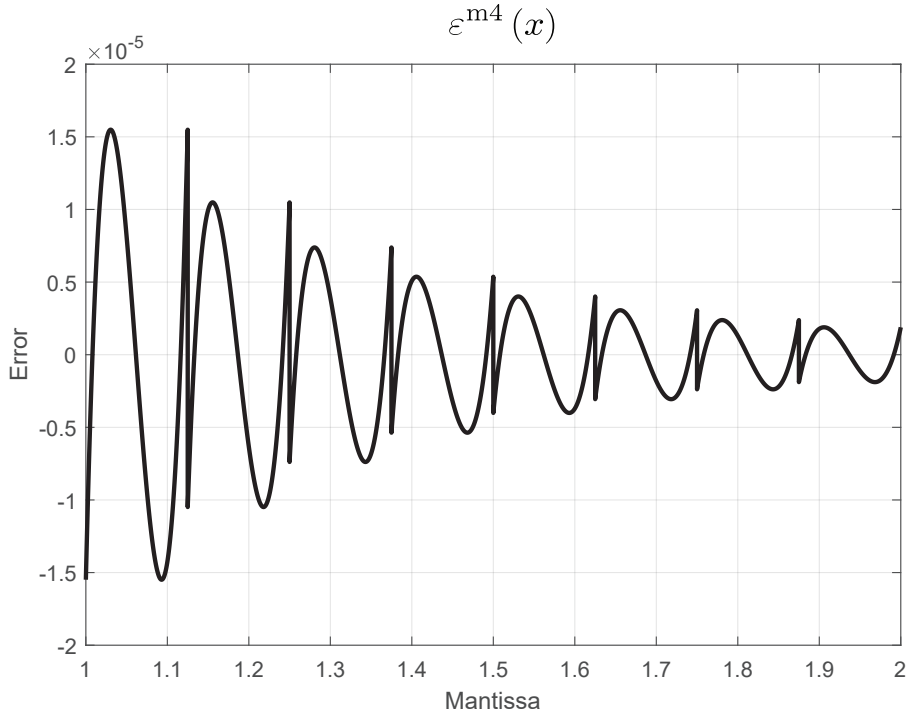


Figure 3.27: Plot of $\varepsilon^{\mathrm{m4}}(x)$ (see (3.55a))

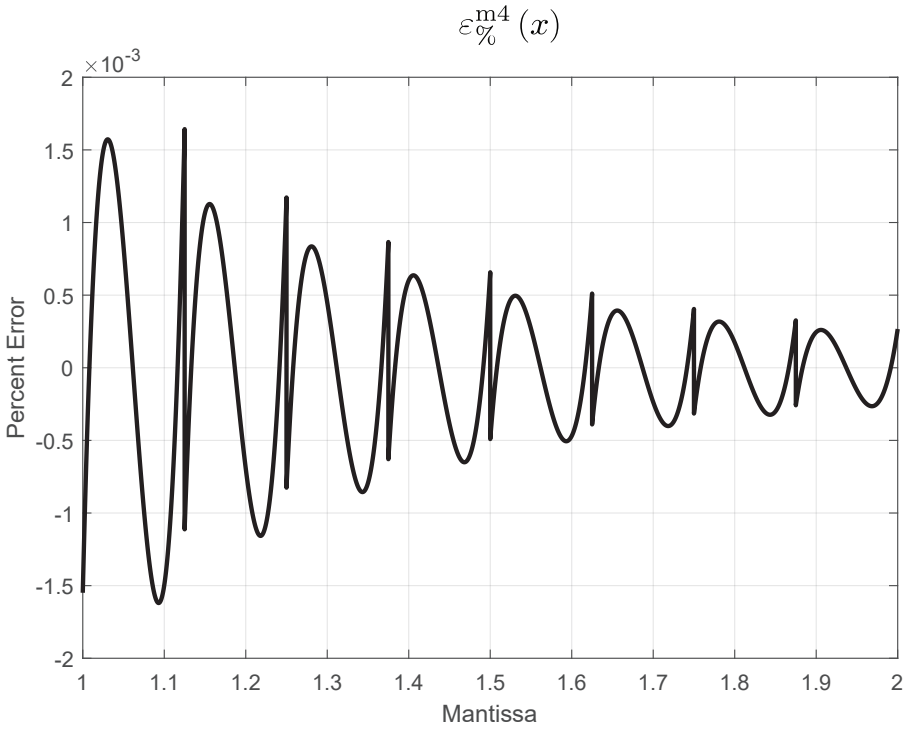


Figure 3.28: Plot of $\varepsilon_{\%}^{\mathrm{m4}}(x)$ (see (3.55b))

3.4.3 Proposed cubic approximation of $\frac{1}{\sqrt{x}}$

The polynomial coefficients are computed by solving the following non linear system obtained from (2.95):

$$p_3 + p_2 + p_1 + p_0 - 1 - L = 0 \tag{3.56a}$$

$$p_3 x_1^3 + p_2 x_1^2 + p_1 x1 + p_0 - \frac{1}{\sqrt{x_1}} + L = 0 \tag{3.56b}$$

$$p_3 x_2^3 + p_2 x_2^2 + p_1 x_2 + p_0 - \frac{1}{\sqrt{x_2}} - L = 0 \tag{3.56c}$$

$$p_3 x_3^3 + p_2 x_3^2 + p_1 x_3 + p_0 - \frac{1}{\sqrt{x_3}} + L = 0 \tag{3.56d}$$

$$8p_3 + 4p_2 + 2p_1 + p_0 - \frac{\sqrt{2}}{2} - L = 0 \tag{3.56e}$$

$$3p_3 x_1^2 + 2p_2 x_1 + p_1 + \frac{1}{2x_1^{1.5}} = 0 \tag{3.56f}$$

$$3p_3 x_2^2 + 2p_2 x_2 + p_1 + \frac{1}{2x_2^{1.5}} = 0 \tag{3.56g}$$

$$3p_3 x_3^2 + 2p_2 x_3 + p_1 + \frac{1}{2x_3^{1.5}} = 0 \tag{3.56h}$$

The solution of the system is:

- **Max Absolute Error**: $0.4076338043 \cdot 10^{-3}$ ($L = -0.4076338043 \cdot 10^{-3}$)
- **Abscissae of error function stationary points**:
  $x_1 = 1.1265200690000$, $x_2 = 1.4613817440000$, $x_3 = 1.8348363710000$
- **Approximating polynomial**

$$P_3(x) \equiv 1.8491816370000 - 1.3015329320000 \cdot x + 0.5387414784000 \cdot x^2 - 0.0867978175400 \cdot x^3 \tag{3.57}$$

To evaluate the accuracy of the approximation $P_3(x)$ the following error functions are defined:

$$\varepsilon(x) \equiv P_3(x) - \frac{1}{\sqrt{x}} \tag{3.58a}$$

$$\varepsilon_{\%}(x) \equiv 100 \cdot \frac{\varepsilon(x)}{\frac{1}{\sqrt{x}}} \tag{3.58b}$$

- **Error Plots** see Figures 3.29, 3.30

3.5 Polynomial approximation of $\sqrt{x}$

3.5.1 Proposed linear approximation of $\sqrt{x}$

In this section some possibilities of approximately express $\sqrt{x}$ with the first degree polynomial

$$P_1(x) = p_1 \cdot x + p_0 \,. \tag{3.59}$$

are explored. The methodology herein applied to deduce the optimal polynomial is the same of the previous sections. For the sake of brevity only final results and error plot are reported.

Approximation over a single interval

$$P_1^{\mathrm{m0}}(x) = 0.4142135620 \cdot x + 0.5946699146 \tag{3.60}$$

To evaluate the accuracy of the $P_1^{\mathrm{m0}}(x)$ solution, the following error functions are introduced:

$$\varepsilon(x) \equiv P_1^{\mathrm{m0}}(x) - \sqrt{x} \tag{3.61a}$$

$$\varepsilon_{\%}(x) \equiv 100 \cdot \frac{\varepsilon(x)}{\sqrt{x}} \tag{3.61b}$$

The plots of the functions $\varepsilon(x)$ and $\varepsilon_{\%}(x)$ are depicted in Figures 3.31 and 3.32, respectively. The abscissa of error function stationary points is $x_r = 1.457106784$.

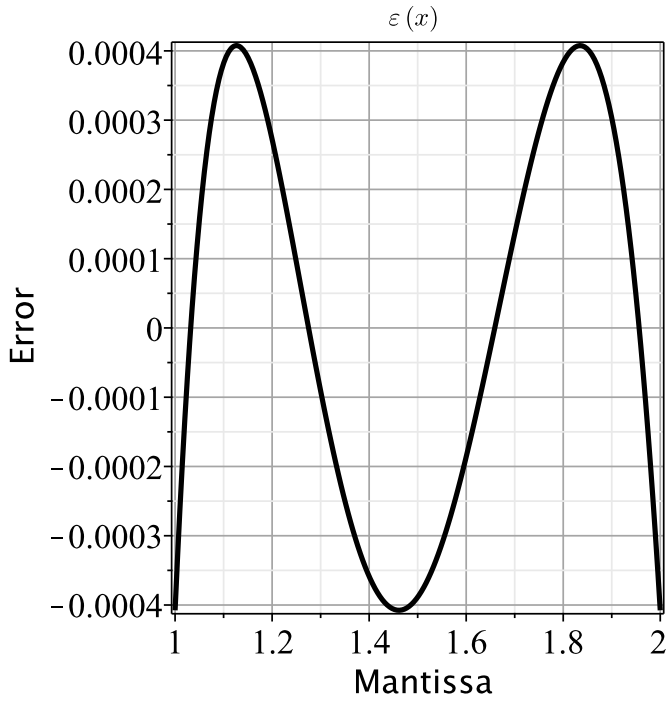


Figure 3.29: Error for 3rd degree polynomial approximating $\frac{1}{\sqrt{x}}$

Approximation over two intervals

$$P_1^{\mathrm{m1}}(x) = \begin{cases} 0.4530818383 \cdot x + 0.5493474291 \text{ if } x \in [1\ ,\ x_r) \\ 0.3814871395 \cdot x + 0.6532846847 \text{ if } x \in [x_r\ ,\ 2) \end{cases} \tag{3.62}$$

To evaluate the accuracy of the $P_1^{\mathrm{m1}}(x)$ solution, the following error functions are introduced:

$$\varepsilon^{\mathrm{m1}}(x) \equiv P_1^{\mathrm{m1}}(x) - \sqrt{x} \tag{3.63a}$$

$$\varepsilon_{\%}^{\mathrm{m1}}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m1}}(x)}{\sqrt{x}} \tag{3.63b}$$

The plots of the functions $\varepsilon^{\mathrm{m1}}(x)$ and $\varepsilon_{\%}^{\mathrm{m1}}(x)$ are depicted in Figures 3.33 and 3.34, respectively.

3.5.2 Proposed quadratic approximation of $\sqrt{x}$

In this section some possibilities of approximately express $\sqrt{x}$ with the second degree polynomial

$$P_1(x) = p_2 \cdot x^2 + p_1 \cdot x + p_0\ . \tag{3.64}$$

are explored.

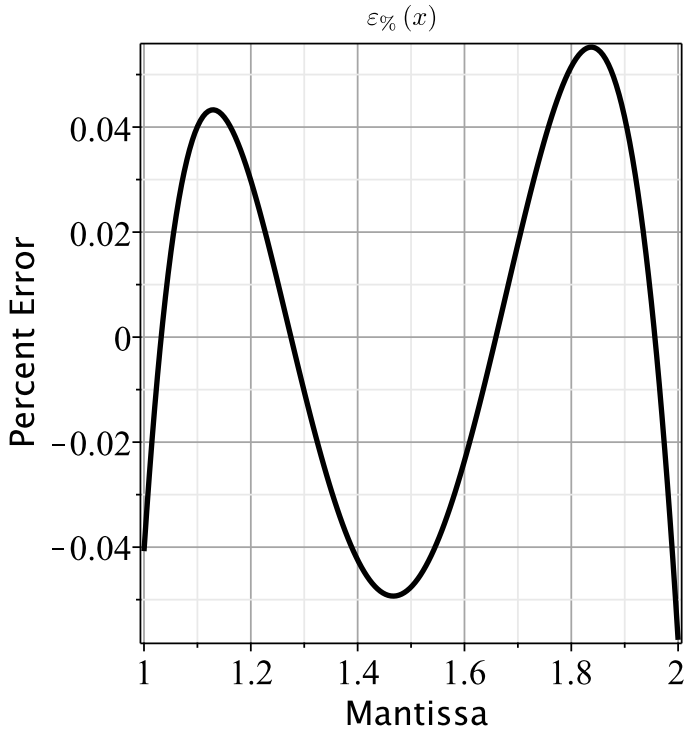


Figure 3.30: Percent error for 3rd degree polynomial approximating $\frac{1}{\sqrt{x}}$

Approximation over a single interval

$$P_2^{\mathrm{m0}}(x) = 0.4456804564 + 0.6262821264 \cdot x - 0.07119874601 \cdot x^2 \tag{3.65}$$

To evaluate the accuracy of the $P_2^{\mathrm{m0}}(x)$ solution, the following error functions are introduced:

$$\varepsilon^{\mathrm{m0}}(x) \equiv P_2^{\mathrm{m0}}(x) - \sqrt{x} \tag{3.66a}$$

$$\varepsilon_{\%}^{\mathrm{m0}}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m0}}(x)}{\sqrt{x}} \tag{3.66b}$$

The plots of the functions $\varepsilon^{\mathrm{m0}}(x)$ and $\varepsilon_{\%}^{\mathrm{m0}}(x)$ are depicted in Figures 3.35 and 3.36, respectively.

Solution refinement by means of optimal approximation domain division: two intervals case

$$P_2^{\mathrm{m2}}(x) = \begin{cases} 0.411805165117755 + 0.681462409969314 \cdot x - 0.0931546635295981 \cdot x^2 \text{ if } x \in [1\ ,\ x_3) \\ 0.489746597412591 + 0.573368012561721 \cdot x - 0.0555877048081109 \cdot x^2 \text{ if } x \in [x_3\ ,\ 2) \end{cases} \tag{3.67}$$

where $x_3 = 1.455319825$

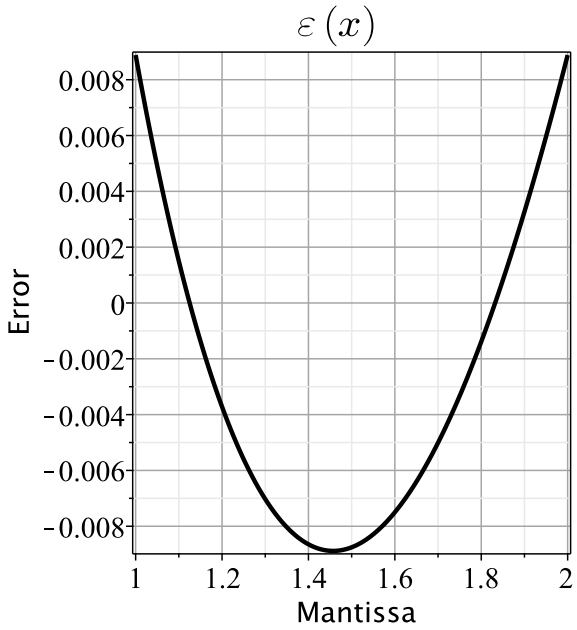


Figure 3.31: Plot of $\varepsilon(x)$ (see (3.61a))

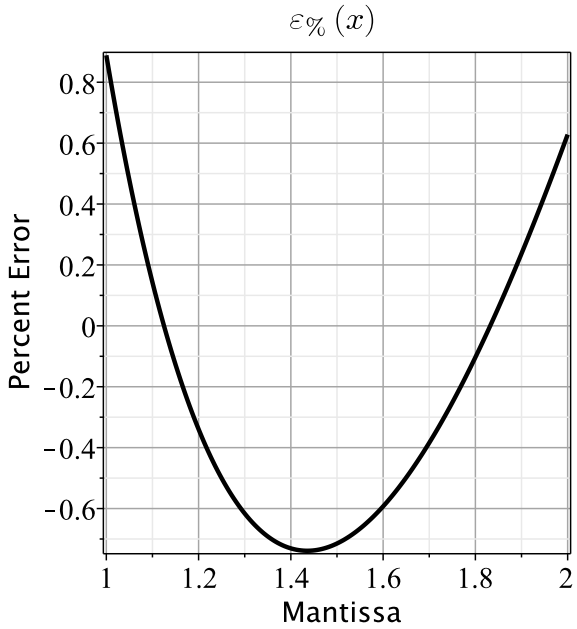


Figure 3.32: Plot of $\varepsilon_{\%}(x)$ (see (3.61b))

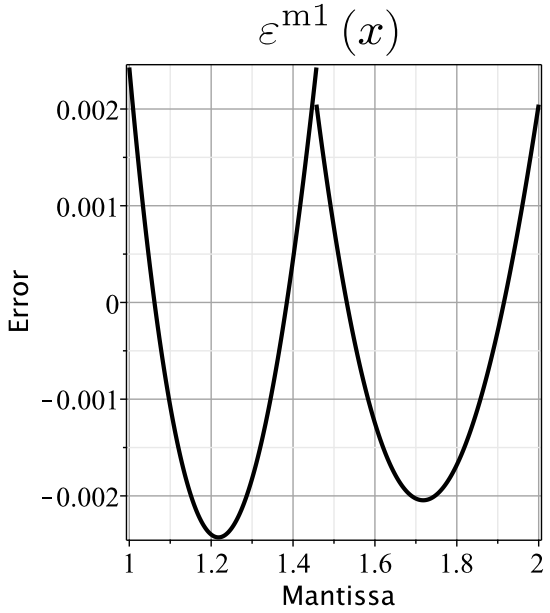


Figure 3.33: Plot of $\varepsilon^{\mathrm{m1}}(x)$ (see (3.63a))

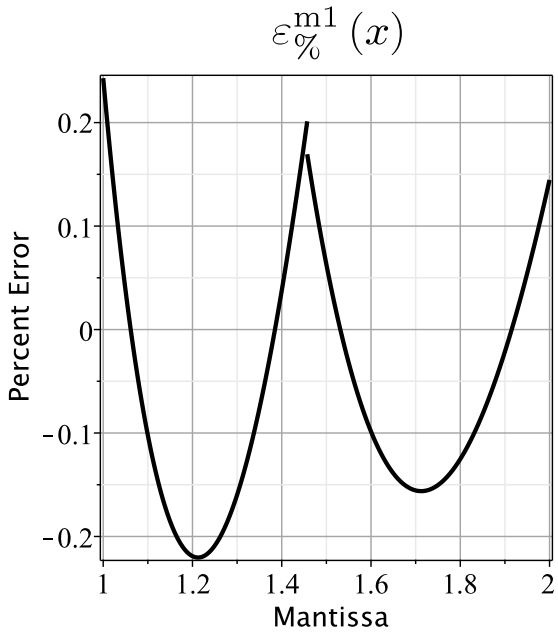


Figure 3.34: Plot of $\varepsilon_{\%}^{\mathrm{m1}}(x)$ (see (3.63b))

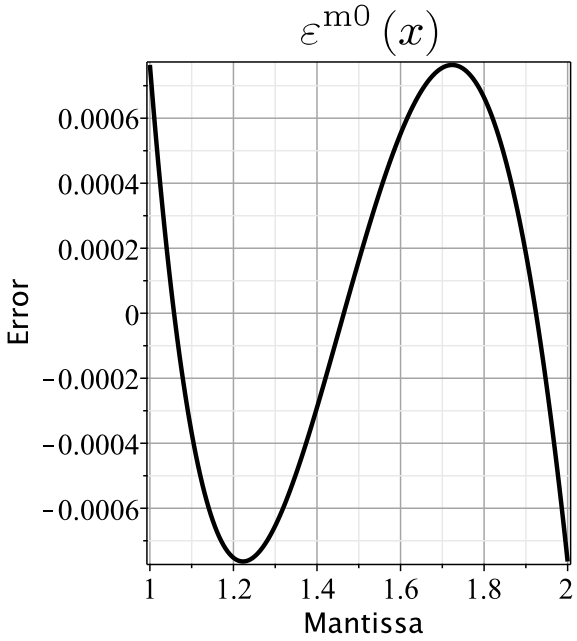


Figure 3.35: Plot of $\varepsilon^{\mathrm{m0}}(x)$ (see (3.66a))

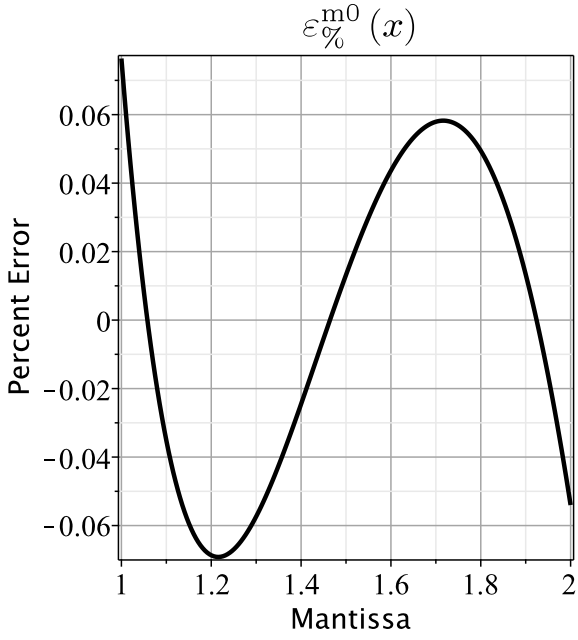


Figure 3.36: Plot of $\varepsilon_{\%}^{\mathrm{m0}}(x)$ (see (3.66b))

To evaluate the accuracy of the $P_2^{\mathrm{m2}}(x)$ solution, the following error functions are introduced:

$$\varepsilon^{\mathrm{m2}}(x) \equiv P_2^{\mathrm{m2}}(x) - \sqrt{x} \tag{3.68a}$$

$$\varepsilon_{\%}^{\mathrm{m2}}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m2}}(x)}{\sqrt{x}} \tag{3.68b}$$

The plots of the functions $\varepsilon^{\mathrm{m2}}(x)$ and $\varepsilon_{\%}^{\mathrm{m2}}(x)$ are depicted in Figures 3.37 and 3.38, respectively.

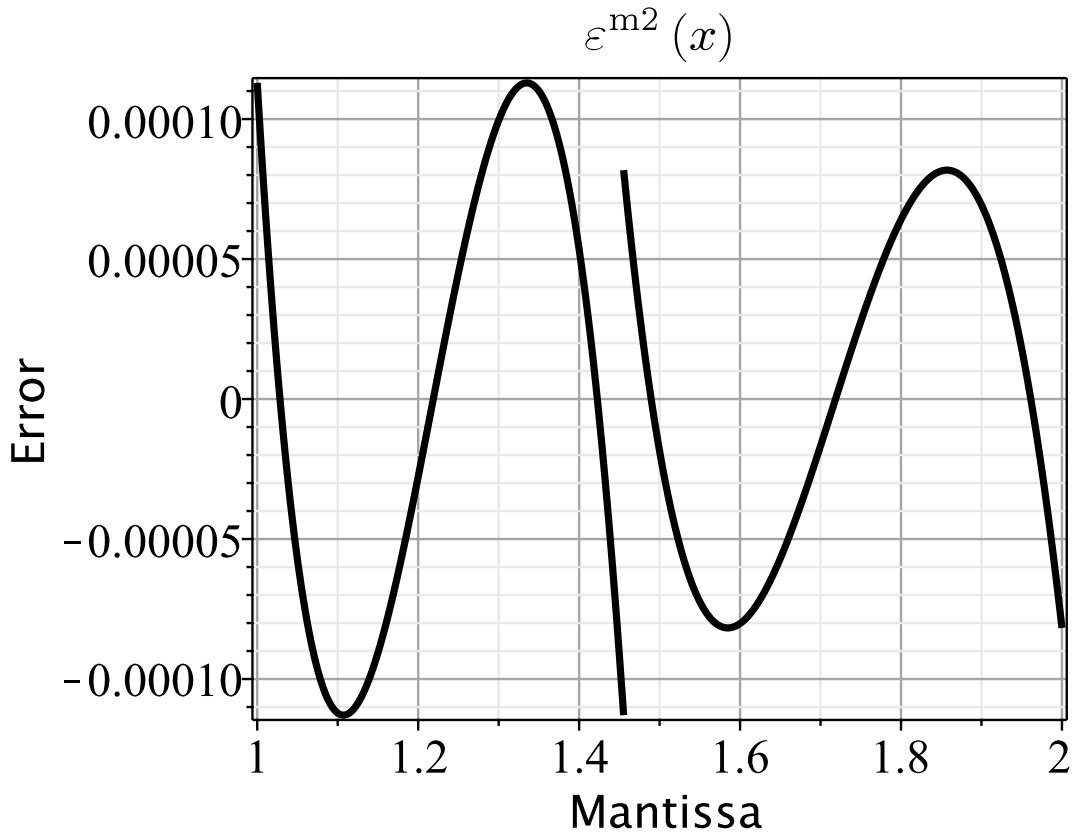


Figure 3.37: Plot of $\varepsilon^{\mathrm{m2}}(x)$ (see (3.68a))

Solution refinement by means of optimal approximation domain division: four intervals case

$$P_2^{\mathrm{m3}}(x) = \begin{cases} 0.393816401927113 + 0.713736157019602 \cdot x - 0.107537120915299 \cdot x^2 \text{ if } x \in [1\ ,\ x_4) \\ 0.432550655992538 + 0.649886461756243 \cdot x - 0.0812056078196054 \cdot x^2 \text{ if } x \in [x_4\ ,\ x_3) \\ 0.471422932847082 + 0.596344109832261 \cdot x - 0.0627571840422756 \cdot x^2 \text{ if } x \in [x_3\ ,\ x_5) \\ 0.510423621696508 + 0.550811506353596 \cdot x - 0.0494606429995583 \cdot x^2 \text{ if } x \in [x_5\ ,\ 2) \end{cases} \tag{3.69}$$

where $x_3 = 1.455319825$, $x_4 = 1.216569547$ and $x_5 = 1.716409253$.

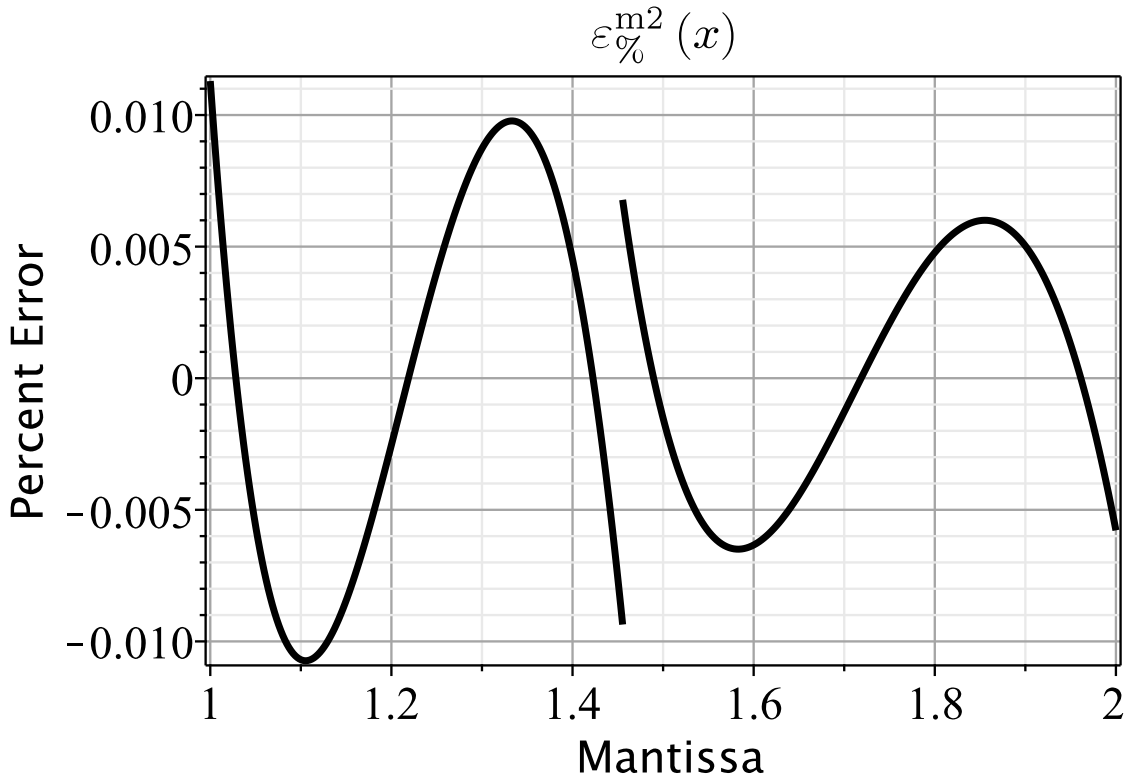


Figure 3.38: Plot of $\varepsilon_{\%}^{\mathrm{m2}}(x)$ (see (3.68b))

To evaluate the accuracy of the $P_2^{\mathrm{m3}}(x)$ solution, the following error functions are introduced:

$$\varepsilon^{\mathrm{m3}}(x) \equiv P_2^{\mathrm{m3}}(x) - \sqrt{x} \tag{3.70a}$$

$$\varepsilon_{\%}^{\mathrm{m3}}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m3}}(x)}{\sqrt{x}} \tag{3.70b}$$

The plots of the functions $\varepsilon^{\mathrm{m3}}(x)$ and $\varepsilon_{\%}^{\mathrm{m3}}(x)$ are depicted in Figures 3.39 and 3.40, respectively.

Solution refinement: approximation domain division in eight intervals

$$P_2{}^{\mathrm{m4}}(x) = \begin{cases} 0.386199363782058 + 0.728092916008656 \cdot x - 0.114288994119807 \cdot x^2 \text{ if } x \in [1\ ,\ 1.125) \\ 0.408357584404978 + 0.688615122160828 \cdot x - 0.0967007905987910 \cdot x^2 \text{ if } x \in [1.125\ ,\ 1.250) \\ 0.429367638600846 + 0.654940076151361 \cdot x - 0.0832043579057413 \cdot x^2 \text{ if } x \in [1.250\ ,\ 1.375) \\ 0.449392142793619 + 0.625771343536133 \cdot x - 0.0725803681584724 \cdot x^2 \text{ if } x \in [1.375\ ,\ 1.500) \\ 0.468558480180898 + 0.600185150754720 \cdot x - 0.0640400371487302 \cdot x^2 \text{ if } x \in [1.500\ ,\ 1.625) \\ 0.486968662790052 + 0.577502999033281 \cdot x - 0.0570528286486680 \cdot x^2 \text{ if } x \in [1.625\ ,\ 1.750) \\ 0.504705926984705 + 0.557213746447617 \cdot x - 0.0512501108669738 \cdot x^2 \text{ if } x \in [1.750\ ,\ 1.875) \\ 0.521839235812598 + 0.538924012154719 \cdot x - 0.0463686070804121 \cdot x^2 \text{ if } x \in [1.875\ ,\ 2.000) \end{cases} \tag{3.71}$$

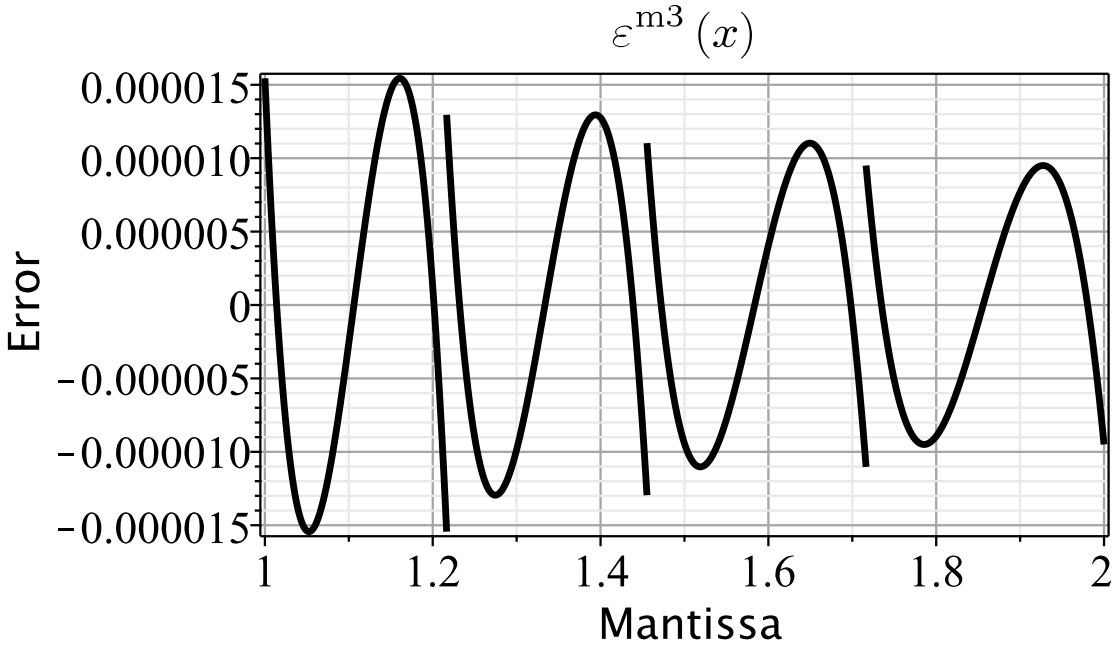


Figure 3.39: Plot of $\varepsilon^{\mathrm{m3}}(x)$ (see (3.70a))

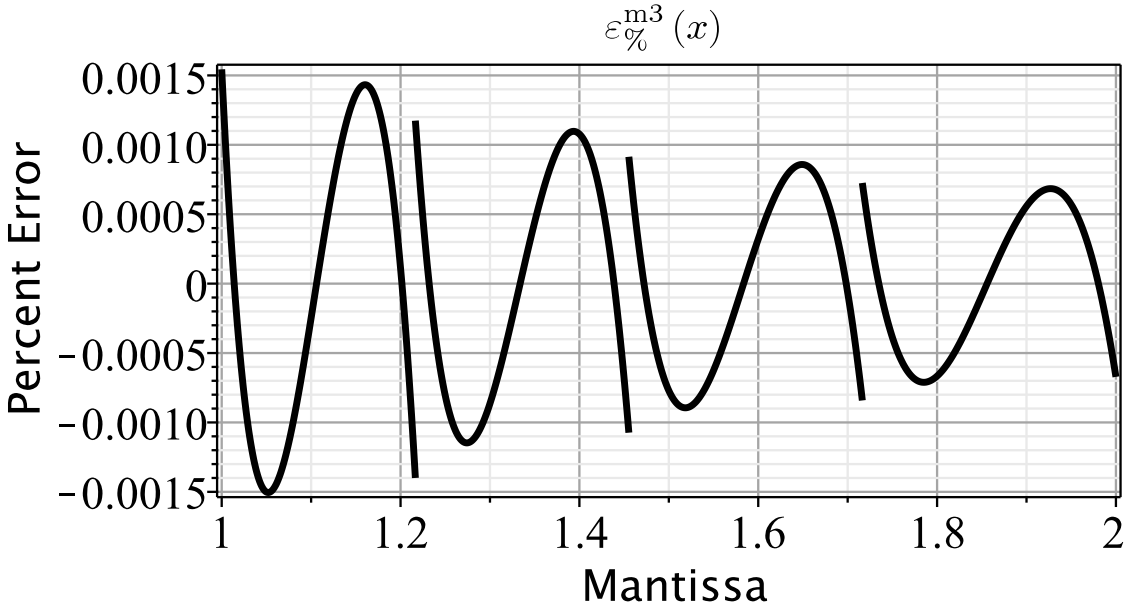


Figure 3.40: Plot of $\varepsilon_{\%}^{\mathrm{m3}}(x)$ (see (3.70b))

To evaluate the accuracy of the approximation ${P_2}^{\mathrm{m4}}(x)$ the following error functions are defined:

$$\varepsilon^{\mathrm{m4}}(x) \equiv {P_2}^{\mathrm{m4}}(x) - \sqrt{x} \tag{3.72a}$$

$$\varepsilon_{\%}^{\mathrm{m4}}(x) \equiv 100 \cdot \frac{\varepsilon^{\mathrm{m4}}(x)}{\sqrt{x}} \tag{3.72b}$$

The plots of the functions $\varepsilon^{\mathrm{m4}}(x)$ and $\varepsilon_{\%}^{\mathrm{m4}}(x)$ are depicted in Figures (3.41) and (3.42), respectively.

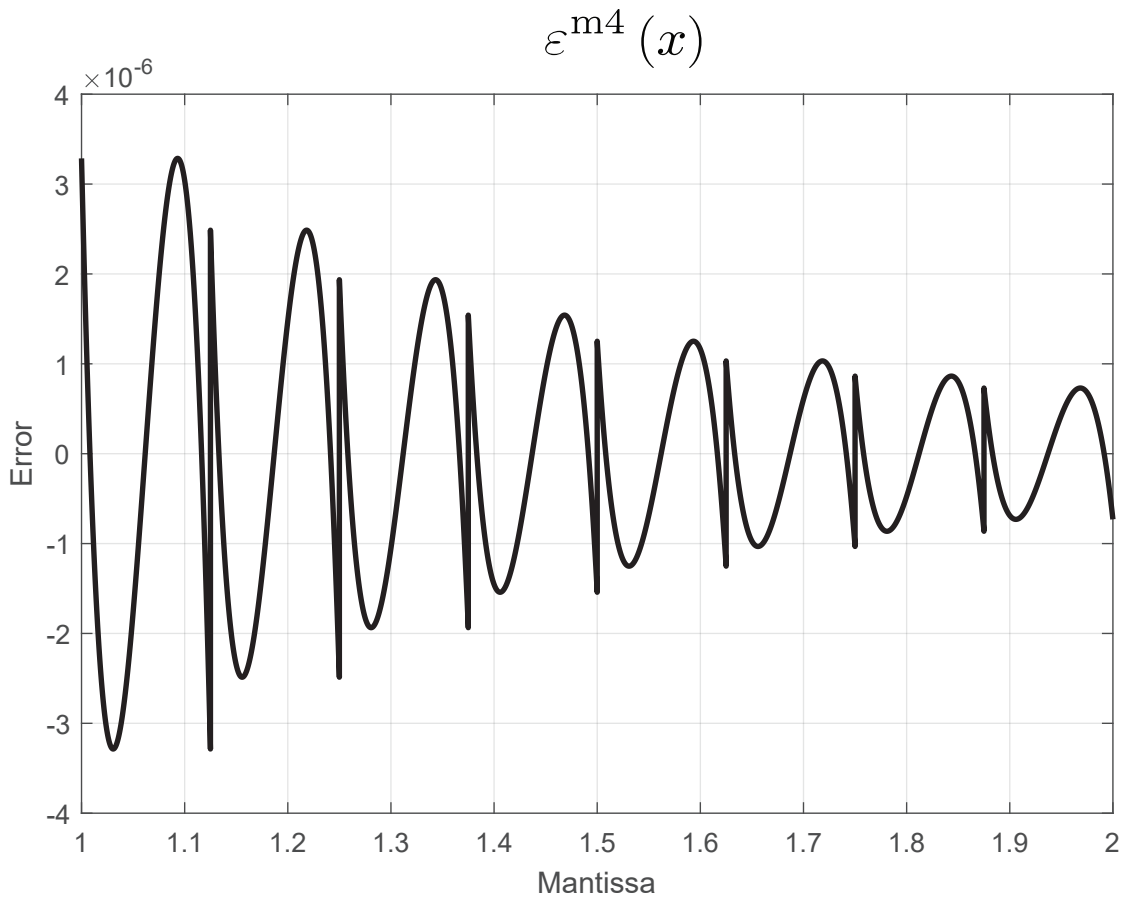


Figure 3.41: Plot of $\varepsilon^{\mathrm{m4}}(x)$ (see (3.72a))

3.6 Summary of polynomial approximation accuracy for both ISQRT and SQRT

It is clear that accuracy increases with the degree of the approximating polynomial as well as the number of sub-intervals. The Table 3.1 summarizes such influence.

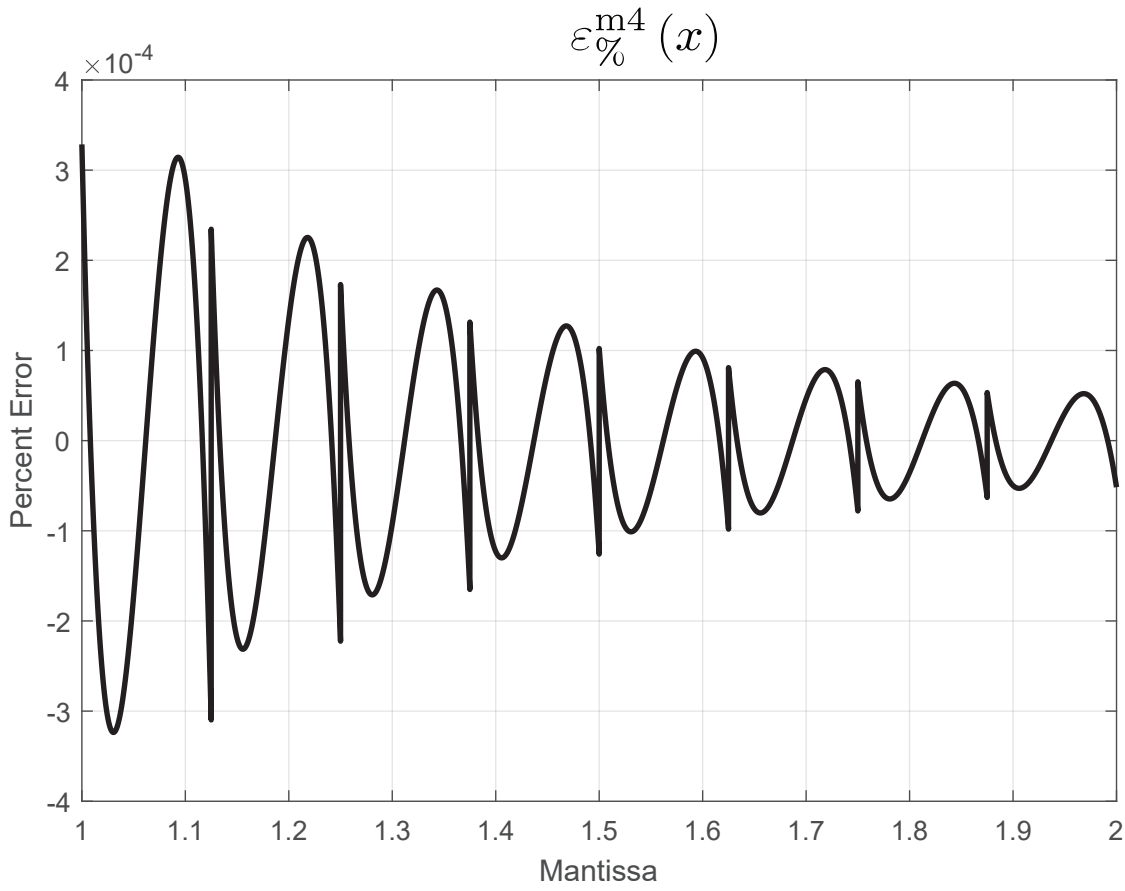


Figure 3.42: Plots of $\varepsilon_{\%}^{\mathrm{m4}}(x)$ (see (3.72b))

Table 3.1: Proposed method: Computation of maximum relative error (MRE) within the range [1 , 2), n: Polynomial degree , N: number of sub-intervals of $\log_2(N)$ amplitude

$1/\sqrt{x}$			$\sqrt{x}$		
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

Chapter 4

Simulation Results and FPGA Implementation

Ut sementem feceris ita metes.
As you sow, so shall you reap.

Cicero, *De Oratore*

4.1 Sensor Fusion for IMU data: Numerical Tests

This section reports the results of different numerical test for the comparison between the original algorithm and the proposed refinements. The `IMU` sensor test data (see Figure 4.1), as well as `Matlab` implementation of Madgwick & Mahony filters, have been downloaded from `https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/`.
In particular, Table 4.1 summarizes the different execution times and the maximum absolute values of E_m. For the purpose of completeness, the numerical test included the results obtained from `Matlab` function `ahrsfilter`. In order to assess the influence on execution time of refinement iteration steps, different configurations of the proposed algorithm have been tested.
For the different algorithms, the plots of E_m versus time are reported in Figures 4.2, 4.3 and 4.4. The tests have been carried out on a Windows 10 PC with an Intel i9-9900K CPU 3.6 GHz and 32 Gb RAM (DDR4)

4.2 FPGA implementation of the modified Madgwick sensor fusion algorithm

The architecture depicted in Figure 4.5 is proposed to implement both Madgwick and the modified Madgwick sensor fusion algorithms. The pinout of this architecture is reported in Figure 4.6.
With reference to Figure 4.5 the architecture main components are:

- `MAC16` - vector scalar product, vector by scalar operations and matrix products.

Table 4.1: Comparison between AHRS 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

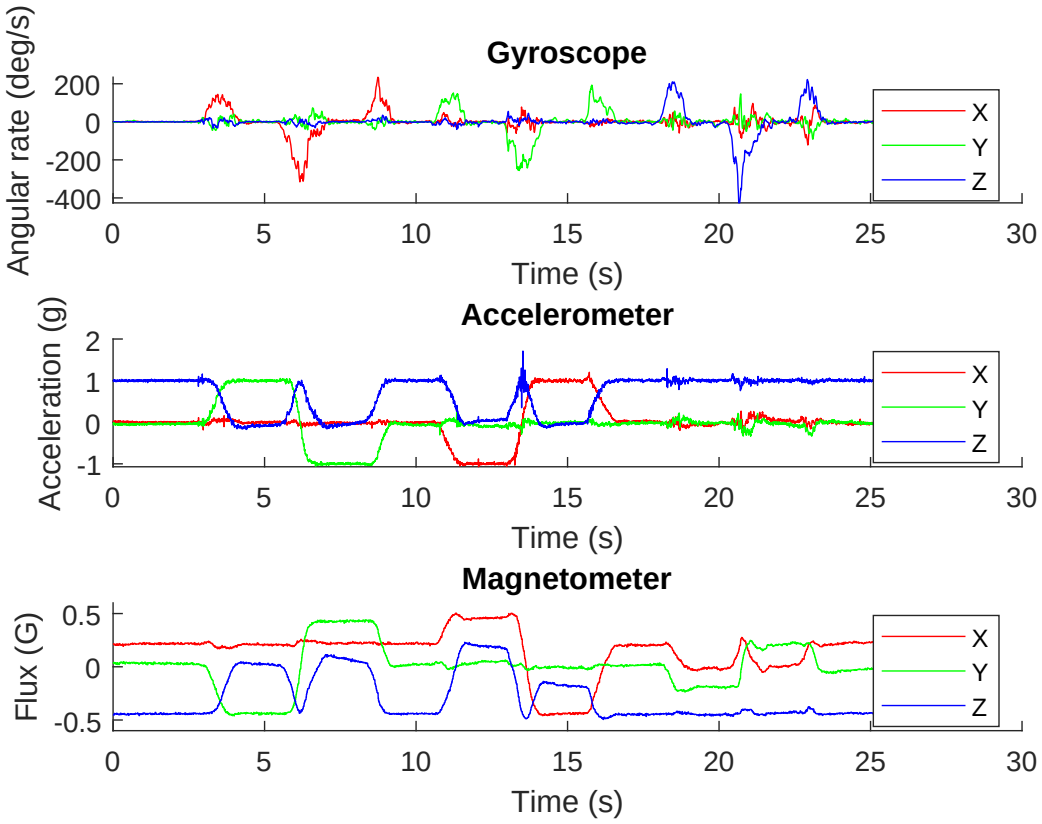


Figure 4.1: IMU Data

- `ADDSUB16` - vectors additions.
- `isqrtSqrt` - computation of both ISQRT and SQRT.
- `qDotReg` - this register stores the quaternion time derivative.
- `tempReg` - register to hold temporary data.
- `MatrixA` - stores the content of matrix **A** (see Section 3.2). The finite state machine of this component is depicted in Figure 4.9.
- `quaternion_matrices` - stores quaternion dependent matrices (e.g. **E**,**L**,**K**). The finite state machine of this component is depicted in Figure 4.10.
- `constants_reg` - stores algorithm constants.The finite state machine of this component is depicted in Figure 4.8.

The proposed architecture has been simulated with the `Modelsim` tool, in particular a virtual IMU sensor (`vimu.vhd`) has been created in `VHDL`. The virtual IMU unit reads the data from a text file and

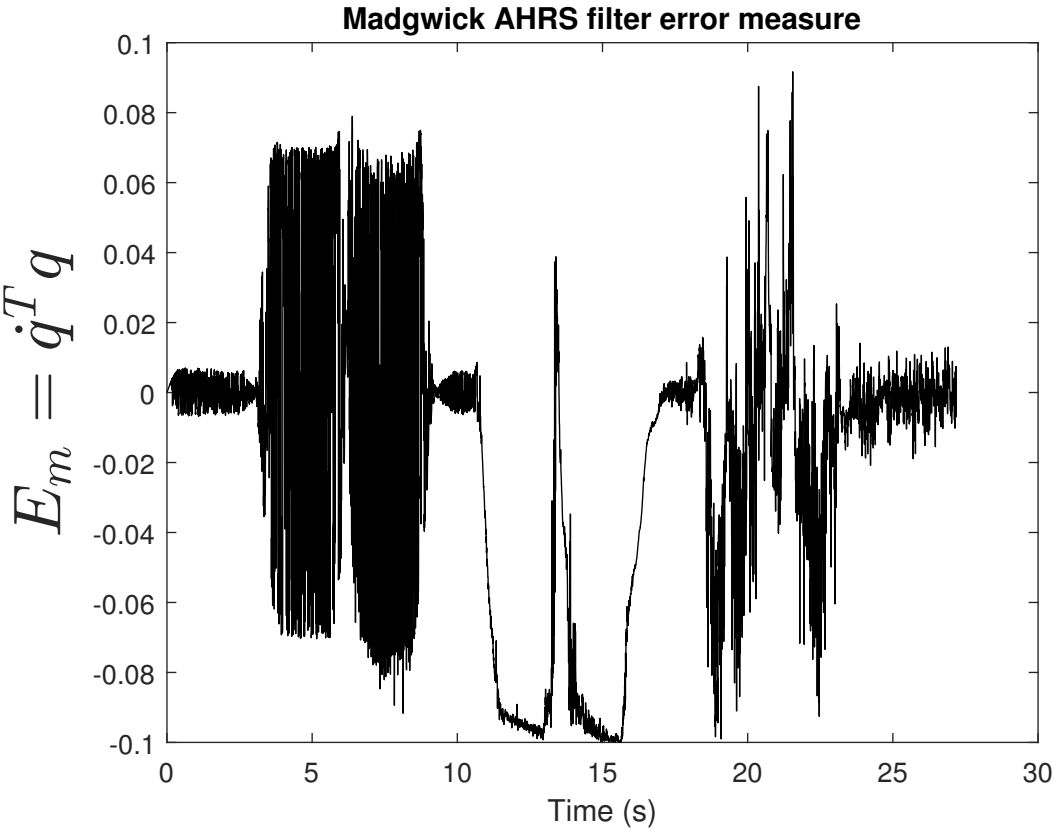


Figure 4.2: Error analysis for the Madgwick algorithm

passes them when requested to the sensor fusion architecture (`sensor_fusion.vhd`). The numerical values of the quaternion elements computed are shown in Figure 4.7

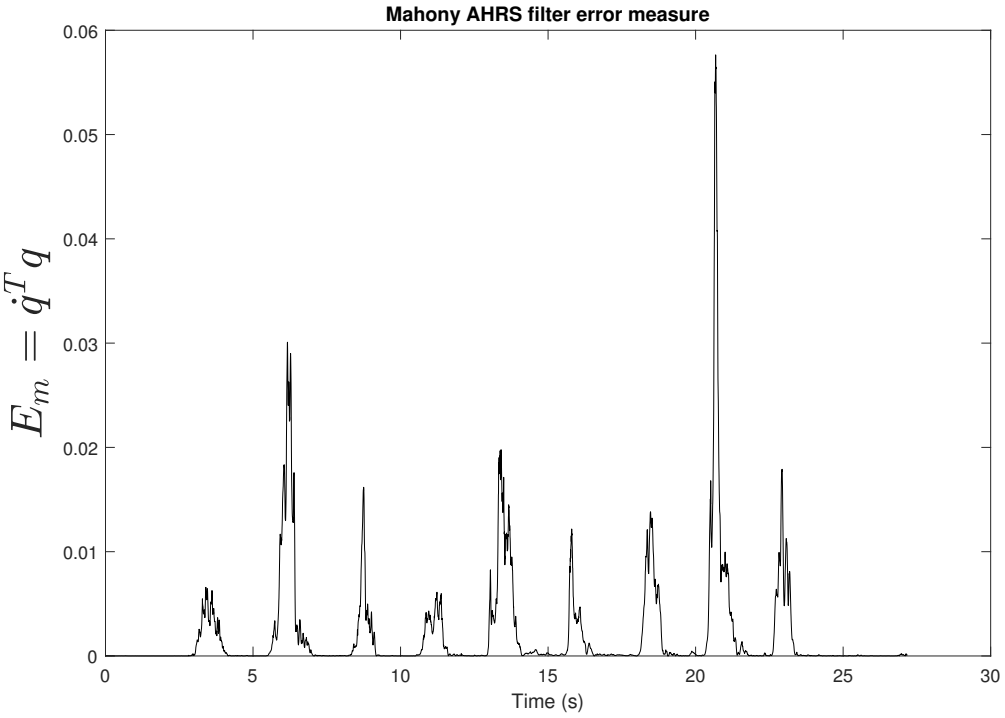


Figure 4.3: Error analysis for the Mahony algorithm

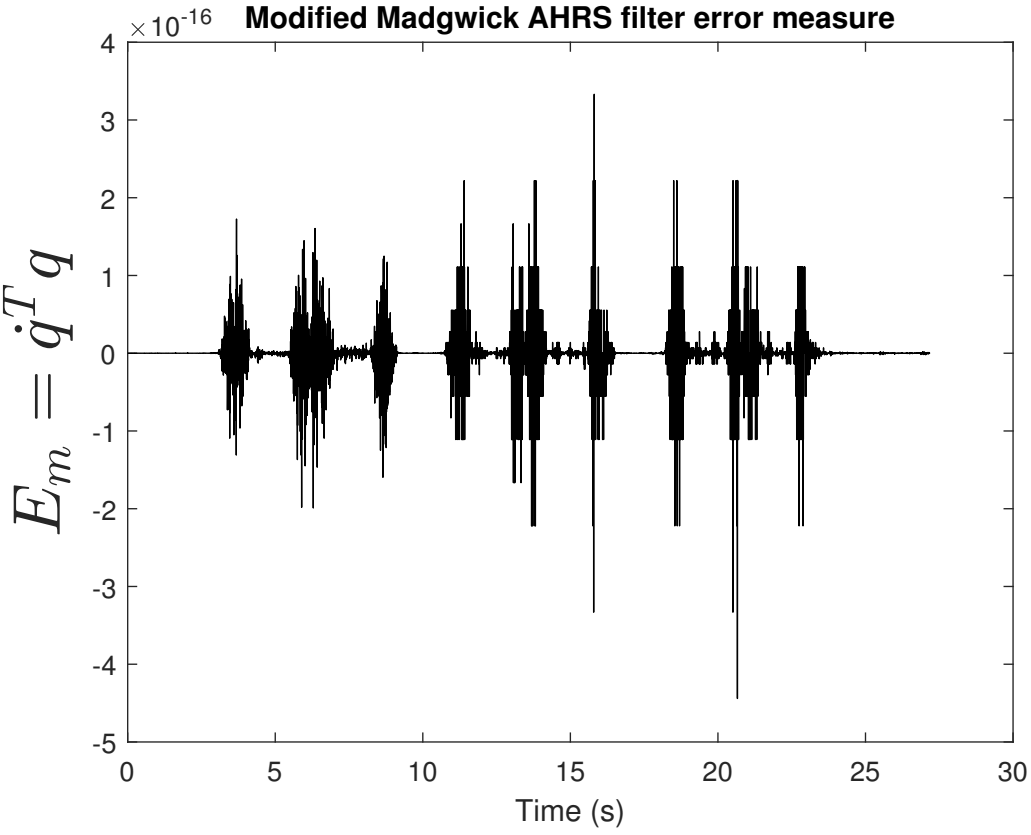


Figure 4.4: Error analysis for the proposed algorithm (4 *refinement steps*)

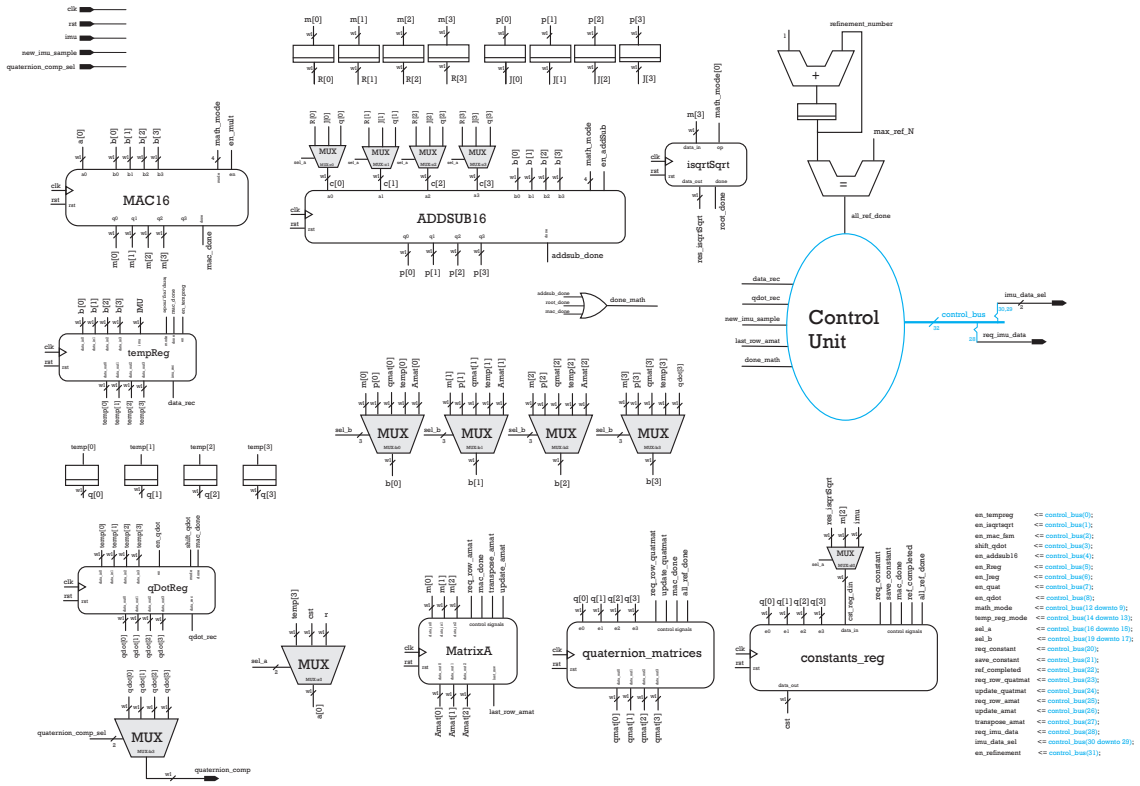
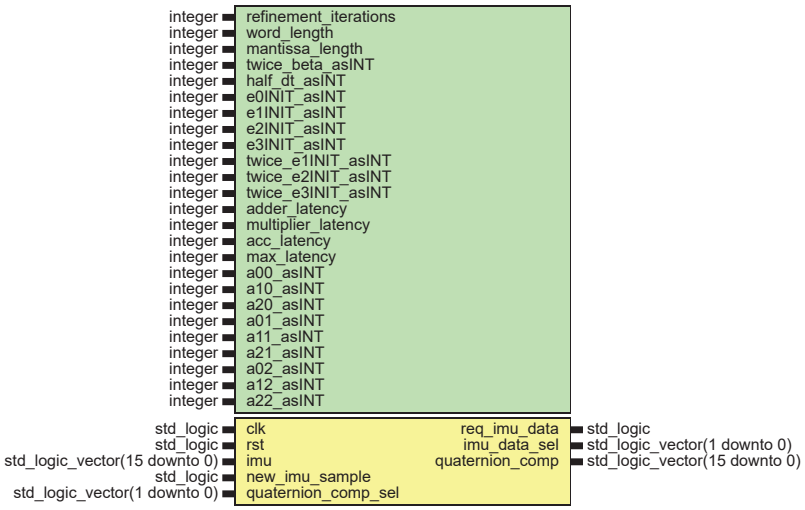


Figure 4.5: Sensor Fusion FPGA architecture



Generics

Generic name	Type	Value	Description
refinement_iterations	integer	3	refinement iterations to execute
word_length	integer	16	word length
mantissa_length	integer	10	mantisa n. of bits
twice_beta_asINT	integer	16#3800#	half beta (see Madgwick algorithm)
half_dt_asINT	integer	16#1419#	half sampling time
e0INIT_asINT	integer	16#3C00#	initial quaternion component e0
e1INIT_asINT	integer	16#0000#	initial quaternion component e1
e2INIT_asINT	integer	16#0000#	initial quaternion component e2
e3INIT_asINT	integer	16#0000#	initial quaternion component e3
twice_e1INIT_asINT	integer	16#0000#	2e1
twice_e2INIT_asINT	integer	16#0000#	2e2
twice_e3INIT_asINT	integer	16#0000#	2e3
adder_latency	integer	4	floating point adder latency
multiplier_latency	integer	5	mac16 floating point multiplier latency
acc_latency	integer	4	mac16 accumulator latency
max_latency	integer	9	mac16 max latency
a00_asINT	integer	16#3c00#	rst value of A(0,0) = 1
a10_asINT	integer	16#0#	rst value of A(0,1) = 0
a20_asINT	integer	16#0#	rst value of A(0,2) = 0
a01_asINT	integer	16#0#	rst value of A(1,0) = 0
a11_asINT	integer	16#3c00#	rst value of A(1,1) = 1
a21_asINT	integer	16#0#	rst value of A(1,2) = 0
a02_asINT	integer	16#0#	rst value of A(2,0) = 0
a12_asINT	integer	16#0#	rst value of A(2,1) = 0
a22_asINT	integer	16#3c00#	rst value of A(2,2) = 1

Ports

Port name	Direction	Type	Description
clk	in	std_logic	clock signal
rst	in	std_logic	active low reset
imu	in	std_logic_vector(15 downto 0)	IMU data
new_imu_sample	in	std_logic	high when new imu data are available
quaternion_comp_sel	in	std_logic_vector(1 downto 0)	select quaternion component to output
req_imu_data	out	std_logic	asserted high to request imu data
imu_data_sel	out	std_logic_vector(1 downto 0)	imu data to import
quaternion_comp	out	std_logic_vector(15 downto 0)	outoput quaternion component

Figure 4.6: Pinout of the Sensor Fusion block

4.2.1 Synthesis results

The architecture depicted in Figure 4.5 has been synthesized successfully on a `Cyclone V 5CEFA5F23C7 FPGA`. The resource occupation and the maximum frequency are reported in Tables 4.2 and 4.3, respectively.

Table 4.2: Resource Utilization

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

Table 4.3: 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

4.3 FPGA Implementation of the proposed approximation for ISQRT and SQRT

In this section, an `FPGA` implementation of the computation flow shown in Figure 3.2 is presented. The distinctive features of our design with respect to existing ones are:

- Both inverse square root and square root operations are carried out with the same architecture. In fact, most of the architectures computes or the square root of the inverse square root. Only Hasnat *et al.* [42] use the same hardware, but the square root is simply $\sqrt{x} = \frac{1}{\sqrt{x}} \cdot x$
- The latency of the algorithm does not rely upon the input precision. The computation method, can be promptly extended to the double precision case.
- Only six clock cycles are necessary.
- No iteration or seed tables are required.
- Only a single multiplier is required.

The `Twente ISQRT & SQRT` architecture (see Figure 4.19) has two sub-components:

- `polyroot`: this component computes the output mantissa;
- `exponent`: this component computes the output exponent (see Figure 4.20).

The pinout of the proposed architecture is depicted in Figure 4.11:

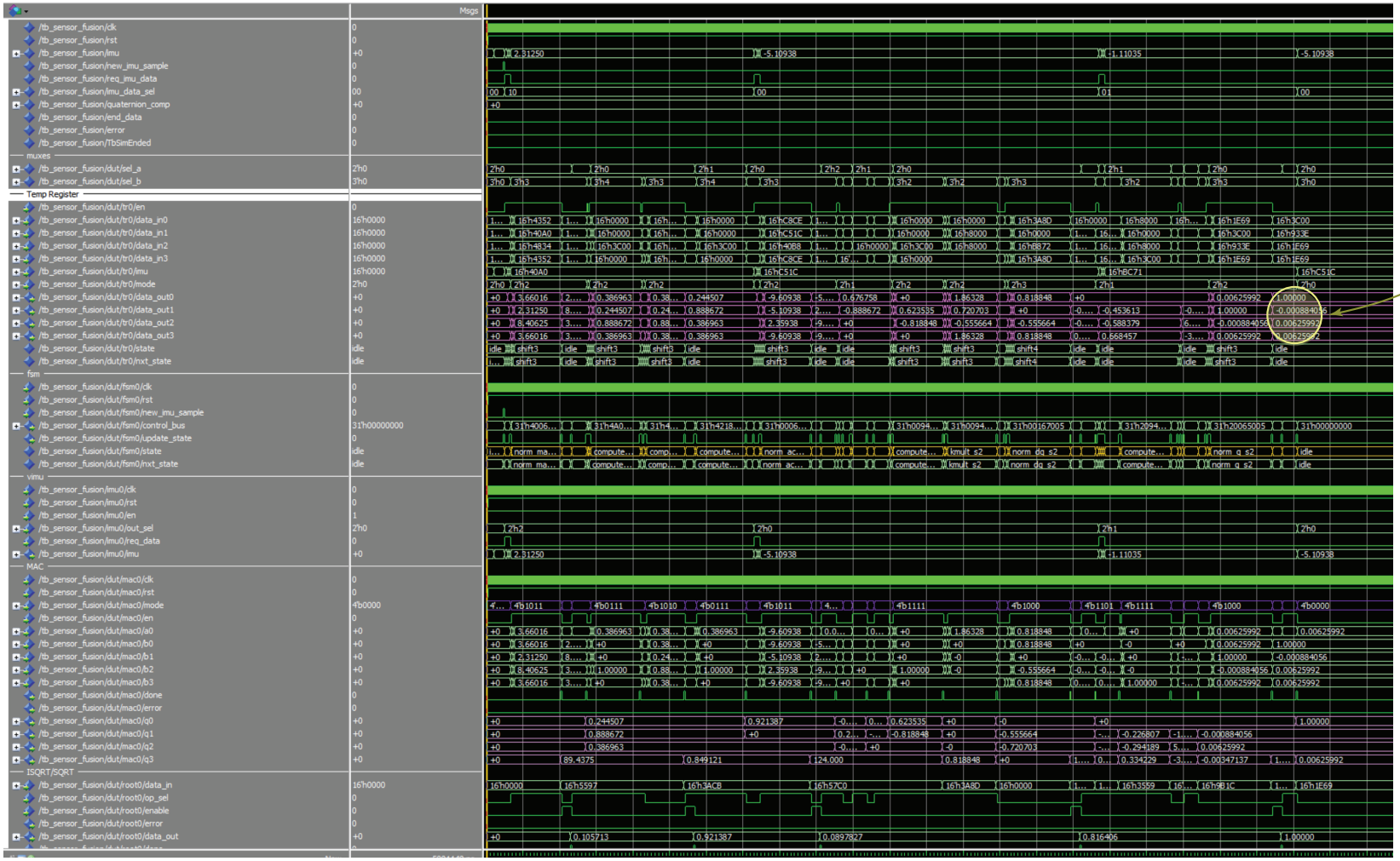


Figure 4.7: Sensor fusion simulation results

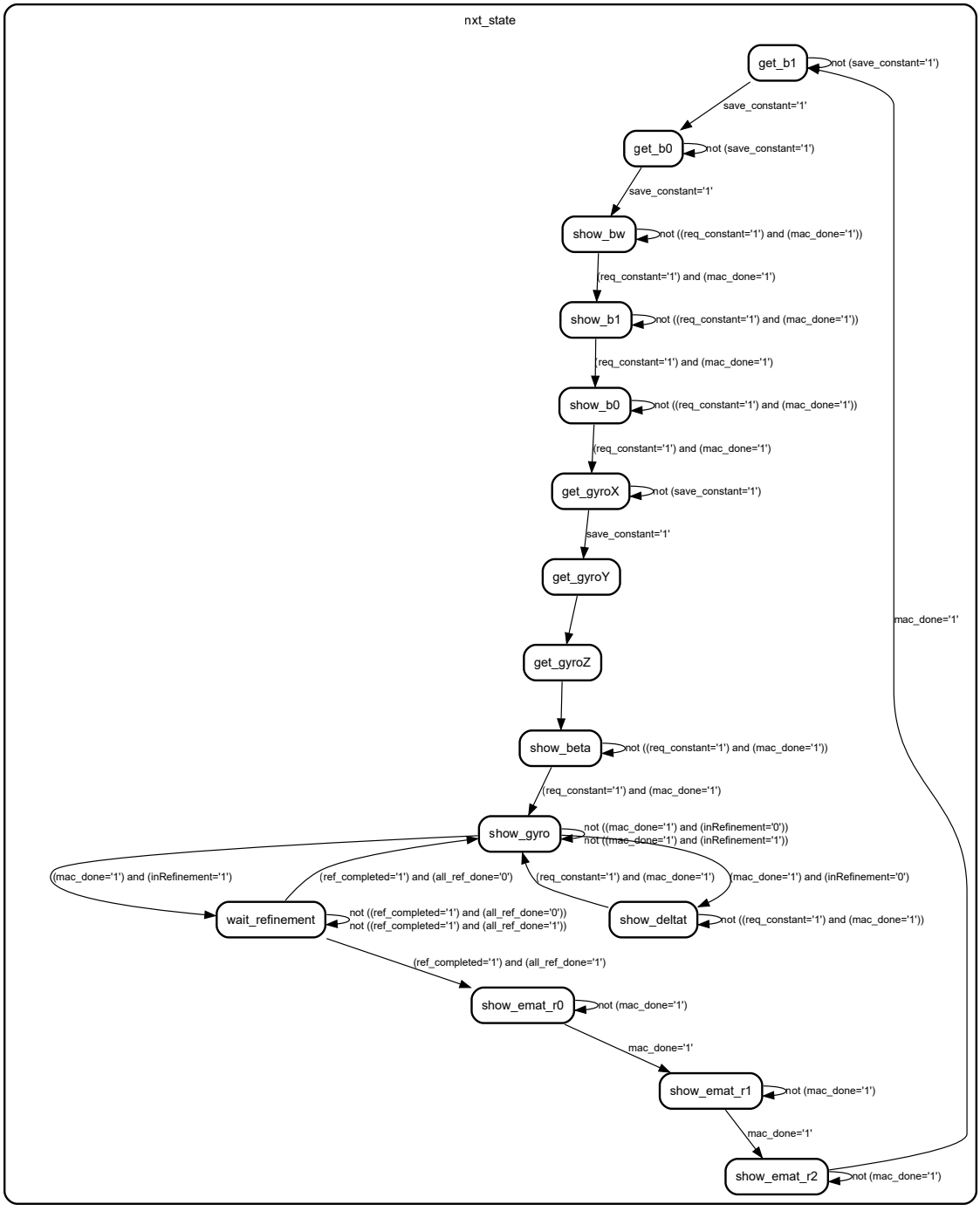


Figure 4.8: Finite state Machine of the `constants_reg` component

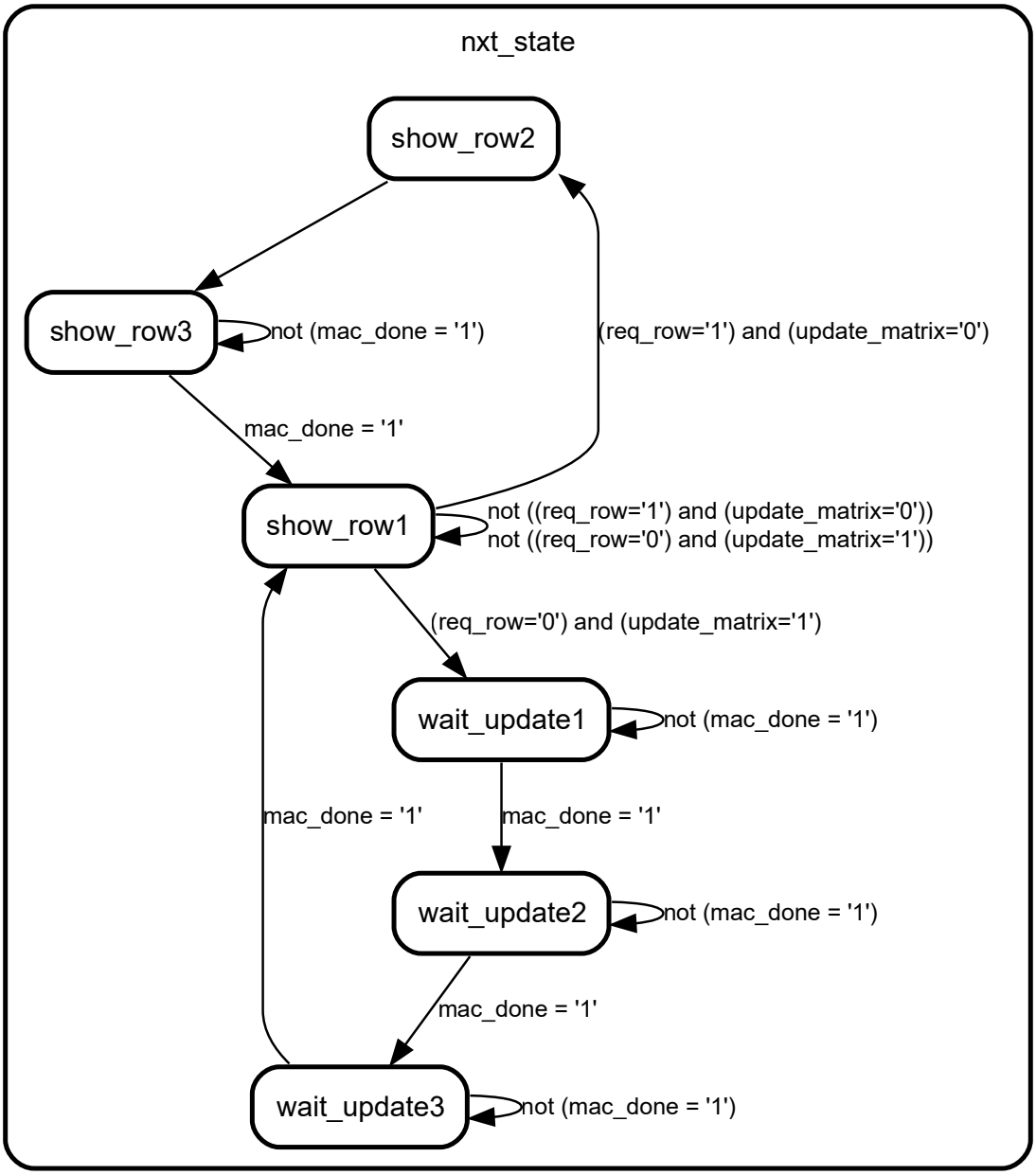


Figure 4.9: Finite state Machine of the `MatrixA` component

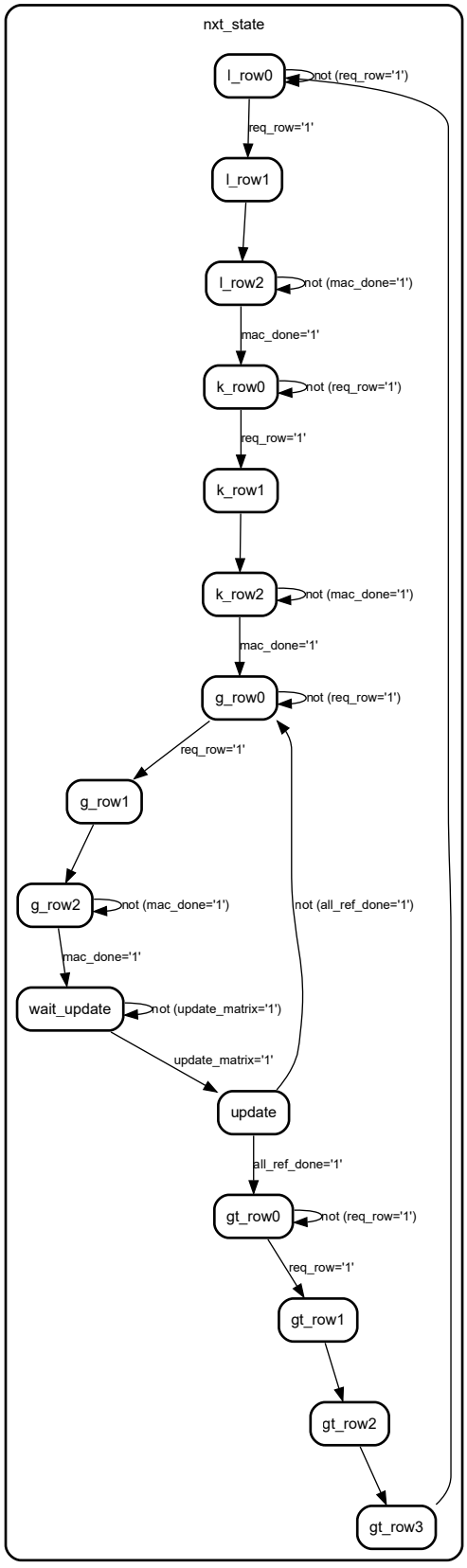
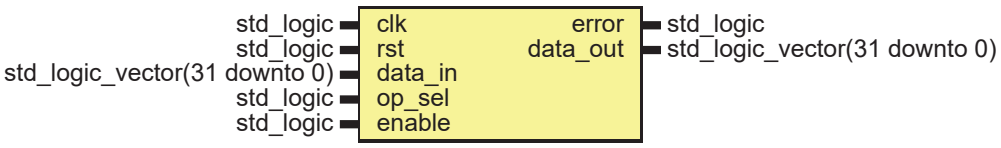


Figure 4.10: Finite state Machine of the `quaternion_matrices` component



Port name	Direction	Type	Description
clk	in	std_logic	input clock
rst	in	std_logic	active low reset signal
data_in	in	std_logic_vector(31 downto 0)	input data
op_sel	in	std_logic	select operation: 0: isqrt, 1: sqrt
enable	in	std_logic	enable signal
error	out	std_logic	error signal
data_out	out	std_logic_vector(31 downto 0)	output data

Figure 4.11: Pinout of the proposed Twente ISQRT & SQRT architecture

4.3.1 Exponent Computation

In IEEE-754 32 bit, 8 bits are dedicated to the exponent and it is expressed in excess 127. Let e and m be the input biased (excess 127) exponent and input mantissa, respectively. The following cases can be distinguished:

- **Inverse Square Root**
  Since it is
  $$\frac{1}{\sqrt{m}} < 1 \text{ for } m \in (1\ ,\ 2)$$
  the output mantissa, multiplied by two, will be in the range $[1\ ,\ 2)$.

  Finally, to account for the multiplication by two of the output mantissa, **the output exponent is reduced by a unit**.

  - **isqrt even unbiased exponent**
    $$e_{out} = \frac{-(e-127)}{2} - 1 + 127 = \frac{379-e}{2} \tag{4.1}$$
  - **isqrt odd unbiased exponent**
    In case of odd unbiased exponent, the exponent adopted is the nearest even number $e-1-127$. Thus, the output biased exponent is computed as follows:
    $$e_{out} = \frac{-(e-1-127)}{2} - 1 + 127 = \frac{380-e}{2} \tag{4.2}$$
    Finally, to account for this exponent modification, the output mantissa is multiplied by $\frac{1}{\sqrt{2}}$.

- **isqrt even unbiased exponent, unity mantissa**
  When the input mantissa $m = 1$, then the polynomial approximation is not used. For these special cases, the mantissa normalization is not required. Thus, the output biased exponent is computed as follows:
$$e_{out} = \frac{-(e-127)}{2} - 1 + 127 = \frac{381-e}{2} \tag{4.3}$$

- **Square Root**

  - **sqrt even unbiased exponent**
$$e_{out} = \frac{e-127}{2} + 127 = \frac{e+127}{2} \tag{4.4}$$

  - **sqrt odd unbiased exponent**
    In case of odd unbiased exponent, the exponent adopted is the nearest even number $e-1-127$. Thus, the output biased exponent is computed as follows:
$$e_{out} = \frac{(e-1)-127}{2} + 127 = \frac{e+126}{2} \tag{4.5}$$
    Finally, to account for this exponent modification, the output mantissa is multiplied by $\sqrt{2}$.

4.3.2 Mantissa Computation

Let $x \neq 1$ be the input mantissa. The output mantissa is computed evaluating a second degree polynomial $P_2(x)$ ((3.71) both for the square root and (3.54) for the inverse square root operations).

The following four cases can be distinguished:

- **isqrt even unbiased exponent**
$$P_2(x) = \left(p_{20}x + p_{10}\right)x + p_{00} \tag{4.6}$$

- **isqrt odd unbiased exponent**
$$P_2(x) = \left(p_{20}x + p_{10}\right)\frac{x}{\sqrt{2}} + p_{01} \tag{4.7}$$
  where $p_{01} = \dfrac{p_{00}}{\sqrt{2}}$

- **sqrt even unbiased exponent**
$$P_2(x) = \left(p_{22}x + p_{12}\right)x + p_{02} \tag{4.8}$$

- **sqrt odd unbiased exponent**
$$P_2(x) = 2\left[\left(p_{22}x + p_{12}\right)\frac{x}{\sqrt{2}} + p_{03}\right] \tag{4.9}$$
  where $p_{03} = \dfrac{p_{02}}{\sqrt{2}}$

Two special cases can be distinguished when computing the output mantissa:

1. The input mantissa is $m_{in} = 1$. In this case the polynomial approximation does not apply and the output mantissa is computed as follows:

$$m_{out} = \begin{cases} 1 & \text{if even unbiased exponent} \\ \sqrt{2} & \text{if odd unbiased exponent} \end{cases} \tag{4.10}$$

2. The input data is 0 (`0x00000000` or `0x80000000`). In this case the polynomial approximation does not apply and the output is zero if the selected operation is the square root. For the inverse square root case an error is returned.

   The coefficients of the approximating polynomials are stored into a `ROM` of 64 words and a word length of 20 bits. These are signed with 18 fractional bits. The following hardware resources are foreseen for the polynomial evaluation:

   - 1× Adder
   - 1× Multiplier
   - 1× ROM to hold the polynomials coefficients
   - Registers to hold $\frac{1}{\sqrt{2}}$ constant, multiplication/addition results and the result mantissa.

   The scheduling of operations is summarized in Table 4.4, whereas the architecture is depicted in Figure 4.12. The ASM chart of the control unit is depicted in Figure 4.13.

Table 4.4: Scheduling of the polynomial mantissa evaluation

	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		$a \leftarrow m + p_1$		$r \leftarrow m + p_0$

4.3.3 Error Generation

An error occurs when:

- the input data is negative
- the input data is zero and the selected operation is the `ISQRT`

An error does not occur when the selected operation is `sqrt` and the input is a negative zero.[1].

4.3.4 System Testing

The main phases of the system testing are shown in Figure 4.14. Python functions generate the input in a given range and compute the output result error. As an example, Figure 4.18, depict the relative error in the range (0 , 1e6]. A similar testing strategy was adopted to verify the `polyroot` component. The error plots of such simulation are depicted in Figures 4.15, 4.16 and 4.17. From these plots it can be deduced that

[1] The `IEEE 754` notation, allows two representation for 0: +0 (`0x00000000` positive zero) and -0 (`0x80000000` negative zero)

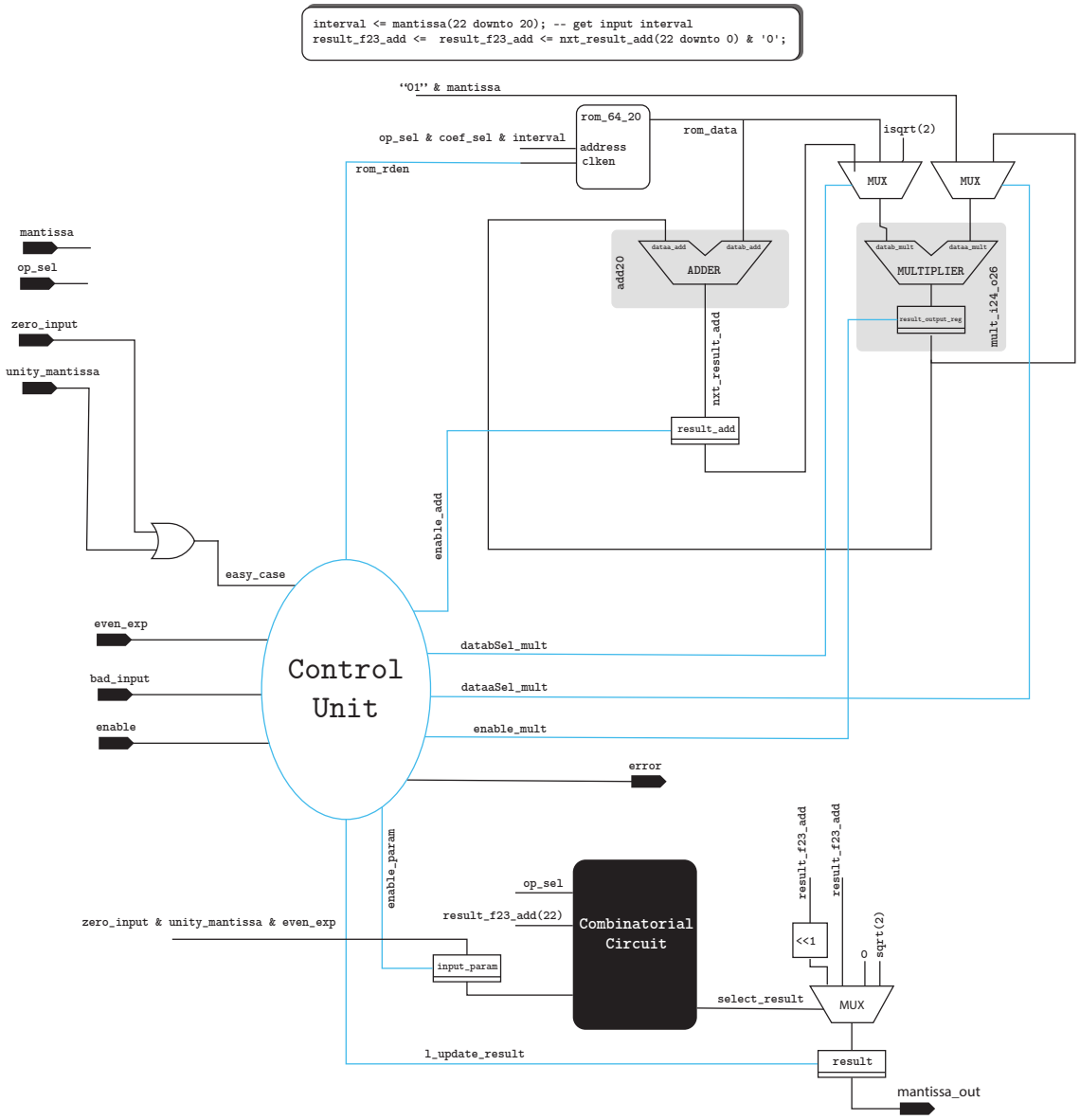


Figure 4.12: Proposed simplified schematic of the mantissa computation block (`polyroot` architecture). For clarity, some details have been omitted.

- The worst-case relative error for the inverse square root case is `1.9706509339401565e-05`
- The worst-case relative error for the square root case is `8.111330020459344e-06`

In both cases the relative error is less than the $\frac{1}{2}$ ulp of the half precision format.

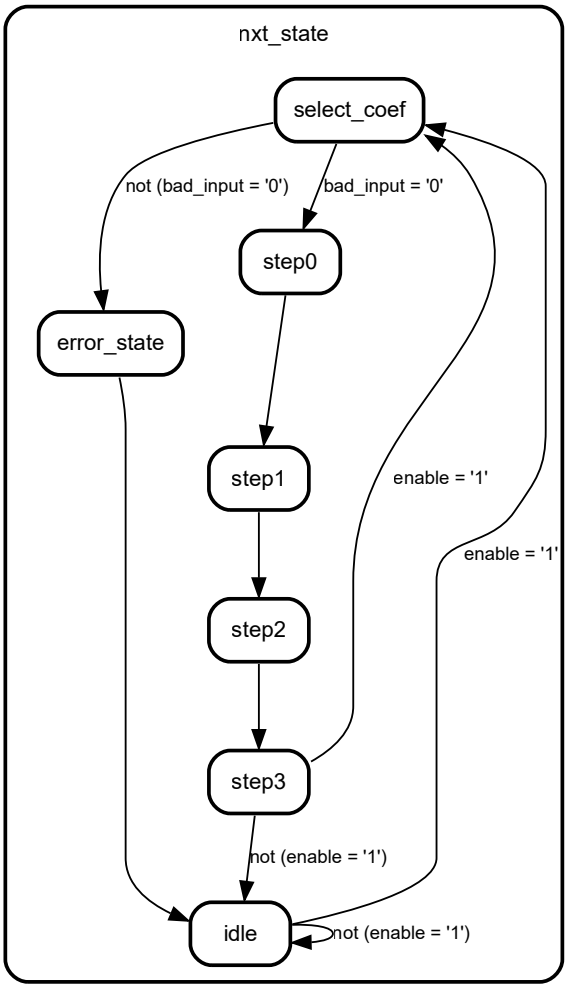


Figure 4.13: ASM chart of the proposed control scheme

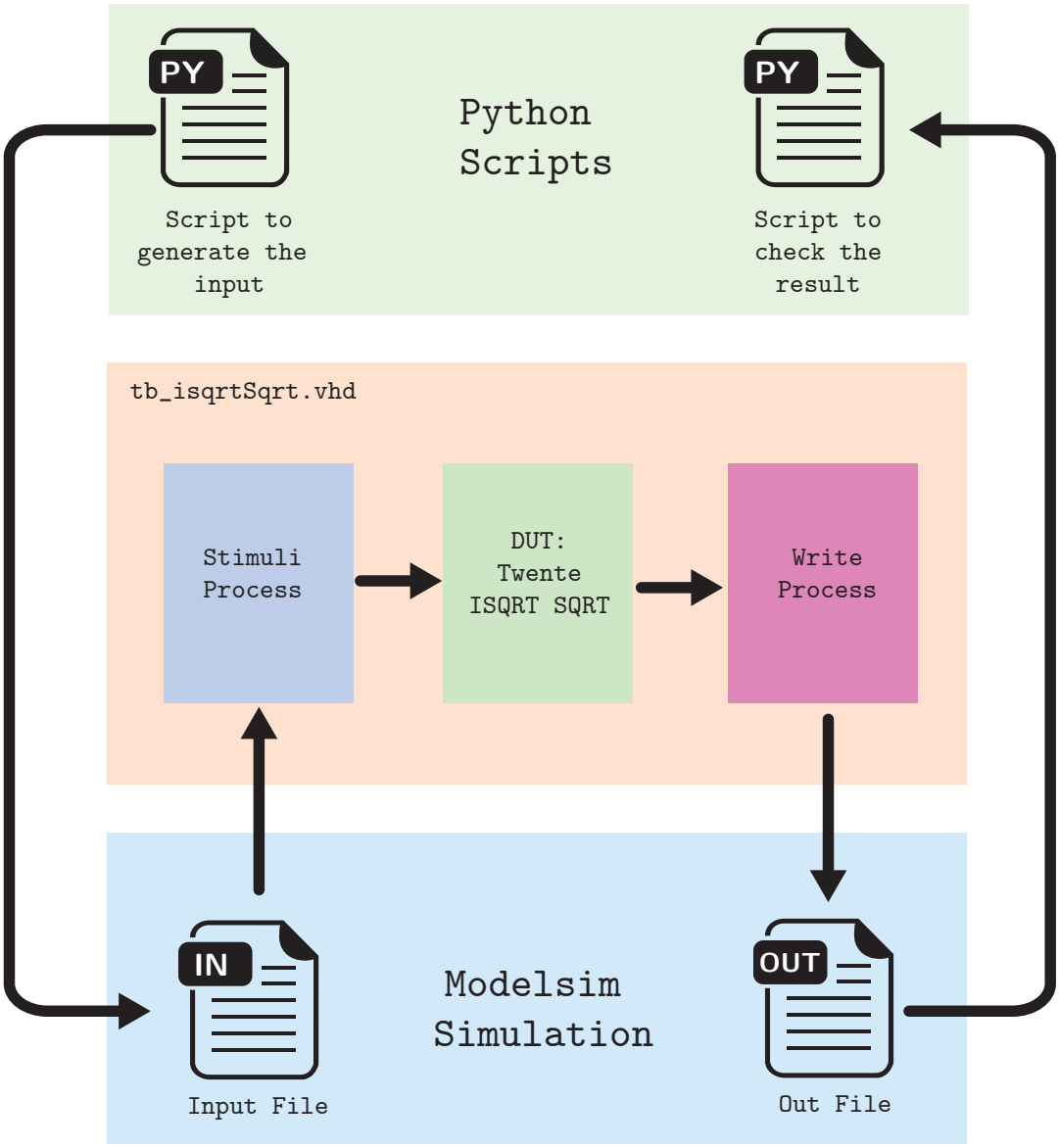


Figure 4.14: Testing Phases of the `Twente ISQRT SQRT`

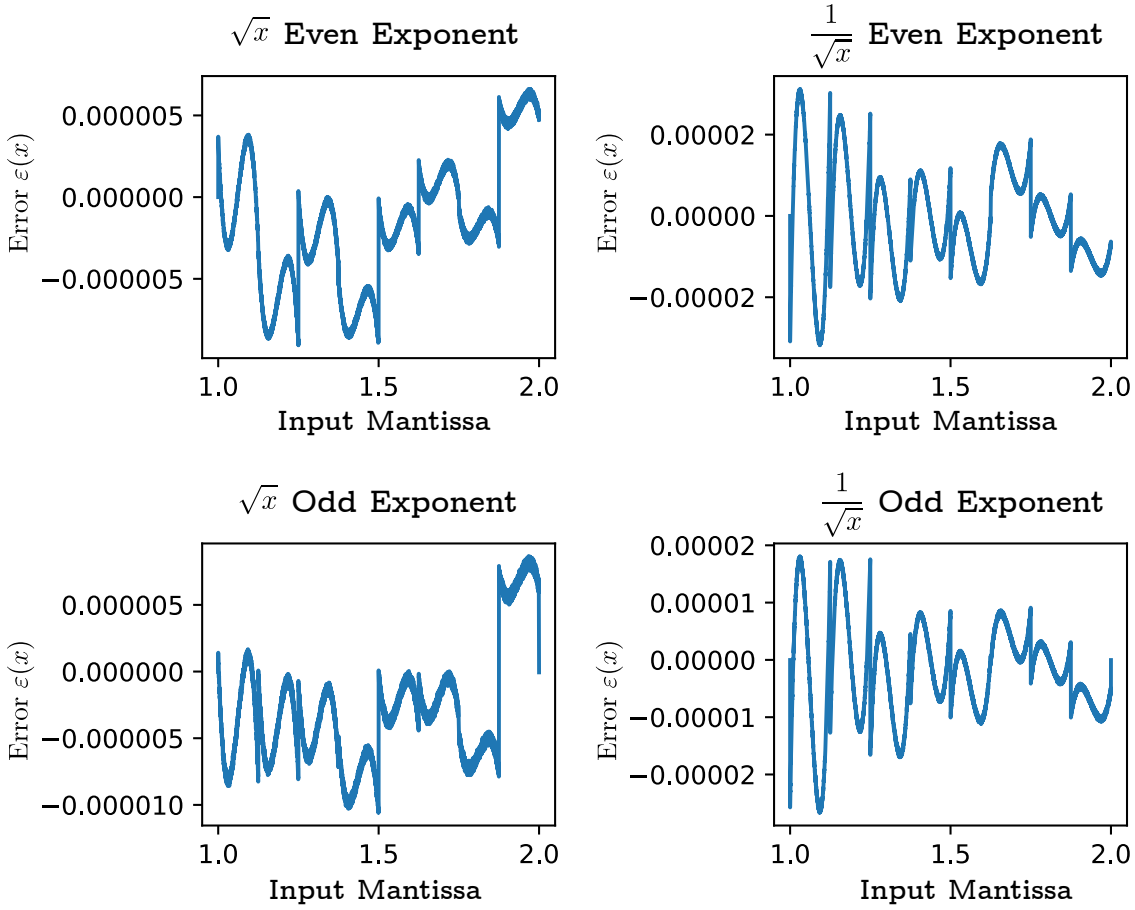


Figure 4.15: Result of mantissa error analysis

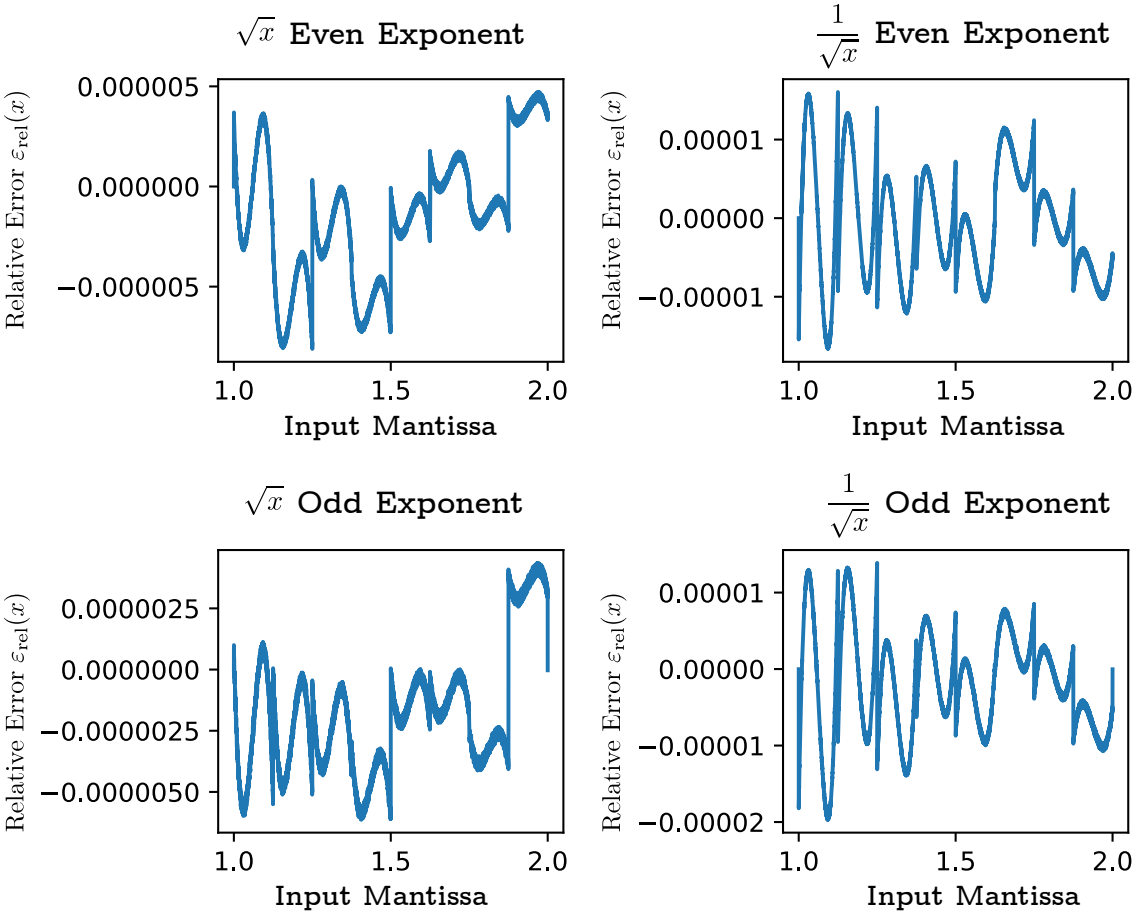


Figure 4.16: Result of mantissa error analysis (relative error).

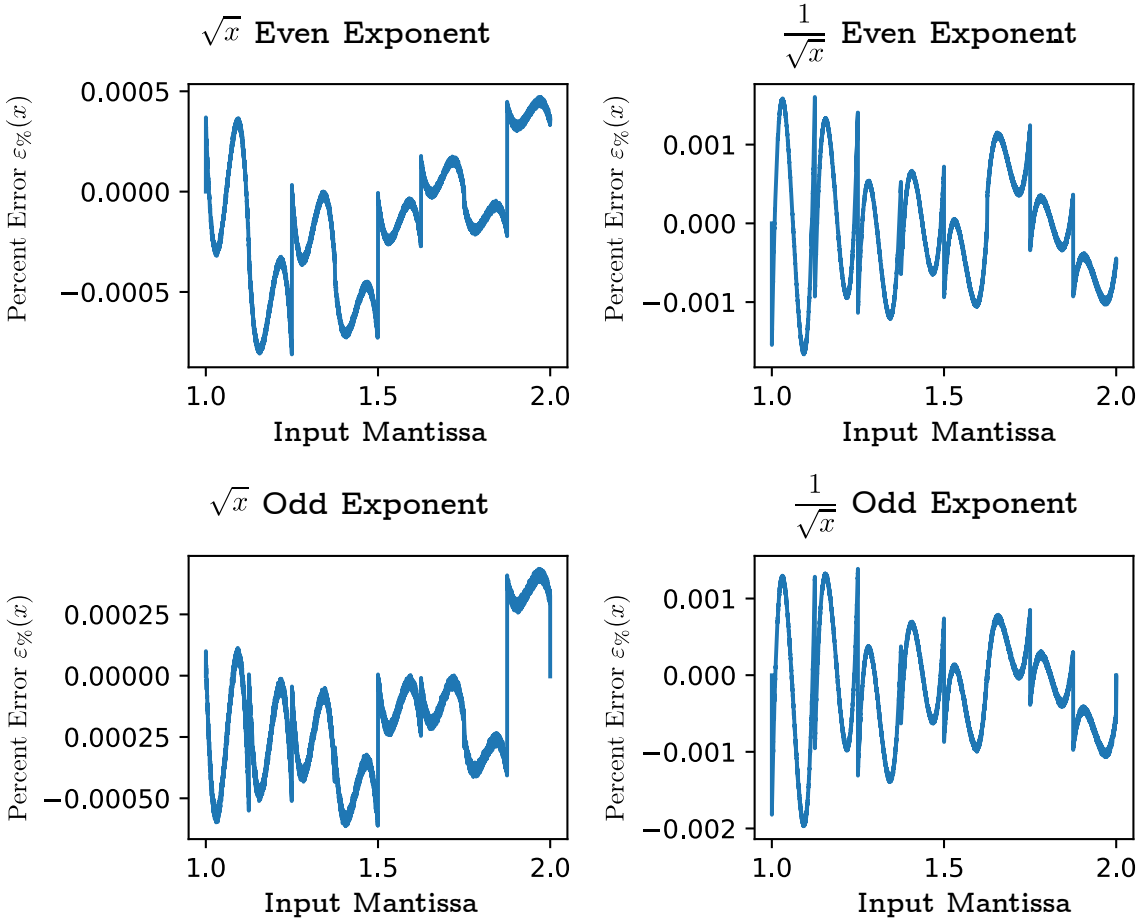


Figure 4.17: Result of mantissa error analysis (percent error).

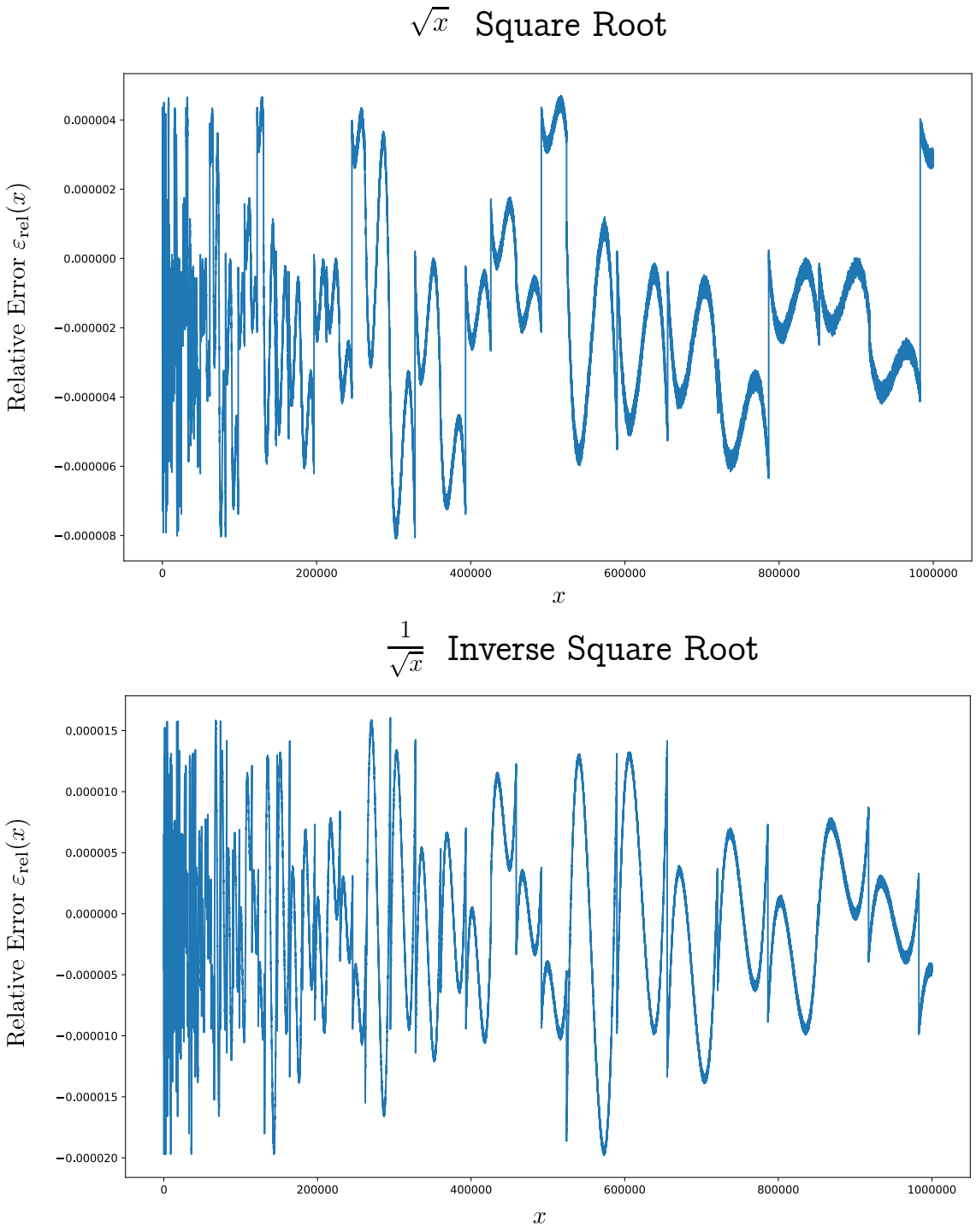


Figure 4.18: System error analysis for the square root and inverse square root cases

4.3.5 Synthesis Result

The proposed architecture has been synthesized successfully on a `Cyclone V 5CEFA2F23I7 FPGA`. The resources occupation are reported and compared with the `ALTFP_INV_SQRT` IP block in Table 4.5.

Table 4.5: Resource comparison with the `ALTFP_INV_SQRT` IP

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

Table 4.6: 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

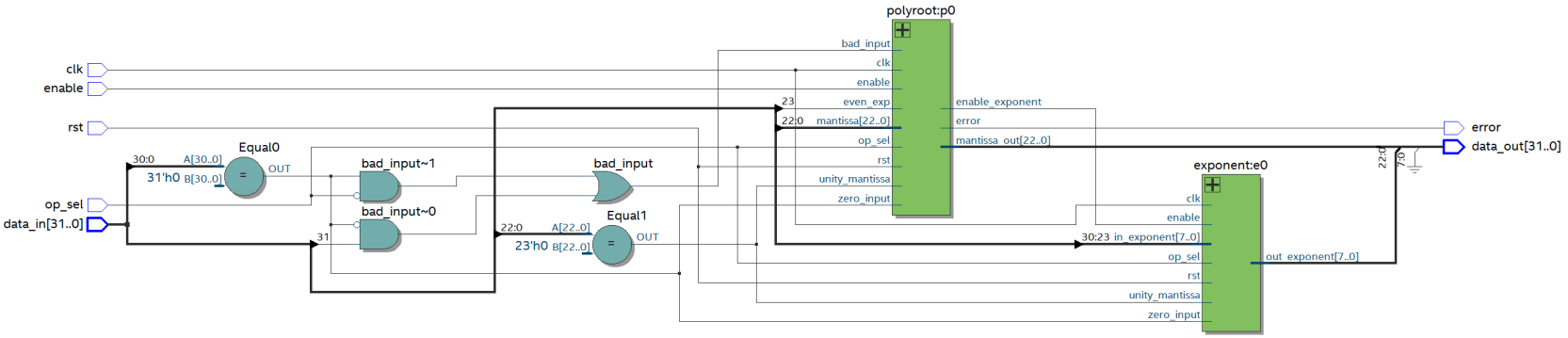


Figure 4.19: System Architecture

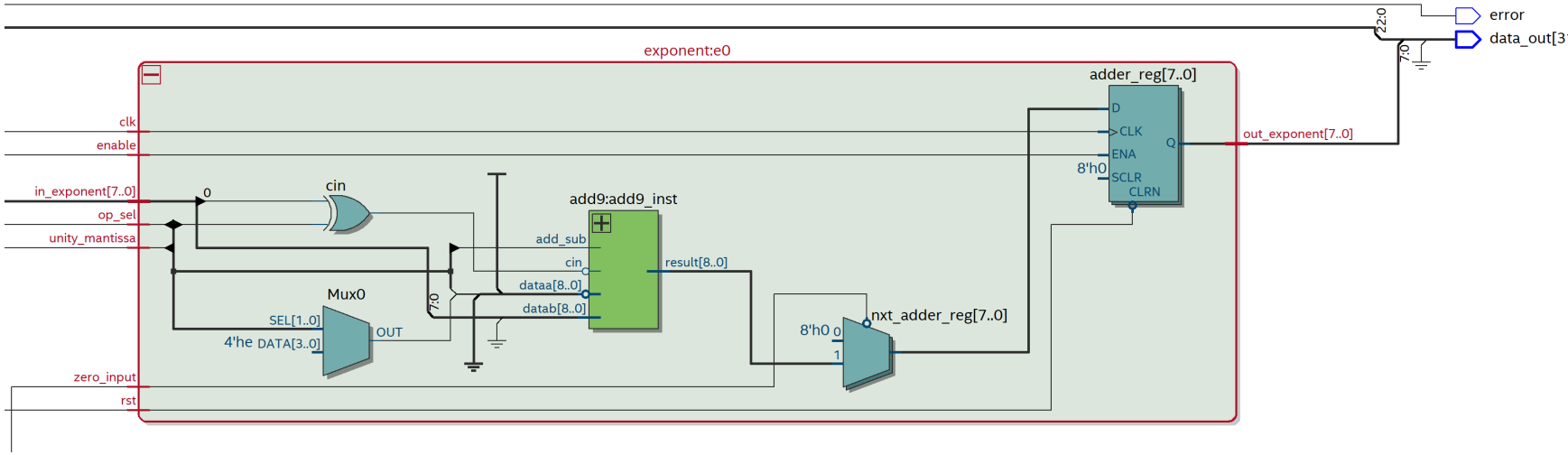


Figure 4.20: Exponent Architecture

Chapter 5

Applications

Espice, adspice, prospice.
Look behind, look here, look ahead.

Latin motto.

The data fusion algorithms are essential tools for the development of Internet of Things multi-sensor based intelligent devices. The data synergically collected from local multiple sensors need to be processed and synthesized to provide low redundancy and reliable information in support of tasks such as:

- decision-making processes;
- operation monitoring and automation;
- test and verify compliance requirements;
- visualize usage patterns.

In this chapter is presented an application of the sensor fusion algorithm previously discussed. In particular, the application herein considered refers to the development of a novel `IMU` based onboard device for bikers vibration comfort analysis and road condition assessment.

The raw accelerometer data provided by the `IMU` sensor contain the gravity component that needs to be removed before further processing.

This can be achieved by means of the following operations, also depicted in Figure 5.1:

- Define sensor orientation transform matrix ${}^{w}_{s}\mathbf{A}$ from the local `IMU` Cartesian reference system to the world Cartesian system.
- Transform the acceleration Cartesian components provided by the accelerometer sensor from the local `IMU` reference into the world reference

$$ {}^{w}\mathbf{a} = {}^{w}_{s}\mathbf{A}\,{}^{s}\mathbf{a} \tag{5.1} $$

- Redefine acceleration in world Cartesian reference by subtracting the gravity vector

$$ {}^{w}\mathbf{a} \rightarrow {}^{w}\mathbf{a} - \mathbf{g} \tag{5.2} $$

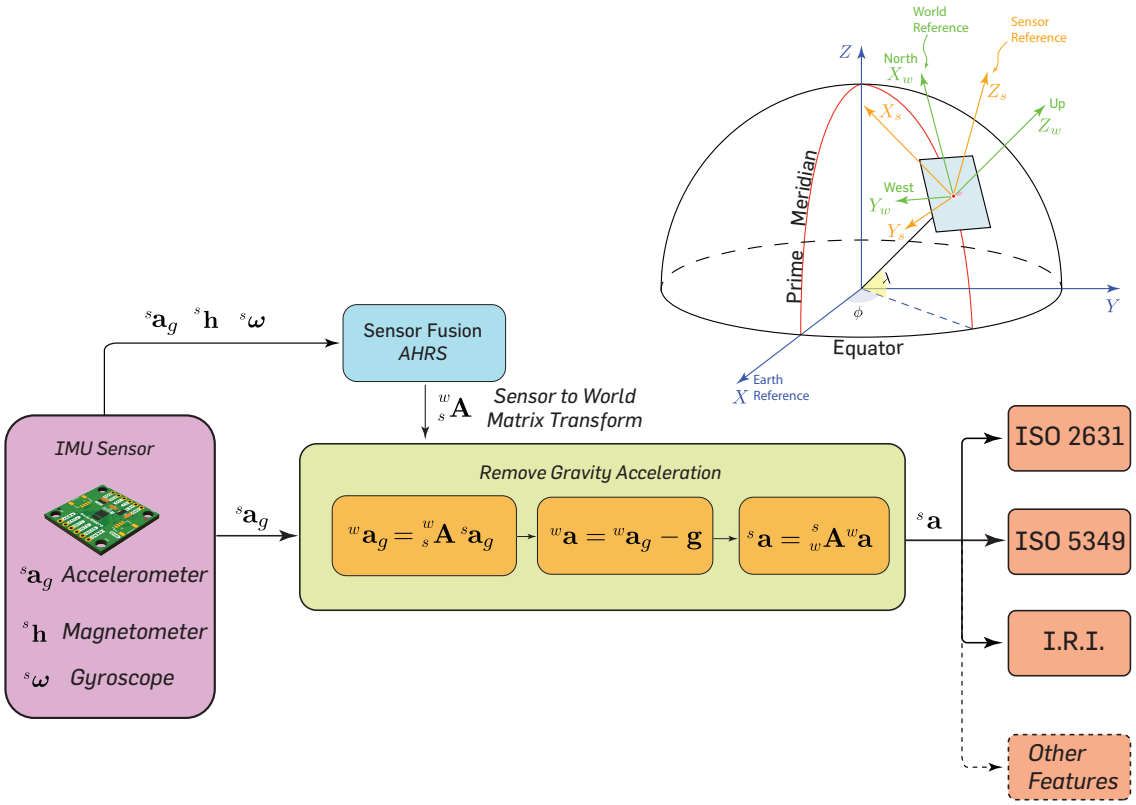


Figure 5.1: Gravity removal from accelerometer data for their processing according to international ISO norms

- Transform the acceleration components back to the local IMU Cartesian reference

$$ {}^s\mathbf{a} = {}^w_s\mathbf{A}^\intercal\, {}^w\mathbf{a} \tag{5.3}$$

In these components the contribution of acceleration gravity is now purged.

5.1 The Smart Bike Concept

The low carbon emission, imposed by legislation in many countries, requires a sustainable mobility based on a new class of energy efficient, sensorized and connected class of vehicles. In the automotive industry, the availability of sensors paved the road toward new applications, such as assisted driving, anti collision systems and big data collection.
The use of bikes, as a city transport vehicles, is nowadays well acknowledged and widespread among population. The modern trend of vehicle digitalization can be already observed in the new models available on the market, and the bikes are no exception. In these early phases, the digitalization of bikes is mainly

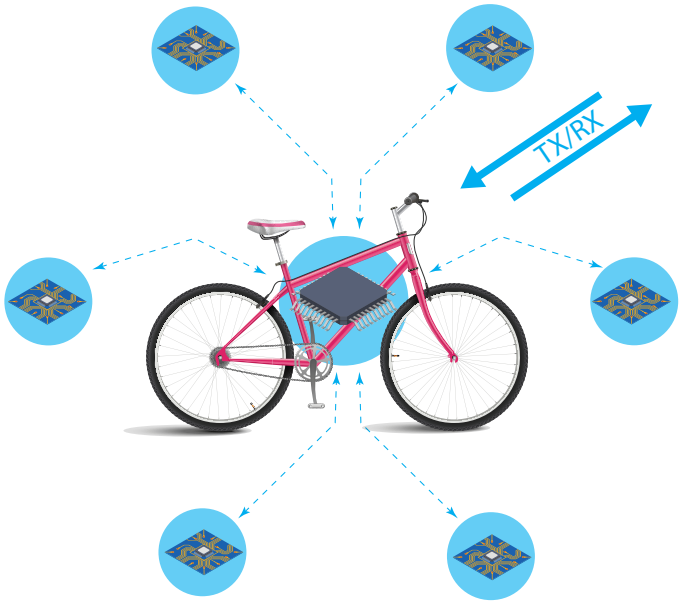


Figure 5.2: The Smart Bike Concept

limited to the electric motor power control. However, replicating the automotive trend, sensorization of bikes offers new possibilities for many purposes.
To make a clear distinction between a *traditional* and this new generation of bikes, the term *Smart Bike* is often reported in advertisement. However, a clear definition of such a term could not be found. In the following, a tentative definition is proposed:

> 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**.

A pictorial conceptual view of a *Smart Bike* is shown in Figure 5.2, where a large variety of sensors/devices, can be imagined:

- accelerometers & gyroscopes;
- compass;
- dash-cam;
- ultrasonic distance meter;
- GPS;
- communication devices;

- air quality sensors;
- temperature and humidity sensors;
- speedometer;
- battery and electric motor power control;
- lights control;
- automatic emergency assistance request;
- ...

5.2 V-AUDIT: a vibration monitoring device

V-AUDIT stands for **V**ibration **Audit**[1] and is herein proposed as *a low cost bike accessory* composed of:

- a unit to collect and *onboard* process of different **I**nertial **M**easurement **U**nit (`IMU`) sensors data;
- different `IMU`s positioned at bike-human body interfaces (seat, handlebar).

Accelerometer data contain a lot of information. However, this needs to be extracted by processing data according to appropriate algorithms and models. A *distinctive* feature of V-AUDIT is the elaboration of accelerometer data according to `ISO`. This choice is fundamental because allows:

- to make use of **safety thresholds** established and widely recognized by the Scientific Community and in Legislation;
- consistency and comparability of processed data for different bike models and riders anthropometry.

5.2.1 Functional Features of V-AUDIT

The device is designed with a modular architecture in mind. In particular, the proposed functional features, are:

1. **Human Vibration Assessment**: (Whole-Body `ISO 2631` [18] + Hand Arm `ISO 5349` [25])
2. **Road Condition Assessment** (`IRI` **I**nternational **R**oughness **I**ndex) [17]
3. **Automatic Accident Detection** The system should be able to detect an accident event and trigger an Automatic Communication Unit and warning signal (e.g.: sound & light).

This last feature is not herein explored.

[1] The term has been inspired by the Latin word *audire* which means "to hear", "pay attention to".

Hand-Arm Vibration
The mechanical vibration that, when transmitted to the human hand-arm system, entails risks to the health and safety of workers, in particular vascular, bone or joint, neurological or muscular disorders [60].

Whole-Body Vibration
The mechanical vibration that, when transmitted to the whole body, entails risks to the health and safety of workers, in particular lower-back morbidity and trauma of the spine [60].

In the following, we will document some examples where the previous functionalities provide **solutions** to the current society needs and mobility related problems.

5.3 The Social Usefulness & Impact of V-AUDIT

5.3.1 The Rider Point of View

Nowadays, bike delivery is considered a cost effective solution for reducing both carbon emissions and delivery times. The bike riders (see Figure 5.3), form a new category of workers.
They make a prolonged and intensive use of bikes traveling all types of city roads. Not all roads, such as paved ones, have been designed and maintained for optimal bike use. Therefore, they are exposed

Figure 5.3: A rider delivery worker

to whole body and hand-arm vibrations, with a significant risk of low back pain and spinal injury or

degeneration [50]. The back load unbalance worsen the problem.
All the legislations of the European Union countries embody the **EU Directive 2002/44/ES** [60] limiting the amount of daily **W**hole **B**ody **V**ibration workers exposure. At the time being, on the bike accessory market, there is not any device that allows the rider to monitor if its daily exposure to vibration overcome the law limits. Because of its importance, the human vibration monitoring is an intensive and actively pursued research area, as witnessed by thousands of contributions published on scientific journals. Notwithstanding this,

> ... *Further research into the effect of cycling conditions and equipment on WBV would be valuable to both the research and cycling communities*[50].

Moreover, Prof. Chris Oliver (Royal College Surgeons of Edinburgh), in a recent interview at The Herald [48], called for:

> *A HIGH-TECH bicycle which measures cyclists' exposure to potentially harmful vibrations from uneven road surfaces could be used to check the safety of cycle lanes and other routes.*

Figure 5.4: Cyclist after a fall

Regarding accidents, a cyclist fall causes a change in the pattern of accelerometer data. When the **fall** happens at night, or in a remote area, the life of the cyclist could be at **risk**. In fact, the cyclist could be passed out or not able to call for help. V-AUDIT detects the change of patterns and triggers **sound and light signals**. **Emergency messages** with geo-localization data are sent.

5.3.2 The Employer Point of View

The EU Directive [60] on workers vibration monitor requires the employer to record:

- the level, type and duration of exposure, including any exposure to intermittent vibration or repeated shocks;
- the exposure limit values and the exposure action values laid down in Article 3 of Directive 2002/44/EC [60];
- any effects concerning the health and safety of workers at particularly sensitive risk;

- any indirect effects on worker safety resulting from interactions between mechanical vibration and the workplace or other work equipment;
- the extension of exposure to whole-body vibration beyond normal working hours under the employer's responsibility.

Table 5.1: Vibration limits established by EU Directive 2002/44/EC [60]

	Hand-Arm, RMS	Whole-Body, RMS	Whole-Body, VDV
Exposure action value	2.5 m/s^2	0.5 m/s^2	9.1 m/s^{175}
Exposure limit	5 m/s^2	1.15 m/s^2	21 $m/s^{1,75}$

Furthermore, the law imposes:

> *... The risk assessment shall be recorded on a suitable medium, according to national law and practice; it may include a justification by the employer that the nature and extent of the risks related to mechanical vibration make a further detailed risk assessment unnecessary. The risk assessment shall be kept up-to-date on a regular basis, particularly if there have been significant changes which could render it out-of-date, or when the results of health surveillance show it to be necessary ...* [60]

As a consequence:

- the measurement setup and the processing of data, for legal purposes, must be carried out consistently with the ISO norms;
- any other measurement, not complying with ISO standards, may not be used in court cases as evidence.

The vibration limits established by the EU Directive are depicted in Table 5.1.

5.3.3 The City Administrator Point of View

The sensitivity of citizens to the road conditions theme is well acknowledged.
However, an updated mapping of potholes within the town boundaries is useful to any city administrator. The availability of such mapping is a valuable tool to city administrators, for planning and organize road maintenance. The potholes (see Figure 5.5) are caused mainly by heavy rains, whose frequency is increasing due to climate changes.
A recent study of *Edinburgh Napier University* researchers reveals that:

> ... Poor road surfaces are exposing cyclists to vibration-related injuries more commonly found among people working with vibrating machinery ...

It should be taken into account that in some countries, like India, the problem of potholes killing cyclist reaches the dramatic peak of **ten deaths a day** [46].

Figure 5.5: Potholes on a Public Road

There is a clear correlation between the vibration absorbed by a cyclist and the status of road surface. Such a correlation hints for a device transforming accelerometer data into a meaningful and widely accepted road quality index, such as the **I**nternational **R**oughness **I**ndex (`IRI`). With appropriate precaution on user security and privacy rights, the big data from V-AUDIT devices installed on a fleet of bikes, could offer an answer to this need.

Figure 5.6: Cyclist with back pain

5.3.4 The Medical Point of View

There is a statistical evidence that exposure to whole-body vibration during cycling increases the risk of various ailments, including low back pain (Figure 5.6), spinal degeneration, hand-arm vibration related pathologies (see Figure 5.7) and more.

Edwards and Holsgrove [50] report that

> *..., it has been reported that in 2008–2009, 58 % of professional cyclists interviewed had experienced low back pain in the previous 12 months (Clarsen et al., 2010 [29]), and recreational cyclists who ride more than 160 km per week are 3.6 times more likely to have back pain compared to those who ride less than 160 km per week* (Schultz & Gordon, 2010 [31]) .

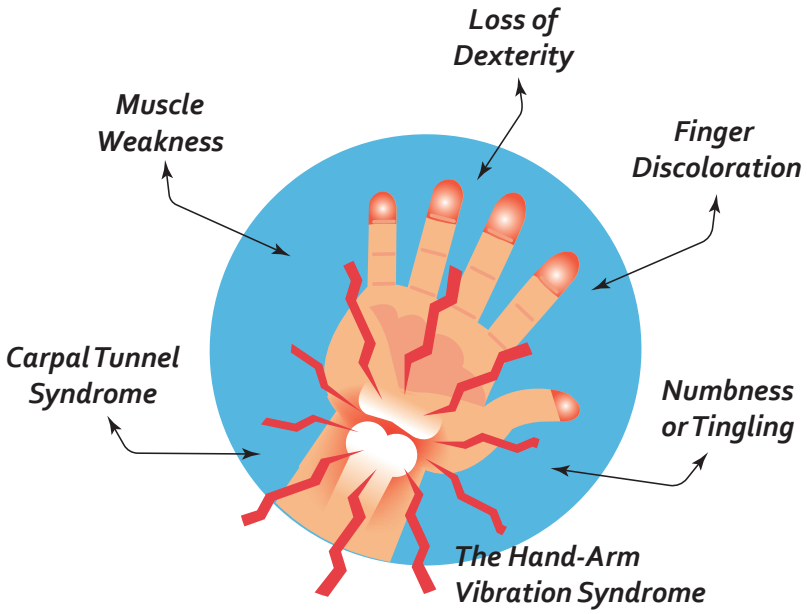


Figure 5.7: Hand-Arm Vibration Syndrome

In the context of vehicle drivers or heavy machinery operators, the dangers of high level of whole-body vibrations are investigated and well known. Since cycling regularly expose riders to vibration, it may cause similar potential risks of back pain. At scientific level, the topic is actively investigated by many years and meaningful contributions are on the record. More recently, the topic become of interest among recreational bikers and some companies started to use it as a *marketing tool* to promote their products [61, 55].

5.4 Bikers Comfort & Vibration Monitoring

5.4.1 Literature review

For the measurement of dynamic comfort in cycling, Olieman *et al.* [33], mounted wireless inertia accelerometers sensors on front wheel axel, rear wheel axel, stem and seatpost. They observed that not only road surface and speed, but also tire pressure has an influence on the measured vibration. The measurements have been done on short laps consisting of multiple types of pavements and three cruise

speeds (18 km/h, 25 km/h and 35 km/h). The sensors used were the ProMove2 from Inertia Technology.[2] The system collected accelerometer data with an ODR of 200 Hz, late post processed on `Matlab`.

Vanwalleghem *et al.* [34], established an absorbed power criterion to assess the cyclist's comfort. According to the authors, this method requires an advanced instrumentation and sensors, but gives more information on the man-machine interface corresponding comfort level, when compared to `ISO 2631` or `BS 6841` [16]. The sensor data, sampled at 1652 Hz rate, have been acquired through National Instruments DAQ modules and processed with the LabView. The entire instrumentation was placed in rucksack with a weight of 9.6 kg.

Chiementin *et al.* [35], within the framework of `ISO 5349-1`, measured the propagation of vibration in the hand-arm system of a cyclist. Their data acquisition system is based on OROS OR 36 (about 5 kg plus battery) [3] carried on the shoulders of the cyclist, as shown in Figure 5.8. The ODR of the system is 20 kHz. The measurement time interval is 5 s. Two tri-axis sensor 4525 B from Bruel and Kjaer are used. These have been placed on the bicycle stem and biker wrist, respectively. The field test on Reims paved street evidenced that the `ISO 5349-1` surveillance threshold limit (2.5 m/s^2) was reached at 5 km/h, whereas the risk threshold limit (5 m/s^2) was overcome of more than three times at 30 km/h. All the data have been at the lab post processed.

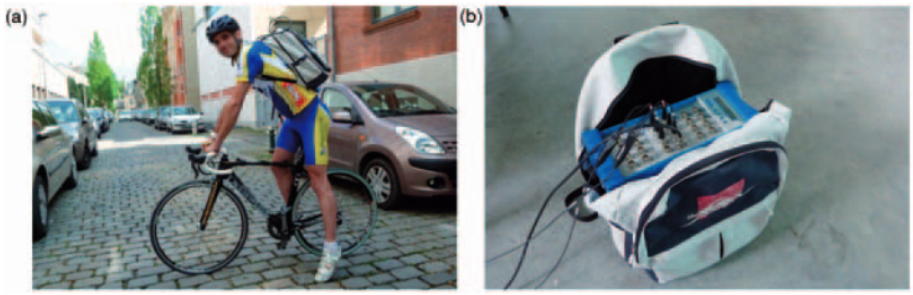


Figure 5.8: Test on a paved street and OROS OR 36 acquisition system [35]

Petrone and Trabacchin [37], designed a test setup to measure and simulate, on a group of real cyclists during steady cycling, the different vibrational patterns to the seatpost due to different road surfaces. The work bench, shown in Figure 5.9, is made of a racing bike frame without the wheels *hinged at the front fork axle and sustained by a hydraulic actuator connected to the seatpost, below the saddle clamp.*

Drouet *et al.* [36, 38], motivated by the interest of bicycle manufacturers in the the dynamic comfort of road bicycles, prepared the laboratory setup shown in Figure 5.10. The bicycle and cyclist were suspended with bungee cables on a custom-made bicycle treadmill. The treadmill, whose speed was set to 15 km/h, had dowels to generate impacts. Front and rear wheel were in contact with the treadmill and a uprised support, respectively. A strain gauge instrumented brake hood and a PCB 352C68 uniaxial accelerometer monitored the force and the acceleration transmitted to the cyclist's hands. The force and

[2] Inertia Technology B.V., Hengelosestraat 583, 7521 AG Enschede, The Netherlands `https://inertia-technology.com`

[3] `https://www.oros.com/solutions/instruments-accessories/or36-mobipack-16-channels-teamwork-analyzer-recorder/`

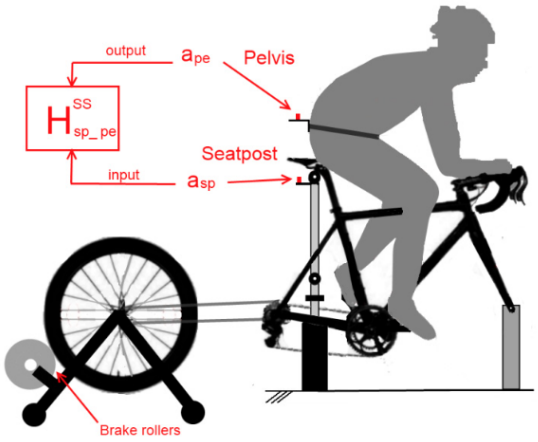


Figure 5.9: Schematics of the seat shaker test bench, in the configuration with the human subject [37]

acceleration signals were collected using a LMS SCADAS 24-bit acquisition system (model SCR01-08B) at a sampling frequency of 8192 Hz. LMS Test.Lab software was used for data processing. In particular, the system computed the instantaneous power absorbed at cyclist's hands.

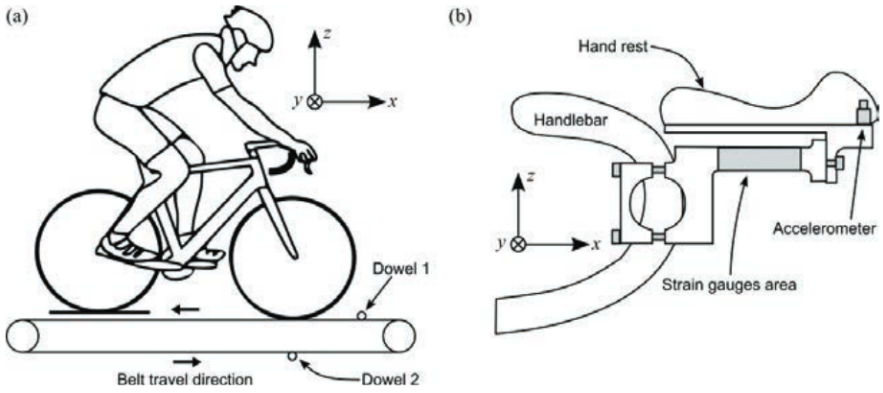


Figure 5.10: (a) Bicycle treadmill with two dowels attached to the belt; (b) Instrumented brake hood [38]

Taylor *et al.* [45, 44], analyzed cyclist exposure to hand-arm vibration. The data are collected by means of Raspberry Pi 3 (Model B) and post processed in Matlab. The experimental set up is depicted in Figure 5.11.

Gogola [51], with the purpose of assessing the cyclists comfort, proposed an entry-level analysis of bik-

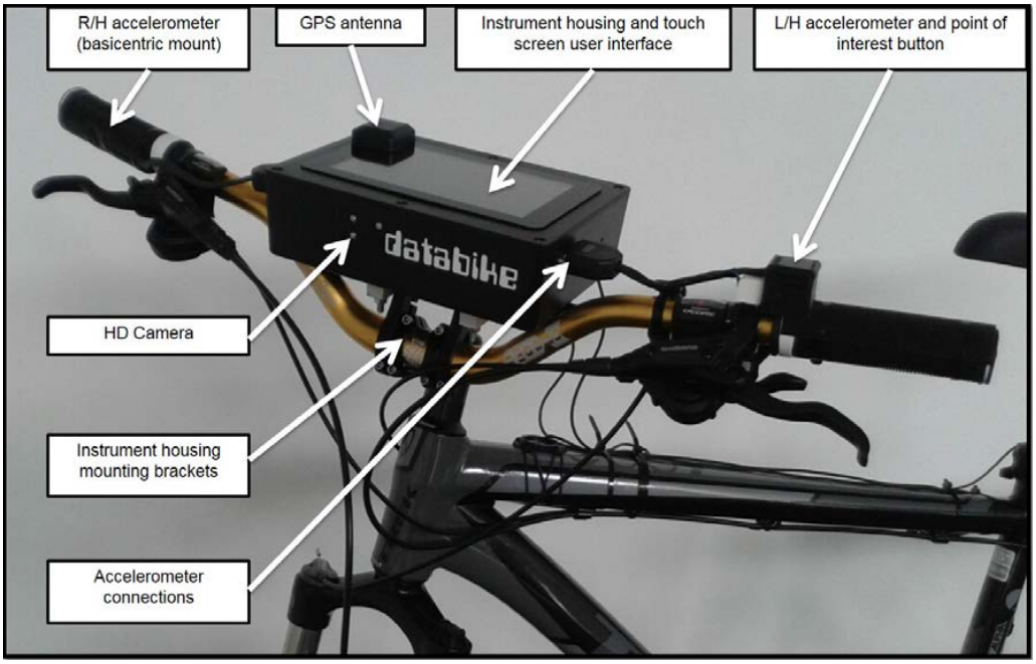


Figure 5.11: Databike of Taylor *et al.* [45]

ers vibrations based on the use of smartphones running `Phyphox`[4] and fitness applications. Different types of road surfaces have been tested: cycletrack paving, unpaved natural trail, residential road with a smooth surface, cracked residential road. However, the author recognizes that his method is not as precise as professional tools. It can only provide rough overview of the human vibration and road quality.

Doria *et al.* [49], developed a low-cost experimental-numerical method for the comfort prediction of city bicycles. The proposed method was validated both on lab and road tests. In particular, in the lab test the front wheel was placed on a vibrating table and accelerations measured at seatpostand steer tube, as shown in Figure 5.12.

5.4.2 Whole Body Vibration: The `ISO 2631` Norm

The primary purpose of `ISO 2631` is to define methods and criteria for evaluation of **whole-body vibration** in relation to:

- human health and comfort;
- the probability of vibration perception;
- the incidence of motion sickness.

[4]This software, developed at RWTH Aachen University, allows to access smartphone sensor data. It is available at `https://phyphox.org`.

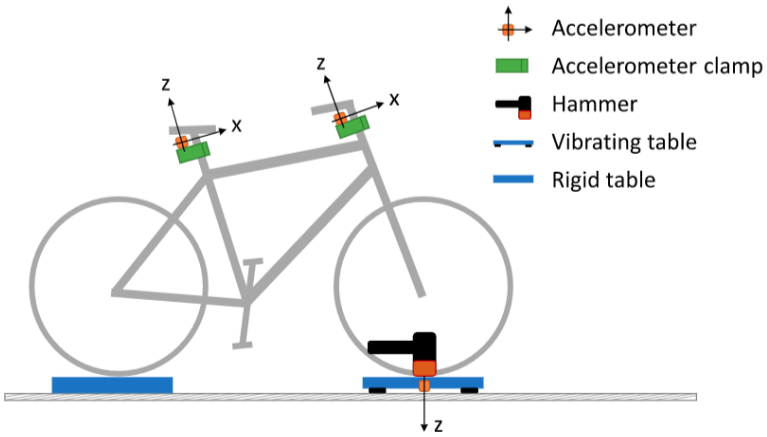


Figure 5.12: Experimental laboratory setup scheme of Doria *et al.* [49]

In particular, the norm covers the periodic, random and transient vibrations transmitted to the human body as a whole through the supporting surfaces such as the seat. For health, comfort and perception assessment, the frequency range considered by the norm is 0.5 - 80 Hz.
Regarding the position of transducers (accelerometers), the norm prescribes that [18]:

> *Vibration which is transmitted to the body from a non-rigid or resilient material (e.g. the seat cushion or couch) shall be measured with the transducer interposed between the person and the principal contact area of the surface. This should be achieved by securing the transducer within a suitably formed mount. The mount shall not greatly alter the pressure distribution on the surface of the resilient material. For measurements on non-rigid surfaces, a person shall adopt the normal position for the environment.*

These rules will be strictly followed when positioning the accelerometer on the seat. The data of the seat accelerometer, are filtered and processed to compute the **V**ibration **D**ose **V**alue (`VDV`). The computation flow of this **health index** is represented in Figure 5.13. The weighting filters are discussed in Section 5.4.4. For numerical testing purposes, a `VDV` computation (considering accelerometer data along z axis only) by means of a Simulink model has been herein developed. The simulation blocks are shown in Figure 5.15. For the orientation of Cartesian axis, refer to Figure 5.14 from the norm `ISO 2631`. The `ISO 2631` norm has been programmed in `Matlab`. The Figure 5.16 depicts the computation of VDV with IMU accelerometer data from `https://www.vibrationdata.com`.
The `ISO 2631` evaluates two methods for vibration assessment:

1. **The weighted RMS acceleration** (Unit: $\frac{\mathrm{m}}{\mathrm{s}^2}$) calculated in accordance with the following equation

$$a_w = \left[\frac{1}{T} \int_0^T a_w^2(t)\,\mathrm{d}t \right]^{\frac{1}{2}} \tag{5.4}$$

2. **The Fourth Power Vibration Dose Value** (Unit: $\frac{\mathrm{m}}{\mathrm{s}^{1.75}}$) calculated in accordance with the following

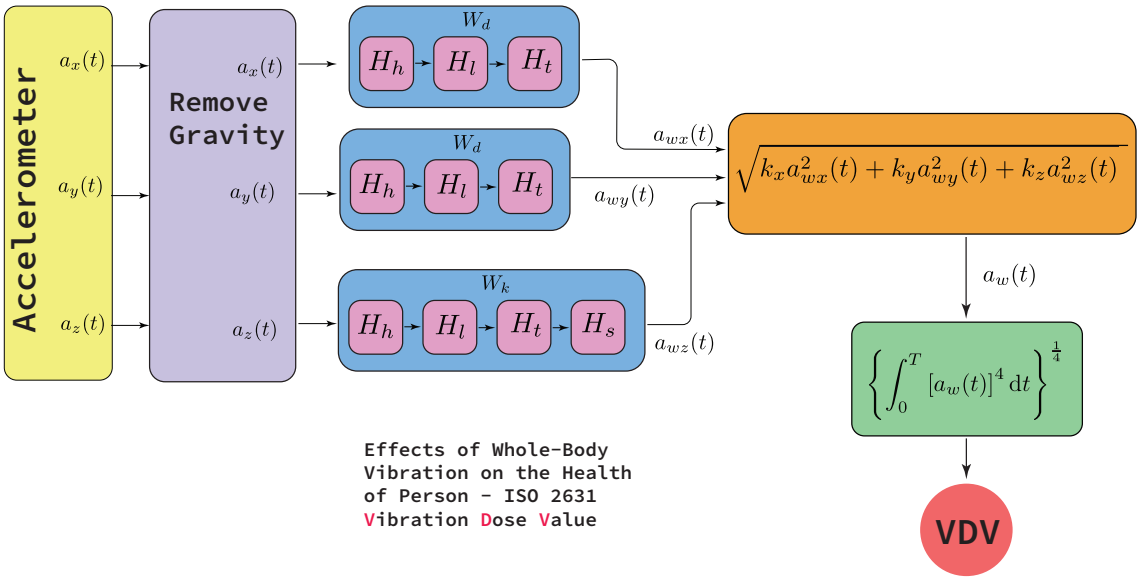


Figure 5.13: Vibration Dose Value for Whole-Body Vibration assessment: $k_x = k_y = 1.4$, $k_z = 1$ (ISO 2631)

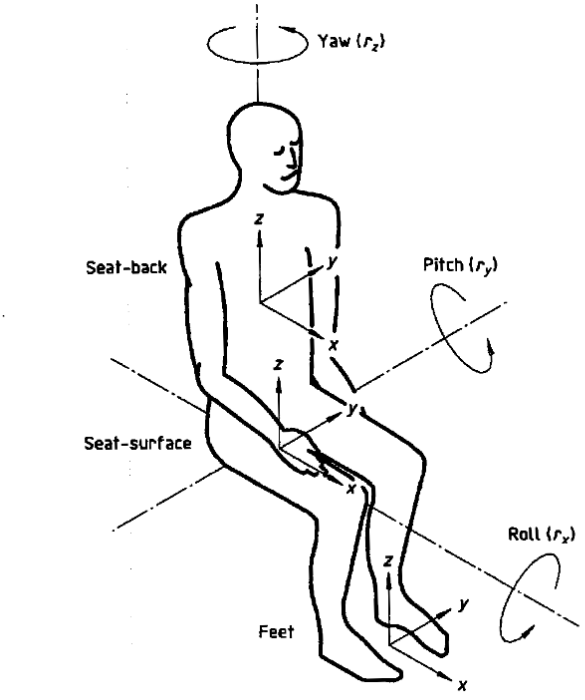


Figure 5.14: ISO 2631 Cartesian axis orientation [18]

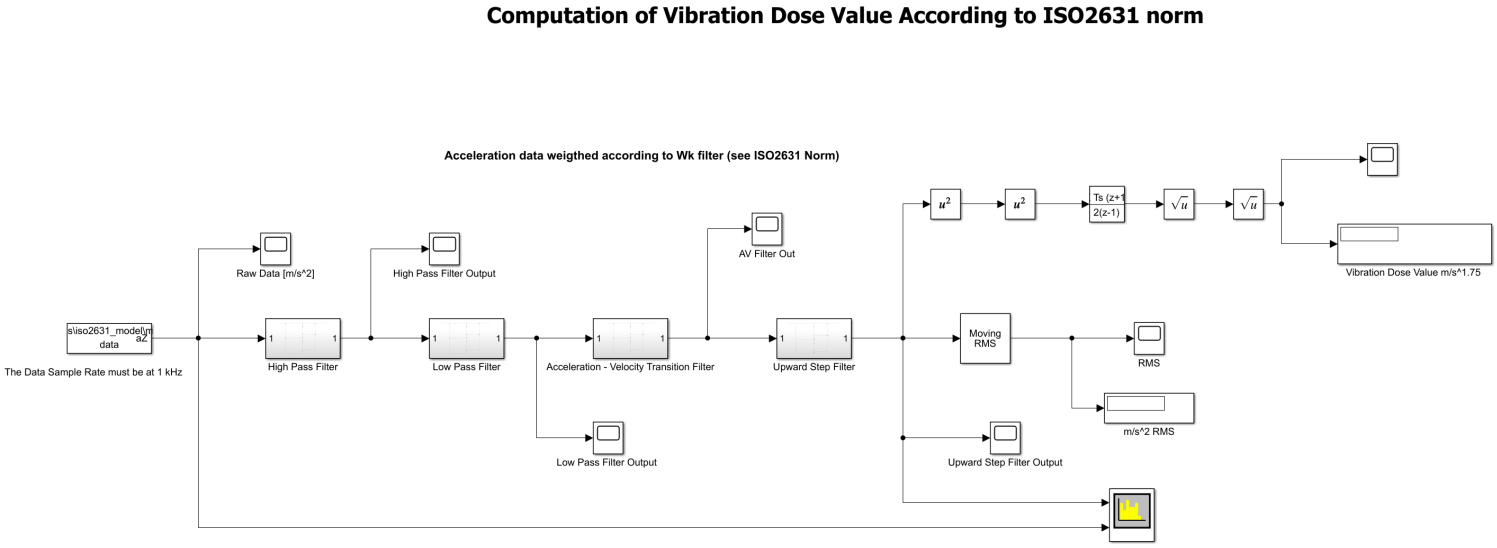


Figure 5.15: Proposed Simulink model of the VDV considering accelerometer data along z axis only

equation

$$\mathrm{VDV} = \left\{ \int_0^T [a_w(t)]^4 \, \mathrm{d}t \right\}^{\frac{1}{4}} \tag{5.5}$$

where:

- $a_w(t)$ is the instantaneous frequency-weighting acceleration;
- T is the duration of measurement.

When the vibration exposure consists of two or more periods, i, of different magnitudes, the vibration dose value for the total exposure should be calculated as follow [18]

$$\mathrm{VDV}_{\mathrm{total}} = \left(\sum_i \mathrm{VDV}_i^4 \right)^{\frac{1}{4}} \tag{5.6}$$

The projected `VDV` on a time period of 8 hours is

$$\mathrm{VDV}(8\mathrm{hours}) = \mathrm{VDV} \sqrt[4]{\frac{T_{\mathrm{exposure}}}{T_{\mathrm{measured}}}} \tag{5.7}$$

5.4.3 Hand-Arm Vibration: The `ISO 5349` Norm

The primary purpose of the `ISO 5349` is to provide a method for the assessment of the hand-transmitted vibration exposure, specified in terms of a frequency-weighted vibration acceleration and daily exposure time [25]. This standard is applicable to periodic and to random or non-periodic vibration. The vibration assessment is achieved by introducing the **Vibration Total Value** a_{hv} measured in $\frac{\mathrm{m}}{\mathrm{s}^2}$. This index follows by processing the frequency-weighted r.m.s. acceleration value measured for axis x, y and z. The axes orientation is reported in Figure 5.18, while the computation flow for a_{hv} is depicted in Figure 5.17.
To compare vibration total values of different duration, the standard recommends to express them in terms of an 8 hour energy-equivalent frequency-weighted vibration total value $A(8)$ defined as [25]:

$$A(8) = a_{hv} \sqrt{\frac{T}{T_0}} \tag{5.8}$$

where:

- T is the exposure duration to the vibration a_{hv};
- T_0 is the reference duration period of 8 hour.

If the total vibration exposure consists of several operation, then $A(8)$ is defined as [25]:

$$A(8) = \sqrt{\frac{1}{T_0} \sum_{i=1}^{N} a_{hvi}^2 T_i} \tag{5.9}$$

where:

- a_{hvi} is the vibration total value for the i^{th} operation slot;
- n number of the individual vibration exposure;
- T_i duration of the i^{th} operation slot.

5.4.4 A Preliminary design of Digital Weighting Filters

Both `ISO 2631` and `ISO 5349` require the raw data, collected from the accelerometers, to be preliminary processed by weighting filters. Frequency weighting on acceleration reflects the assumed importance of different frequency in causing health. There is a general agreement that the frequency ranges of interest are:

- **Whole body vibration**: 0.5 - 80 Hz (health, comfort and perception) [18];
- **Hand-Arm vibration**: 8 - 1000 Hz [25]

Band-limiting filters restrict the effects of the measured values of vibration frequencies. Outside these ranges the frequency dependence on health is not yet agreed among scientists.
The `ISO 2631` norm specifies 6 different weighting filters: W_k, W_d, W_f, W_c, W_e and W_j (see Equations (5.15)). However, for health assessment purposes only the W_k and W_d filters are required.
The `ISO 5349` specify only one weighting filter W_h ((5.15c)).
These filters are respectively defined as a cascade of second order filters: H_h, H_l, H_t, H_s and H_w:

- **High Pass Filter**

$$H_h = \frac{1}{1 + \frac{\sqrt{2}\omega_1}{s} + \frac{\omega_1^2}{s^2}} \tag{5.10}$$

- **Low Pass Filter**

$$H_l = \frac{\omega_2^2}{s^2 + \sqrt{2}\omega_2 s + \omega_2^2} \tag{5.11}$$

- **Acceleration - Velocity Transition Filter**

$$H_t = \frac{1 + \frac{s}{\omega_3}}{1 + \frac{s}{Q_4\omega_4} + \frac{s^2}{\omega_4^2}} \tag{5.12}$$

- **Upward Step Filter**

$$H_s = \frac{\left(\frac{s^2}{\omega_5^2} + \frac{s}{Q_5\omega_5} + 1\right)\omega_5^2}{\left(\frac{s^2}{\omega_6^2} + \frac{s}{Q_6\omega_6} + 1\right)\omega_6^2} \tag{5.13}$$

- H_w

$$H_w = \frac{(s + \omega_7)\,\omega_8^2}{\left(s^2 + \frac{\omega_8}{Q_7}s + \omega_8^2\right)\omega_7} \tag{5.14}$$

where $\omega_i = 2\pi f_i$ and f_1, f_2, f_3, f_4, f_5, f_6, Q_4, Q_5, Q_6 are specified for each weighting filter in Table 5.2.

$$W_k = H_h(s)H_l(s)H_t(s)H_s(s) \tag{5.15a}$$
$$W_d = H_h(s)H_l(s)H_t(s) \tag{5.15b}$$
$$W_h = H_h(s)H_l(s)H_w(s) \tag{5.15c}$$

In this section, a preliminary study on the digitalization of the weighting filters is carried out. The discretization of the filters is carried out by means of the *bilinear transform*; the z-transform of H_h, H_l, H_t, H_s is reported in the following:

- High Pass Filter
$$H_{hd} := \frac{4z^2 - 8z + 4}{\left(2\sqrt{2}\Omega_1 + \Omega_1^2 + 4\right)z^2 + \left(2\Omega_1^2 - 8\right)z - 2\sqrt{2}\Omega_1 + \Omega_1^2 + 4} \tag{5.16}$$

- Low Pass Filter
$$H_{ld} := \frac{\Omega_2^2 z^2 + 2\Omega_2^2 z + \Omega_2^2}{\left(2\sqrt{2}\Omega_2 + \Omega_2^2 + 4\right)z^2 + \left(2\Omega_2^2 - 8\right)z - 2\sqrt{2}\Omega_2 + \Omega_2^2 + 4} \tag{5.17}$$

- Acceleration - Velocity Transition Filter
$$\begin{aligned} H_{td}^{Num} &= Q_4\Omega_4^2(\Omega_3 + 2)z^2 + \left(Q_4\Omega_4^2(\Omega_3 + 2) + Q_4\Omega_4^2(\Omega_3 - 2)\right)z \\ &\quad + Q_4\Omega_4^2(\Omega_3 - 2) \\ H_{td}^{Den} &= \left(Q_4\Omega_4^2 + 4Q_4 + 2\Omega_4\right)\Omega_3 z^2 + \left(2Q_4\Omega_4^2 - 8Q_4\right)\Omega_3 z \\ &\quad + \left(Q_4\Omega_4^2 + 4Q_4 - 2\Omega_4\right)\Omega_3 \\ H_{td} &= \frac{H_{td}^{Num}}{H_{td}^{Den}} \end{aligned} \tag{5.18}$$

- Upward Step Filter
$$H_{sd1} := \frac{Q_6\left(\left(Q_5\Omega_5^2 + 4Q_5 + 2\Omega_5\right)z^2 + \left(2\Omega_5^2 - 8\right)Q_5 z + Q_5\Omega_5^2 + 4Q_5 - 2\Omega_5\right)}{Q_5\left(\left(Q_6\left(\Omega_6^2 + 4\right) + 2\Omega_6\right)z^2 + \left(2\Omega_6^2 - 8\right)Q_6 z + Q_6\left(\Omega_6^2 + 4\right) - 2\Omega_6\right)} \tag{5.19}$$

where Ω_i is the normarlized[5] warped frequency defined as

$$\Omega_i = 2\tan\left(\frac{\pi f_i}{f_s}\right) \tag{5.20}$$

The comparison between the continuous and discrete transfer functions of the weighting filters are depicted in Figures 5.19 - 5.20.
Since all the current applications of `ISO 2631` rely upon **post-processing** of collected data, the implementation of the weighting filters is software based for general purpose PC (e.g. coded in `C` or `Matlab`).

[5] The system sampling frequency is f_s

Weighting	Band Limiting		Acc.-Vel. Transition			Upward step				H_w		
	f_1 Hz	f_2 Hz	f_3 Hz	f_4 Hz	Q_4	f_5 Hz	Q_5	f_6 Hz	Q_6	f_7 Hz	f_8 Hz	Q_7
W_k	0.4	100	12.5	12.5	0.63	2.37	0.91	3.35	0.91	-	-	-
W_d	0.4	100	2.0	2.0	0.63	∞	-	∞	-	-	-	-
W_h	6.31	1258.9	-	-	-	-	-	-	-	15.915	15.915	0.64

Table 5.2: Transfer function parameters of the weighting filters

Our intended approach is novel and different:

1. the biker vibration monitoring is made **on board**, with a device that can be embedded in the bicycle, to immediately warn the user of potential health risks;
2. human vibration is assessed on the basis of data acquired within a *limited time interval* (minutes), our procedure makes **daily exposure measurements** with continuous data acquisition;
3. the target is to implement a **low cost**, **energy efficient**, **battery powered** unit to monitor vibration. The implementations hardware is meant to be an SOPC.
4. the modularity of our design, based on low cost and widely available off-the-shelf components, **allows the extension** of this study outcome to other devices and vehicles.

5.5 Road conditions assessment

The assessment of road surface roughness is an important task periodically carried out on motorable roads with many measuring techniques and tools. More recently, the attention to the conditions of unmotorable roads for pedestrians, wheel-chair users, bikers is substantially grown and legislation guidelines have been issued [6].

The potholes can be a danger to cyclists and others. The proliferation of smartphones has brought new data sources as well as interesting initiatives, like the one of Cycling UK. By means of the app `FillThatHole`[7] the users can report potholes and road defects. Cycling UK contacts Councils to get the road repaired.

The smartphones can easily collect data, but their use has several challenges, such as limited mode inference, sample bias, privacy issues. The problems associated with the adoption of these technologies for bicycle conditions monitoring have been reviewed by Lee & Sener [53].

With reference to the measurement of road roughness with bike mounted devices, a description of previous studies, is provided in the next section.

5.5.1 Previous studies

A pioneer study about road track roughness and bike usage has been carried out by Wigan and Cairney [13] of the Australian Research Board. The panel riders were not given any specific instructions about speed, other than a request to ride as they normally would, at an appropriately comfortable speed. The experimental setup consisted of an accelerometer, an amplifier, FM modulation circuits, and oscillator, together with a battery. A SONY Walkman cassette to record analog signals. Spectral analysis was used to examine the nature of the vibrations recorded by the accelerometer. A Hewlett-Packard spectrum analyzer was used to determine the power distribution of the accelerometer signal against the frequency of the vibrations.

[6] `https://safety.fhwa.dot.gov/ped_bike/tools_solve/fhwasa12018/`
`https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/510268/national-propensity-to-cycle-full-report.pdf`

[7] `https://www.fillthathole.org.uk/`

Qian and Nietmeier [43] proposed to measure vertical distance from bicycle shaft bottom to ground, as shown in Figure 5.21.

Zang *et al.* [47] used bicycle-mounted smartphones to measure the surface roughness of roads as well as to identify potholes and humps along the roads.

Their method for the `IRI` measurement is based on a rigid model. In particular, the vertical displacement is obtained by the double integration of vertical acceleration

$$\sum_i h_i = \int \left(\int |a_v| \, \mathrm{d}t \right) \mathrm{d}t \tag{5.21}$$

At the beginning of the acquisition process the bicycle is stationary and only the acceleration gravity (1 g) is recorded, therefore

$$\bar{a}_x \bar{a}_x + \bar{a}_y \bar{a}_y + \bar{a}_z \bar{a}_z = 1 \tag{5.22}$$

where $\bar{a}_x$, $\bar{a}_y$ and $\bar{a}_z$, are the average acceleration values along x, y, and z axes during 5 seconds, as obtained from the accelerometer sensor on the smartphone.

In particular, taking into account Eq. (5.22), the value of a_v is computed as follows:

$$a_v = \frac{a_x \bar{a}_x + a_y \bar{a}_y + a_z \bar{a}_z}{\sqrt{\bar{a}_x \bar{a}_x + \bar{a}_y \bar{a}_y + \bar{a}_z \bar{a}_z}} = a_x \bar{a}_x + a_y \bar{a}_y + a_z \bar{a}_z \tag{5.23}$$

The bicycle riders were requested to keep the speed and the bicycle posture as stable as possible. A high-pass filter was applied (setting the value cutoff frequency as 0.5 Hz) to clean the sensor data when calculating vertical acceleration. The study proposes an algorithm for the recognition of potholes and bumps on roads.

A study from Toljic *et* al. [59] from Technische Universität Wien, monitored the cyclist speed under different conditions (age, hours, surface roughness, wind speed, etc.). For their study, the road roughness was computed from the vertical accelerations measured on a bicycle mounted smartphone (see Figure 5.23) vertical acceleration.

The bike had a weight of about 15 kg and tire pressure was 4.5 bars (checked before each series of tests). The test tracks lengths were 55 m, wich equals a measuring duration of about 10 s. The sensor had a sampling rate of 50 Hz. The phone was inclined by 10° and some corrections were required also to subtract the acceleration gravity.

The point-to-point distances along each axis direction have been obtained by means of a double numerical integrations of $a_j(t)$ ($j = x, y, z$)

$$s_j = \int \left(\int a_j(t) \mathrm{d}t \right) \mathrm{d}t \tag{5.24}$$

The post processing of data demonstrated some discrepancies, such as a low ranking of big regular coblestones. Likely this is due to the studs of the tire, causing more vibrations on a flat surface like asphalt, that penetrate the gaps between the cobblestones, thereby avoiding additional vibrations.

Shtayat *et al.* focused on evaluating pavement conditions using simple and widely available monitoring equipment. In particular, a smartphone application and bicycle as a test vehicle were used to measure pavement vibration along a road segment [54].

All the studies reviewed concluded that an extensive work needs to be done in this area of research.

5.5.2 International Roughness Index - IRI

It is well known that the quality of transportation infrastructure affect bikers safety [28]. The presence of potholes, cracks, patches, and bumps and more generally road surface roughness, affect people's riding experience and could be the cause of accidents. Enhancing the road infrastructure is therefore of major importance for city administrators as insufficient infrastructure is a significant barrier for cycling. Hardinghaus & Papantoniou [52] state that:

> ...infrastructure investments are the main focal point of municipal cycle strategies. To meet the demand of cyclists, it is crucial to have a good understanding of cyclists' route choice behavior.

Their investigation has shown that the availability of well maintained and dedicated bike infrastructure along main streets appears to be a stable strategy, regardless of individual and local characteristics.
The American Society of Testing and Materials (ASTM) defines road roughness as follows:

> *The deviations of a pavement surface from a true planar surface with characteristic dimensions that affect vehicle dynamics, ride quality, dynamic loads, and drainage, for example, longitudinal profile , transverse profile, and cross slope.*

More generally, according to the ISO 4287 - 1997 [19], given a profile, as shown in Figure 5.24, its arithmetic roughness R_a is defined as:

$$R_a = \frac{1}{L}\int_0^L \left| y(l) \right| \mathrm{d}l \tag{5.25}$$

where $y(l)$ is the absolute ordinate value and L is the sampling horizontal length. Clearly the one provided above is a very general definition that applies to any profile.

There are different methods for assessing the road pavement state (e.g. ISO 8608 norm [40]), but a widely accepted road roughness indicator is the **I**nternational **R**oughness **I**ndex (IRI), introduced in the early 1980 by the highway engineering community [9, 15].

The method is based on the so called -quarter-car model shown in Figure 5.25. The IRI is an accumulation of the simulated relative motion between sprung and unsprung masses, normalized by the length L of the profile:

$$IRI = \frac{1}{L}\int_0^{\frac{L}{V}} |\dot{x}_1 - \dot{x}_2| \, \mathrm{d}t \tag{5.26}$$

where:

- x_1 and x_2, are the vertical displacements of sprung and unsprung masses;
- V is the vehicle speed (80 kmh):
- L the road length.

The state of road surface is a matter of concern for local authorities. In developed countries, the **P**avement **M**anagement **S**ystems (PMS) handle many duties of road maintenance and rely upon sophisticated and expensive equipment installed on special vehicles. However, cost effective solutions to assess the road pavement status is always a pursued research topic. In particular, the World Bank terminology classifies measurement devices according to the following Information Quality Level (IQL) scale [56]:

- (IQL) 1: roughness reported at 10-20 m intervals;

- (IQL) 3: roughness reported at more than 100 m intervals;
- (IQL) 4: most cell phone based systems.

Since our goal is the development of a low cost device, the IRI assessment methods, based on sophisticated measurement systems (e.g. profiler based on ultrasonic or laser) mounted on passenger cars, have been herein avoided. In fact, the speeds of these vehicles are not comparable with those of bicycles.

Our main focus is oriented toward a cost effective solution to assess the road pavement status by measuring the vehicle's vibration with a inertia sensor. A good example is offered by the system proposed by Havinga *et al.* [39] where the pervasive nature of smartphone devices to collect, process and share these data was used to detect road anomalies.

The roadways roughness is a matter of concern for any cyclist. This greatly influences the bikers' safety as well as the velocity choice.

For passenger car vehicles, an important investigation by Ahlin & Granlund [20] of Swedish National Road Administration, found a close correlation between the whole body vibration index, computed with the ISO 2631, and the reference ride quality criterion such as the International Roughness Index. The study concluded to relate IRI-values to WBV-levels as follows:

> *The conversion from IRI to vertical WBV in a passenger car can often be done by a factor 0.16. But depending on actual roughness wavelength and vehicle speed, the conversion factor ranges from 0.04 to 1.4. This shows that there can be a difference of a factor 30 between the worst and the best ride, given the very same IRI-value.*

Unfortunately, a similar study is not yet available for bikes.

A common requirement of all the studies relating road roughness is an easily operable measurement setup. For instance, the measurement system proposed by Havinga *et al.* [39] implemented a wavelet decomposition analysis for signal processing of inertial sensor signals and Support Vector Machine (SVM) for anomaly detection and classification

One of the research tools would be an experimental setup, to be used by groups of cyclists, with different ages and bicycle types, passing along different test routes or tracks. The challenge of such a setup would be to prevent the cyclists from being biased and to simultaneously research the influence on comfort of different tire types and tire pressures.

5.5.3 Proposed model for the computation of road roughness

As discussed in a previous section, all the available methods for the estimation of a road profile by means of an accelerometer mounted on a bike, are based on a rigid model. A novel refined approach, herein proposed, could be based by taking into account the elasticity and damping of tires. On this regard, the experimental data collected and published by the Research Unit at Delft University[8] are of paramount importance for our purposes.

The proposed dynamic model, depicted in Figure 5.26, aims to obtain the road profile directly from the IMU accelerometer data.

The vertical motion of the mass, is governed by the following differential equation deduced through the application of Newton's law[9]:

[8] http://bicycle.tudelft.nl/tiretestingvertical/

[9] Since the computation of the road profile from acceleration requires the removal of gravity acceleration, the weight force has been omitted from the differential equation.

$$m\ddot{x} + c\left(\dot{x} - \dot{y}\right) + k\left(x - y\right) = 0 \tag{5.27}$$

where:

- m represents the cyclist and bicycle masses;
- x is the vertical displacement of the mass m;
- k and c represent tire stiffness and damping respectively;
- t is the time;
- $y(t)$ is the road profile function;
- dots represent differentiation with respect to time.

The first and second derivative of a discrete function f can be respectively expressed by means of the following backward finite difference formulas:

$$\dot{f}_i = \frac{f_{i-2} - 4f_{i-1} + 3f_i}{2\Delta t} \tag{5.28a}$$

$$\ddot{f}_i = \frac{-f_{i-3} + 4f_{i-2} - 5f_{i-1} + 2f_i}{\Delta t^2} \tag{5.28b}$$

where f_i is the function value at time $t = t_i$ and Δt is the differentiation step.
Taking into account equation (5.28a), the differential equation (5.27) can be discretized as follows:

$$m\ddot{x}_i + k(x_i - y_i) + \frac{c}{2\Delta t}\left(x_{i-2} - 4x_{i-1} + 3x_i - y_{i-2} + 4y_{i-1} - 3y_i\right) = 0\,. \tag{5.29}$$

Thus, the road profile at time t_i is deduced solving the previous one with respect to y_i

$$y_i = \frac{2\Delta t\,(m\ddot{x}_i + kx_i) + c\left(3x_i - 4x_{i-1} + x_{i-2} + 4y_{i-1} - y_{i-2}\right)}{2k\Delta t + 3c} \tag{5.30}$$

Moreover, from equation (5.28b), one obtains:

$$x_i = \frac{\ddot{x}_i\Delta t^2 + 5x_{i-1} - 4x_{i-2} + x_{i-3}}{2}\,. \tag{5.31}$$

The combined use of (5.30) and (5.31) allows the casting of the following recursive *finite difference* scheme to compute y_i from $\ddot{x}_i$:

1. Let $y_0, y_1, y_2 \leftarrow 0$
2. Let $x_0, x_1, x_2 \leftarrow 0$
3. **for** $i = 4 : \ldots$

$$x_i \leftarrow \frac{\ddot{x}_i\Delta t^2 + 5x_{i-1} - 4x_{i-2} + x_{i-3}}{2}\,.$$

$$y_i \leftarrow \frac{2\Delta t\,(m\ddot{x}_i + kx_i) + c\left(3x_i - 4x_{i-1} + x_{i-2} + 4y_{i-1} - y_{i-2}\right)}{2k\Delta t + 3c}$$

Compute traveled distance l_i from starting position

$i \leftarrow i+1$

4. **end for**

Finally, the roughness index is readily computed from (5.25):

$$R_a = \frac{1}{L}\int_0^L \left|y(l)\right| \mathrm{d}l$$

5.5.4 Numerical Validation

The following procedure has been adopted to validate the proposed finite differences scheme:

- **Simulate Acceleration Data**
  - A *test* road profile y_{test}, with two bumps, has been generated;
  - A constant bike speed v has been assumed;
  - By solving equation (5.27), the mass acceleration $\ddot{x}$ has been computed[10].
- **Application of the proposed method**
  - Estimate the road profile y_{est} by means of the proposed method
- **Compare** y_{est} and y_{test}.

The simulated acceleration and the comparison of y_{est} and y_{test} are plotted .vs. time in Figures 5.27 - 5.28, respectively.
The values of tire stiffness k and damping coefficient c

$$k = 183000 \left[\frac{\mathrm{N}}{\mathrm{m}}\right]$$
$$c = 200 \left[\frac{\mathrm{Ns}}{\mathrm{m}}\right]$$
$$m = 90 \left[\mathrm{Kg}\right]$$

are from the Delft University[11].

FIX-ME GIt of Matlab program

5.6 Prototype Hardware Components

In the following sections are listed the main hardware components to achieve the features described for V-AUDIT . These can be summarized in the followings:

- Collect data from at least 3 `IMU` platforms or other sensors/devices, installed on the bike;
- Process data to implement the functionalities previously described;
- Store Processed/Raw Data into an `SD Card`

[10] Before processing, in real life application, gravity must be removed from accelerometer data

[11] `http://bicycle.tudelft.nl/tiretestingvertical/`

5.6.1 System Requirements

The selected hardware platform should satisfy the following requirements:

- Handling of different sensors interfaces (I2C,SPI,CAN-bus, ...) ;
- Capability of simultaneously collect data from the different sensors/devices installed on the bike;
- Integrate/Process in parallel data from different sensors/devices;
- Should be able of handling DSP algorithms;
- Low power capabilities;
- A single chip should host the entire processing system (this does not refer to the accelerometer);
- Reconfiguration capability to host other application developed in the Smart Bike project.

The market offers different hardware platforms to implement embedded systems devices. In particular, the most widely adopted ones are:

- ASIC;
- DSP;
- Microcontroller;
- FPGA.

In the following, a brief evaluation of the different platform, according to the previously selected requirements, is made.

- **ASIC**. ASIC offers the lowest power consumption and lowest unit cost once the chip production is established. However, since our project is financial resource constrained, it will be not possible to start from scratch the development of a new ASIC . Moreover, an ASIC oriented development, violates the reconfiguration capability requirement of the platform.
- **DSP**. These devices offer an hard wired implementation of many digital signal processing algorithms, thus reducing the development time and the time-to-market. All these advantages, come at the expenses of a reduced flexibility. Our application requires the interface with different sensors/devices, thus making DSP unfeasible for our purpose.
- **Microcontroller**. These devices are best suited to implement sequential algorithms. Moreover, many vendors offer microcontrollers with embedded DSP to enhance signal processing applications. MCU are generally more cost effective and power efficient than FPGA. However, they are not feasible to perform parallel operations and are consistent with processing at high data-rate.
- **FPGA**. Thanks to their reconfigurability, FPGAs, offer the most flexible alternative where to develop the target application. In particular, major FPGA vendors offer the capability to embed into the FPGA silicon a processor core. This can be either *hard* (direct silicon implementation) or *soft* (implemented using FPGA sources). This approach is defined as System on a Programmable Chip (SoPC). The mixed design pattern allows a short development cycle. During the design process, features and specification, or reallocation of software and hardware partition, can be changed without big impact on the overall costs. This design methodology is recognized as economical in low to medium quantity production.

The table depicted in Figure 5.29 is a qualitative comparison between ASIC, DSP, Microcontroller and FPGA. The previous discussion motivated the decision of carrying out the present project as an FPGA-based SoPC. Since Cyclone V FPGA family ensures *the lowest system cost and power requirement for a wide spectrum of general logic and DSP applications*, the development activities will be carried out with two Cyclone V 5CEFA2F23 and 5CEFA5F23 boards.
To carry out this development experience, the FPGA bards from QMTECH [12] (see Figure 5.30) have been chosen. One of the main reason of this choice is the possibility of mounting different FPGA boards on the same expansion board.
The typical setup for debugging an IMU module is depicted in Figure 5.31.

5.6.2 IMU Placement

The IMU sensors on the bike handle bar and seat will be installed according to ISO 2631 and ISO 5349 standards. For the IRI computation, it was decided to install the sensor on the fork. The positions of the IMU sensors are depicted in Figure 5.32. To protect the IMU sensors from dust and damage, a preliminary design of the sensor shelter is depicted in Figure 5.33.

[12] http://www.chinaqmtech.com/

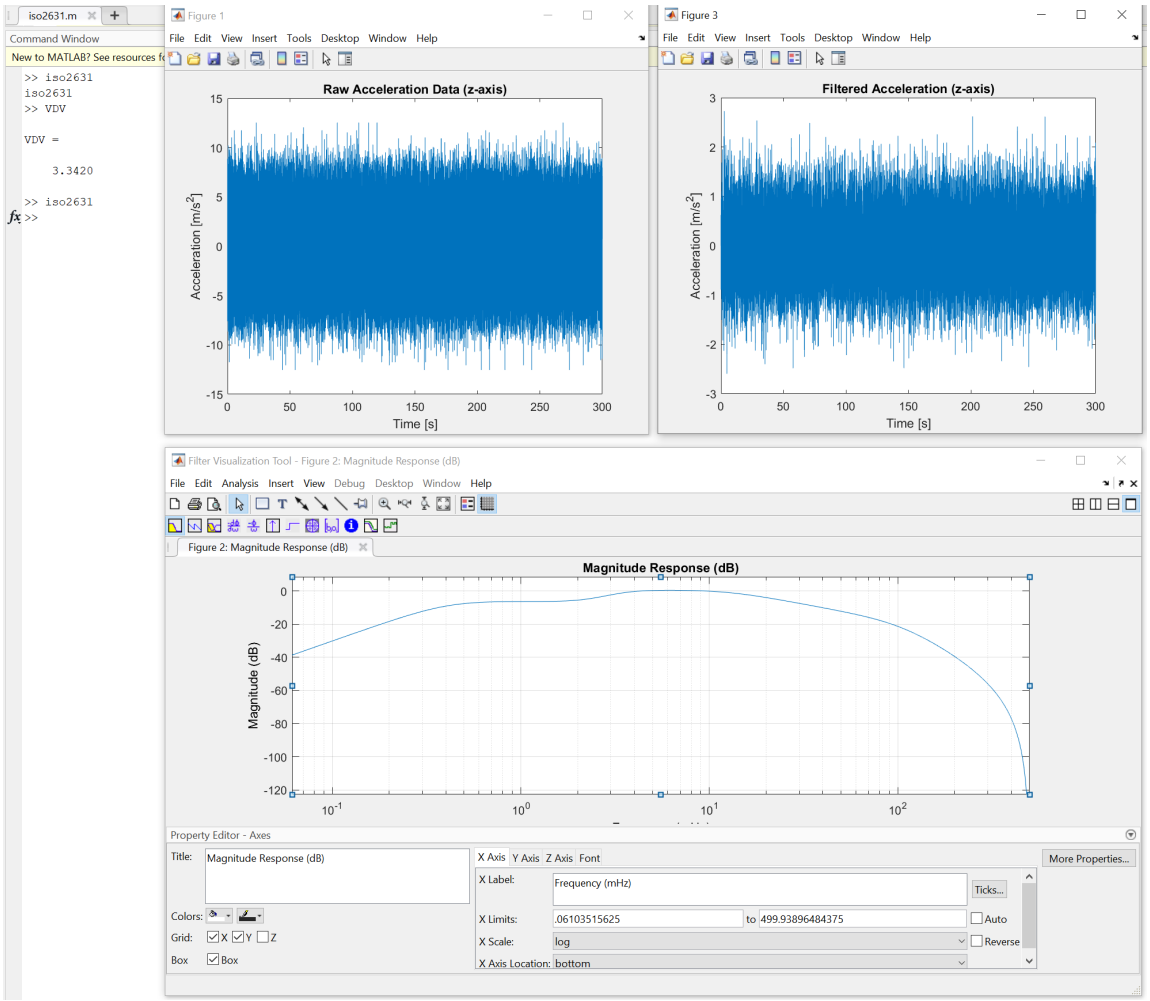


Figure 5.16: Vibration Dose Value `Matlab` computation.

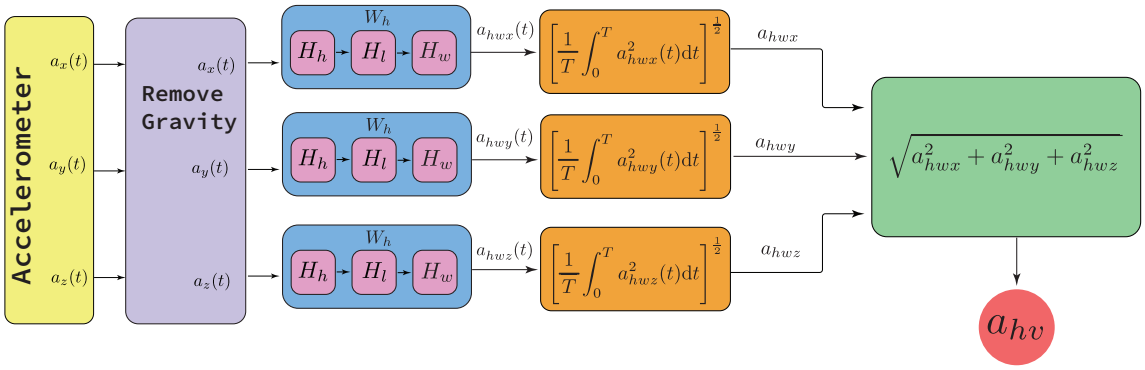


Figure 5.17: Computation Flow of the Vibration Total Value a_{hv}, ISO 5349

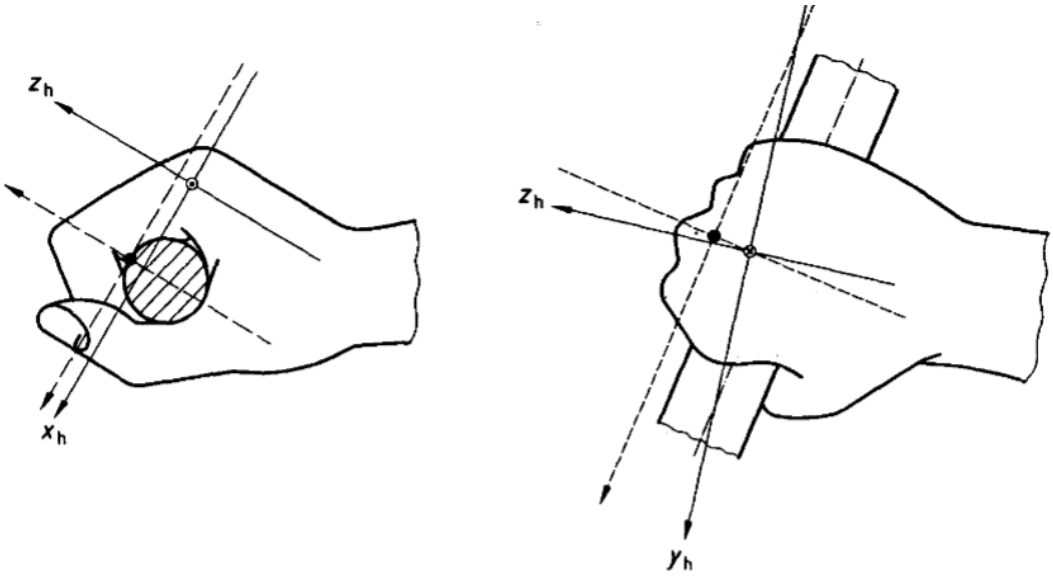


Figure 5.18: ISO 5349 Cartesian axis orientation [25]

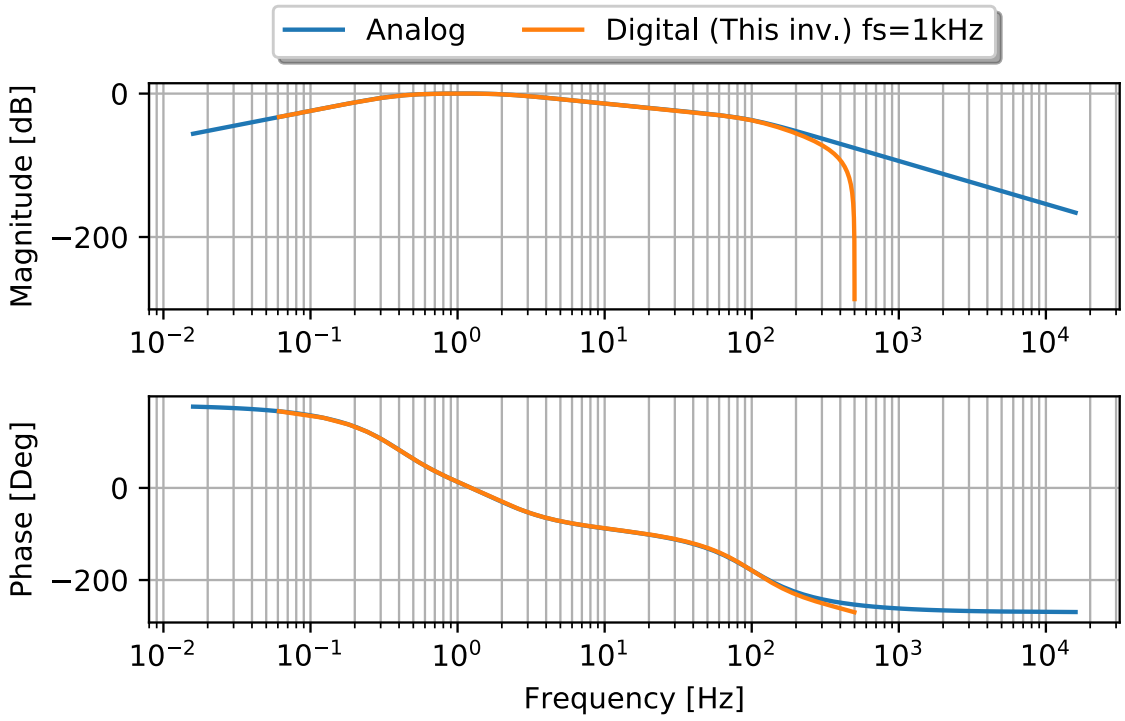


Figure 5.19: Weighting filter W_d

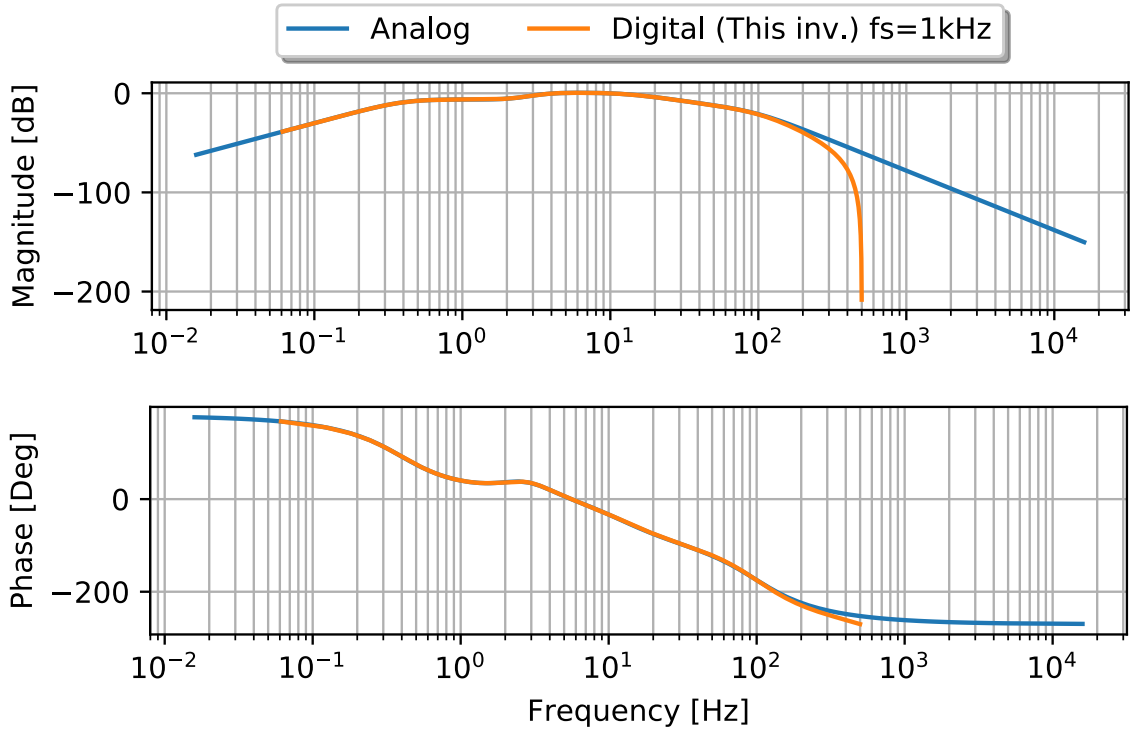


Figure 5.20: Weighting filter W_k

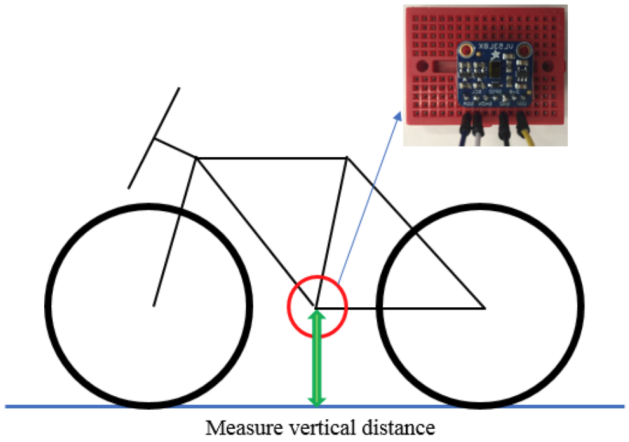
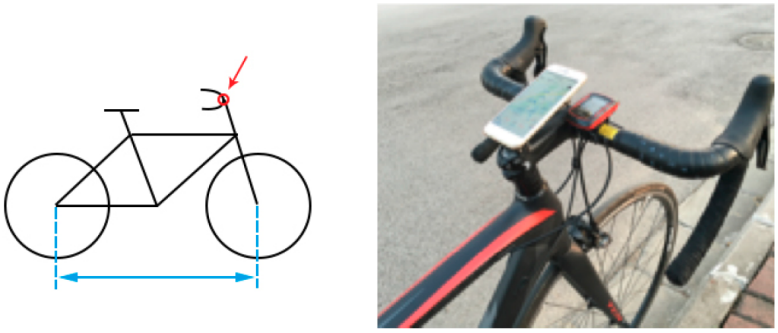


Figure 5.21: Qian and Dietmeier [43]: Positioning of accelerometers for measuring bike vertical displacements



(a) (b)

Figure 5.22: Installation of the smartphone to collect acceleration data according to Zang *et al.* [47]

Figure 5.23: Measuring accelerations with smartphone for road roughness analysis [59]

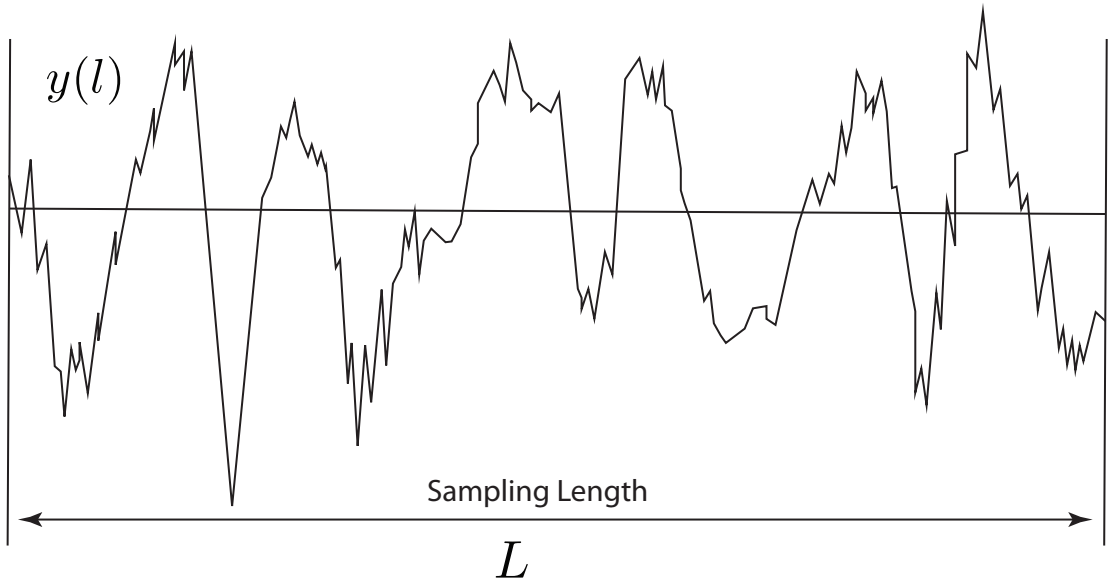


Figure 5.24: Example of roughness profile (adapted from [19])

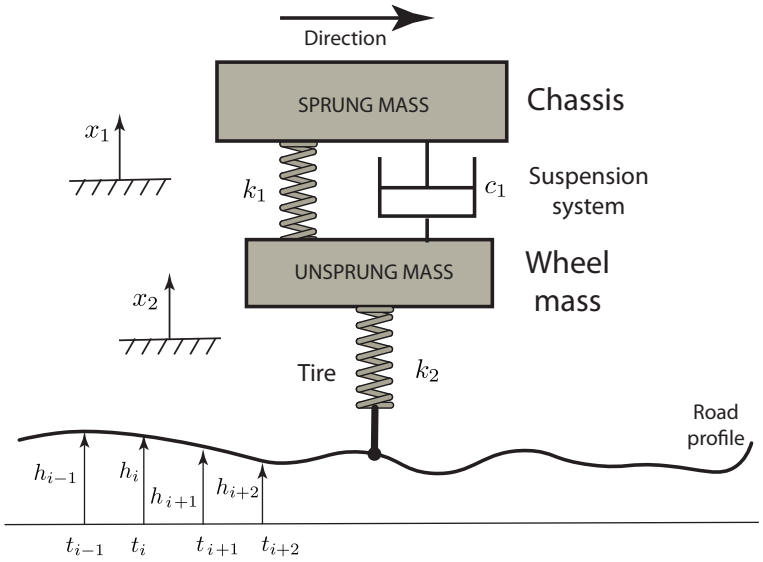


Figure 5.25: The quarter-car model

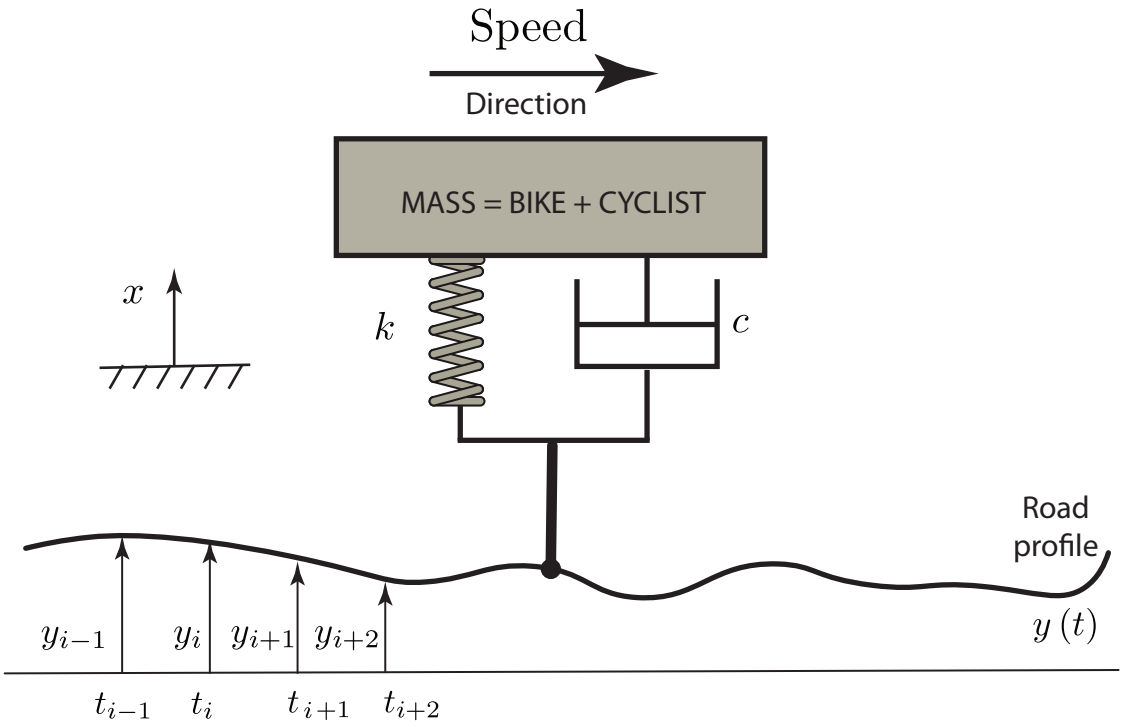


Figure 5.26: Road Profile Computation: Proposed dynamic model of the bicycle and biker

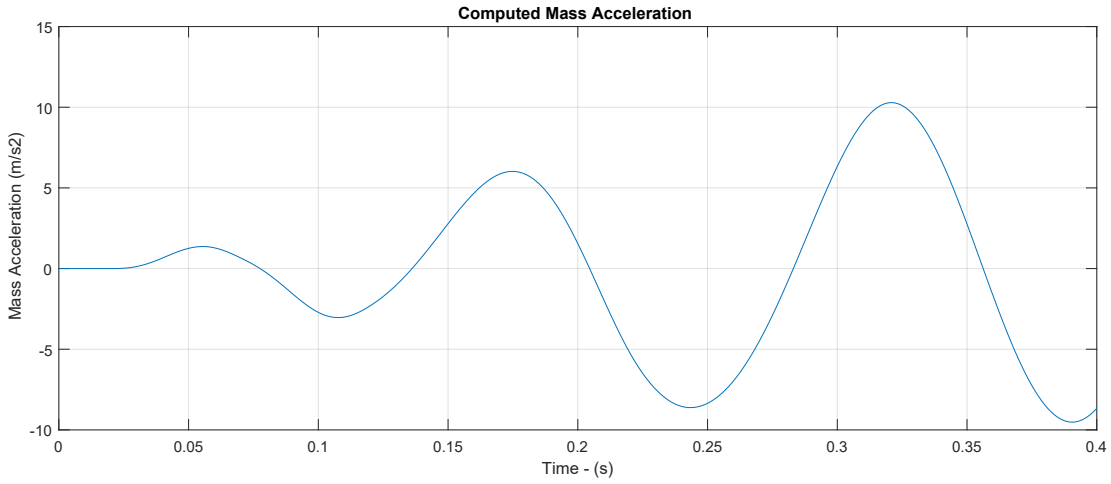


Figure 5.27: Simulated Mass Acceleration

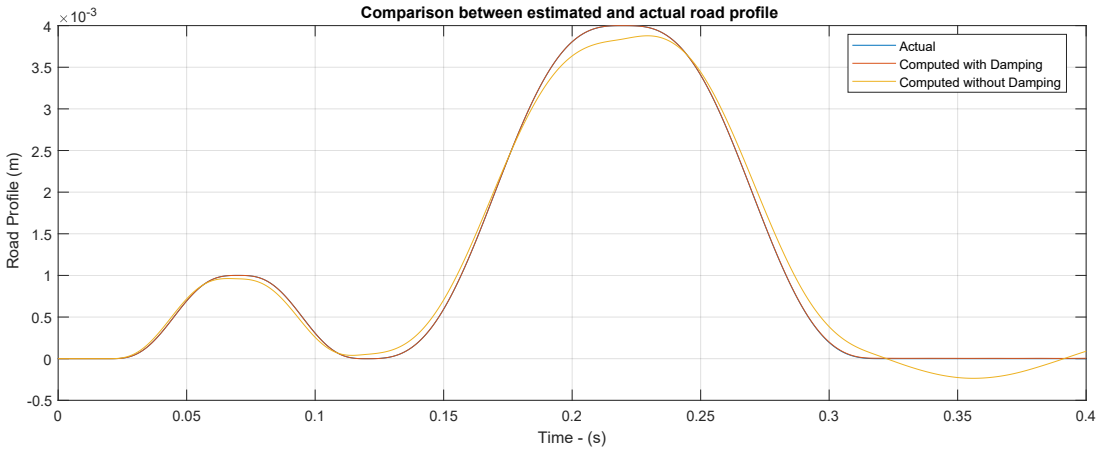


Figure 5.28: Road Profile Comparison

Feature	ASIC	DSP	uC	FPGA
SW Flexibility	-	✓✓	✓✓	✓✓
HW Reconfigurability	✗	✗	✗	✓✓
Dev. Time Cost	✗	✓✓	✓✓	✓✓
Production Costs	✓✓	✓✓	✓✓	✓
Performance	✓	✓	✓	✓✓
Power Efficiency	✓✓	✓	✓	✓

Legend: ✓✓ - Good ✓ - Moderate ✗ - Poor

Figure 5.29: Qualitative comparison between `ASIC`, `DSP`, Microcontroller and `FPGA`

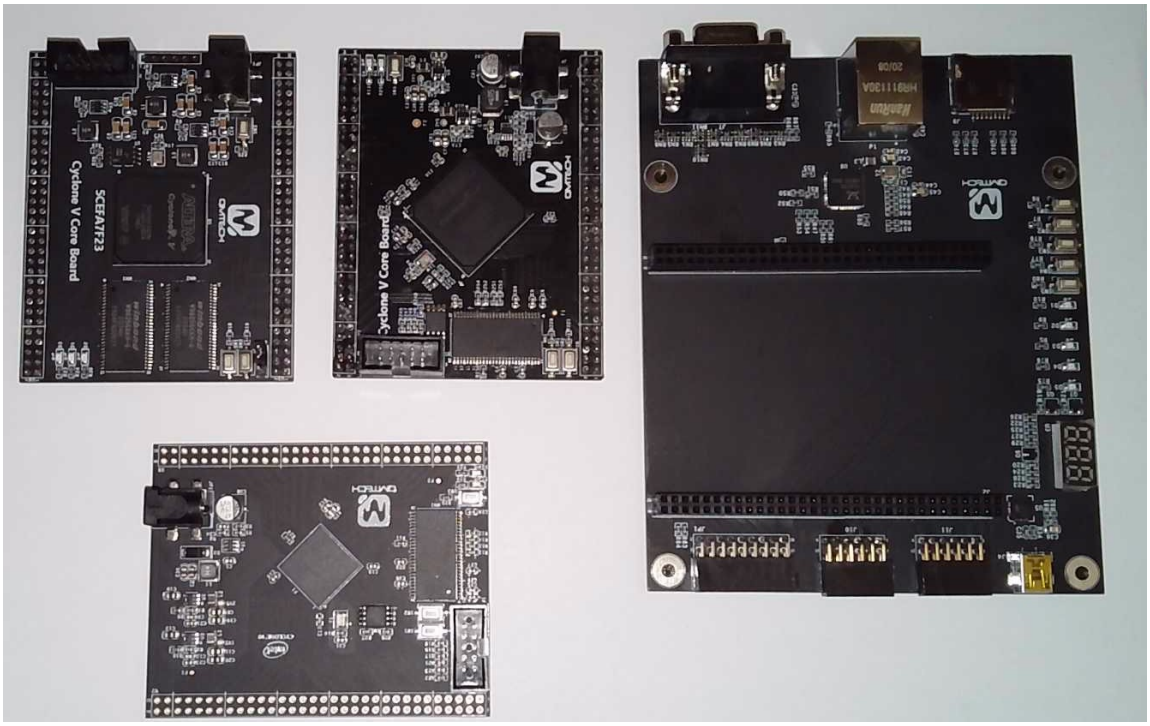


Figure 5.30: FPGA Development Board Ecosystem

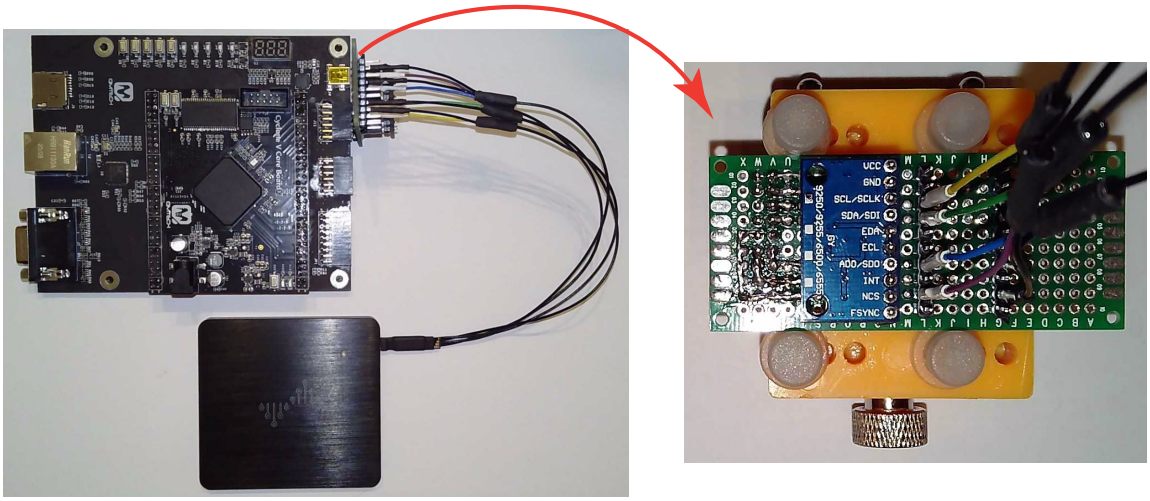


Figure 5.31: Debug with logic analyzer

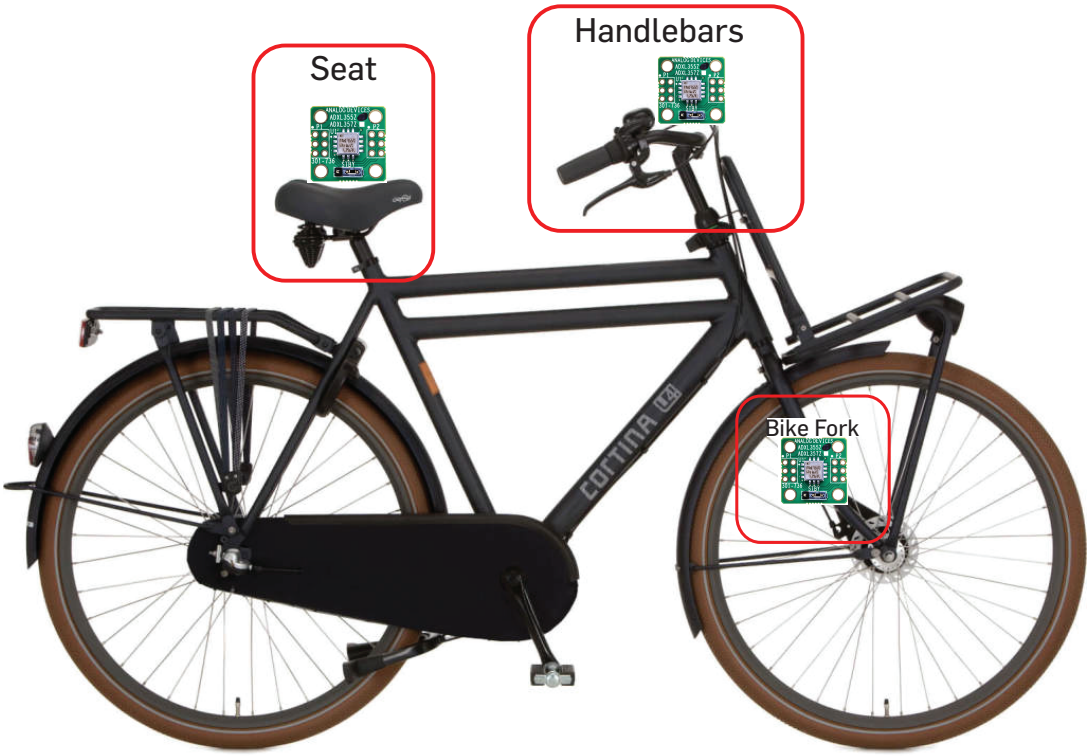


Figure 5.32: Installation position of the IMU sensors

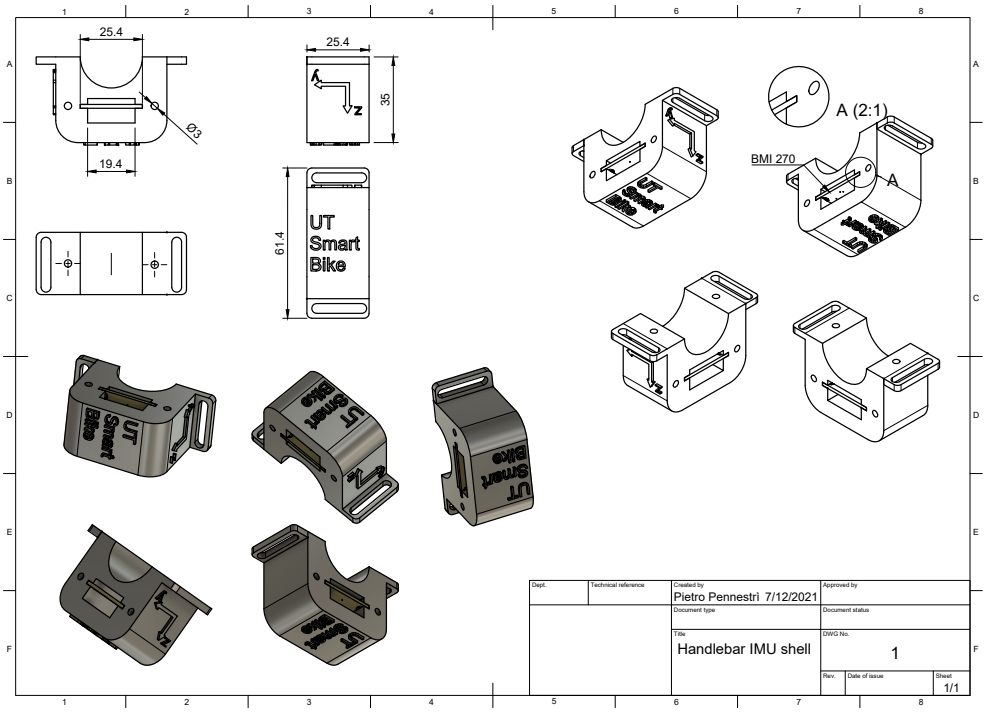


Figure 5.33: `IMU` Shell

Conclusions and Future Work

This work tackled the problem of estimating the spatial orientation of a single 9-DoF IMU platform. The gyroscope, accelerometer and magnetometer data have been fused using a novel approach derived from the classic Madgwick algorithm. In particular, the following ameliorations have been proposed:

- The Madgwick algorithm has been translated from quaternion into matrix notation.
- The algebraic structure to evaluate the gradient has been simplified into linear matrix operations.
- A kinematic consistency condition between quaternion components and their time derivatives has been added to the computational flow.

As a result of the modifications listed above, the modified Madwick algorithm has greater computational efficiency and tunable accuracy.

Due to their great flexibility and low power consumption, FPGAs are attractive to run sensor fusion applications at the edge. Thus, a custom architecture to execute the proposed sensor fusion algorithm was implemented. Since the normalization of vectors is a common executed task in sensor fusion, its optimization justified the development of a novel architecture to compute, with shared resources, both $\sqrt{x}$ and $\frac{1}{\sqrt{x}}$. In particular, for these algebraic operations, novel polynomial approximations, based on the Chebyshev min-max criterion, have been deduced. The improvements with respect to recently published algorithms have been highlighted using numerical examples.
The description of a novel `IMU` based onboard device for bikers vibration comfort analysis and road condition assessment according to international standards concludes the thesis.

As future work, such a device needs to be prototyped and tested under real conditions. Design iterations will be certainly required. Although omitted in this thesis, data communication, storage and use for different socially meaningful purposes has to be developed. Moreover, since the device enhances bikers safety and, more generally, sustainable mobility, the author desires to make it commercially available at low cost to the thousands of workers that everyday use bikes.

Bibliography

[1] P.L. Chebyshev. “Sur les parallélogrammes les plus simples symmétriques autour d’un axe”. In: *Oeuvres de P.L. Tchebichev*. Vol. II. Published in 1878. Académie Impériale del Sciences, 1907, pp. 709–715.

[2] F. Klein. *Elementary mathematics from an advanced standpoint: Higher complex numbers, especially quaternions*. Dover Publications Inc., 1960.

[3] E.I. Remez. *General computational methods of Chebyshev approximation: The problems with linear real parameters*. Vol. I. Translation series. U.S. Atomic Energy Commission, Division of Technical Information, 1962. (Visited on 10/30/2021).

[4] J.R. Rice. *The approximation of functions*. Addison-Wesley Publishing Company, 1964.

[5] G.H. Golub and L.B. Smith. *Chebyshev Approximation of Continuous Functions by Chebyshev System of Functions*. Technical Report. 1967. (Visited on 10/30/2021).

[6] G.H. Golub. “Chebyshev Approximation of Continuous Functions by a Chebyshev System of Functions [E2]”. In: *Communications of the ACM* 14.11 (1971), pp. 737–746. (Visited on 10/30/2021).

[7] D.M. Henderson. *Euler Angles, Quaternions, and Transformation Matrices*. 1977.

[8] J Rooney. “A survey of representations of spatial rotation about a fixed point”. In: *Environment and Planning B: Planning and Design* 4.2 (1977), pp. 185–210. ISSN: 0265-8135.

[9] T.D. Gillespie, M.W. Sayers, and L. Segel. *Calibration of response-type road roughness measuring systems*. Tech. rep. Washington, D.C., 1980, p. 88. (Visited on 08/18/2021).

[10] P. Nikravesh. “Spatial kinematic and dynamic analysis with Euler parameters”. In: *Computer Aided Analysis and Optimization of Mechanical System Dynamics*. Ed. by E.J. Haug. NATO ASI Series. Springer Verlag, 1984, pp. 261–281.

[11] “IEEE Standard for Binary Floating-Point Arithmetic”. In: *ANSI/IEEE Std 754-1985* (1985), pp. 1–20. DOI: 10.1109/IEEESTD.1985.82928.

[12] P.E. Nikravesh, R.A. Wehage, and O.K. Kwon. “Euler Parameters in Computational Kinematics and Dynamics. Part 1”. In: *ASME Journal of Mechanisms, Transmissions, and Automation in Design* 107 (Sept. 1985), pp. 358–365.

[13] M.R. Wigan and P.T. Cairney. *Road and Track Roughness Factors for Bicycle Usage*. Transportation Research Record. 1985.

[14] M.D. Shuster. “A Survey of Attitude Representations”. In: *Journal of the Astronautical Sciences* 41.4 (1993), pp. 439–517. (Visited on 08/05/2021).

[15] M. Saywers. *On the Calculation of International Roughness Index from Longitudinal Road Profile*. Tech. rep. 1995.

[16] M. Griffin. *Handbook of Human Vibration*. Academic Press, 1996.

[17] M.W. Sayers and S.M. Karamihas. *Interpretation of Road Roughness Profile Data*. Tech. rep. The University of Michigan Transportation Reesearch Institute, 1996.

[18] *ISO 2631 - 1 Mechanical Vibration and Shock - Evaluation of Human Exposure to Whole-Body Vibration*. ISO. 1997.

[19] *ISO 4287 - Geometrical Product Specifications (GPS) - Surface Texture: Profile Method - Terms, definitions and surface texture parameters*. Tech. rep. 1997.

[20] K. Ahlin and N.O.J. Granlund. "Relating road roughness and vehicle speeds to human whole body vibration and exposure limits". In: *International Journal of Pavement Engineering* 3.4 (2002), pp. 207–216. ISSN: 1029-8436. DOI: 10.1080/10298430210001701. (Visited on 07/27/2021).

[21] B. Kuipers. *Quaternions and rotation sequences*. Princeton Univ Press, 2002.

[22] A.L. Schwab. *Quaternions, Finite Rotation and Euler Parameters*. Tech. rep. Delft University of Technology, 2002.

[23] R. Mahony, T. Hamel, and J.-M. Pflimlin. "Complementary filter design on the special orthogonal group SO(3)". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. 2005, pp. 1477–1484. DOI: 10.1109/CDC.2005.1582367.

[24] A.L. Schwab and J.P. Meijaard. "How to draw Euler angles and utilize Euler parameters". In: *Proceedings of IDETC/CIE 2006*. DETC2006-99307. ASME. Sept. 2006, p. 7.

[25] *ISO 5349 - 1 Mechanical Vibration - Measurement and Evaluation of Human Exposure to Hand - Transmitted Vibration*. iso. 2007.

[26] R. Mayans. "The Chebyshev Equioscillation Theorem". In: *Convergence* (2007). URL: https://www.maa.org/press/periodicals/loci/joma/the-chebyshev-equioscillation-theorem (visited on 10/30/2021).

[27] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. "Nonlinear complementary filters on the special orthogonal group". In: *IEEE transactions on automatic control* 53.5 (2008), pp. 1203–1218. ISSN: 0018-9286. DOI: 10.1109/{TAC}.2008.923738. URL: http://ieeexplore.ieee.org/document/4608934/ (visited on 09/26/2021).

[28] C. C. O. Reynolds et al. "The impact of transportation infrastructure on bicycling injuries and crashes: A review of the literature". In: *Environmental Health: A Global Access Science Source* 8 (2009), p. 47.

[29] B. Clarsen, T. Krosshaug, and R. Bahr. "Overuse injuries in professional road cyclists". In: *The American Journal of Sports Medicine* 38.12 (2010), 2494–2501.

[30] S. O. H. Madgwick. *An Efficient Orientation Filter for Inertial and Inertial/Magnetic Sensor Arrays*. Tech. rep. Bristol, UK: Report x-io and University of Bristol, Apr. 2010.

[31] S.J. Shultz and S.J. Gordon. "Recreational cyclists: The relationship between low back pain and training characteristics". In: *International Journal of Exercise Science* 3.3 (2010), pp. 79–85.

[32] Sebastian O. H. Madgwick, Andrew J. L. Harrison, and Ravi Vaidyanathan. "Estimation of IMU and MARG orientation using a gradient descent algorithm". In: *2011 IEEE International Conference on Rehabilitation Robotics*. 2011, pp. 1–7. DOI: 10.1109/ICORR.2011.5975346.

[33] Mark Olieman, Raluca Marin-Perianu, and Mihai Marin-Perianu. “Measurement of dynamic comfort in cycling using wireless acceleration sensors”. In: *Procedia Engineering* 34 (2012), pp. 568–573. ISSN: 18777058. (Visited on 06/20/2021).

[34] Joachim Vanwalleghem et al. “Design of an instrumented bicycle for the evaluation of bicycle dynamics and its relation with the cyclist’s comfort”. In: *Procedia Engineering* 34 (2012), pp. 485–490. ISSN: 18777058. (Visited on 06/20/2021).

[35] X. Chiementin et al. “Hand–arm vibration in cycling”. In: *Journal of Vibration and Control* 19.16 (2013), pp. 2551–2560. ISSN: 1077-5463.

[36] J. Lépine, Y. Champoux, and J.-M. Drouet. “Road bike comfort: on the measurement of vibrations induced to cyclist”. In: *Sports Engineering* 17.2 (2014), pp. 113–122. ISSN: 1369-7072. (Visited on 06/20/2021).

[37] N. Petrone and F. Trabacchin. “Development of a test bench for the subjective and objective evaluation of the vibrational comfort of bicycle components”. In: *Volume 3: 16th International Conference on Advanced Vehicle Technologies; 11th International Conference on Design Education; 7th Frontiers in Biomedical Devices*. ASME, 2014, V003T01A033. ISBN: 978-0-7918-4634-6. (Visited on 06/20/2021).

[38] Jean-Marc Drouet, Catherine Guastavino, and Nicolas Girard. “Perceptual Thresholds for Shock-type Excitation of the Front Wheel of a Road Bicycle at the Cyclist’s Hands”. In: *Procedia Engineering* 147 (2016), pp. 724–729. ISSN: 18777058. DOI: 10.1016/j.proeng.2016.06.264. (Visited on 06/20/2021).

[39] P. Havinga et al. “Roads: A road pavement monitoring system for anomaly detection using smart phones”. In: *Big data analytics in the social and ubiquitous context*. Ed. by Martin Atzmueller et al. Vol. 9546. Lecture notes in computer science. Cham: Springer International Publishing, 2016, pp. 128–146. ISBN: 978-3-319-29008-9. (Visited on 08/18/2021).

[40] *ISO8608: Mechanical vibration —Road surface profiles —Reporting of measured data*. Tech. rep. 2016.

[41] Marius Cornea. “Ulps and relative error”. In: *2017 IEEE 24th Symposium on Computer Arithmetic (ARITH)*. IEEE, 2017, pp. 90–97. ISBN: 978-1-5386-1965-0. DOI: 10.1109/{ARITH}.2017.30.

[42] Abul Hasnat et al. “A fast FPGA based architecture for computation of square root and Inverse Square Root”. In: *2017 Devices for Integrated Circuit (DevIC)*. IEEE, 2017, pp. 383–387. ISBN: 978-1-5090-4724-6. DOI: 10.1109/{DEVIC}.2017.8073975. URL: http://ieeexplore.ieee.org/document/8073975/ (visited on 11/07/2021).

[43] X. Qian and D. Niemeier. “Predicting Bicycle Ride Quality from Dynamic Measurements and Pavement Roughness Index”. In: *6th Annual International Cycling Safety Conference*. Davis, California, 2017.

[44] M.D. Taylor. “Cyclist exposure to hand-arm vibration and pavement surface improvement”. In: University of Strathclyde, 2017. (Visited on 06/20/2021).

[45] M.D. Taylor, A. Edgar, and M. Rayne. “Cycling pavement assessment using hand-arm vibration exposure”. In: *Proceedings of the Institution of Civil Engineers - Infrastructure Asset Management* (2017). (Visited on 06/20/2021).

[46] Amrit Dhillon. *More deadly than terrorism: potholes responsible for killing 10 people a day in India*. The Guardian. 2018. URL: https://www.theguardian.com/world/2018/jul/25/more-deadly-than-terrorism-potholes-responsible-for-killing-10-people-a-day-in-india.

[47] Kaiyue Zang et al. “Assessing and Mapping of Road Surface Roughness based on GPS and Accelerometer Sensors on Bicycle-Mounted Smartphones.” In: *Sensors (Basel, Switzerland)* 18.3 (2018).

[48] Helen McArdle. *Cyclists: Why uneven roads and potholes are endangering your health*. The Herald. 2019. URL: https://www.heraldscotland.com/news/17858097.cyclists-uneven-roads-potholes-endangering-health/.

[49] A. Doria et al. “An experimental-numerical method for the prediction of on-road comfort of city bicycles”. In: *Vehicle System Dynamics* (2020), pp. 1–21. ISSN: 0042-3114. (Visited on 06/20/2021).

[50] P.I. Edwards and T.P. Holsgrove. “Thunder road - whole-body vibration during road cycling, and the effect of different seat post designs to minimise it”. In: *Journal of Sports Sciences* (2020). DOI: 10.1080/02640414.2020.1829361.

[51] M. Gogola. “Analyzing the Vibration of Bicycles on Various Road Surfaces in the City of Zilina”. In: *The Archives of Automotive Engineering* 88.2 (2020), pp. 77–97. (Visited on 06/20/2021).

[52] M. Hardinghaus and P. Papantoniou. “Evaluating Cyclists’Route Preferences with Respect to Infrastructure”. In: *Sustainability* 12.8 (2020), p. 3375. ISSN: 2071-1050. (Visited on 08/21/2021).

[53] Kyuhyun Lee and Ipek N. Sener. “Emerging data for pedestrian and bicycle monitoring: Sources and applications”. In: *Transportation Research Interdisciplinary Perspectives* (2020), p. 100095. ISSN: 25901982.

[54] Amir Shtayat et al. “Using a smartphone software and a regular bicycle to monitor pavement health statues”. In: *2020 2nd International Conference on Robotics Systems and Vehicle Technology*. New York, NY, USA: ACM, 2020, pp. 121–126. ISBN: 9781450389068. DOI: 10.1145/3450292.3450312.

[55] T. Holsogrove. *Rough ride: whole-body vibrations might be worse for you than you think*. Feb. 2021. URL: https://cyclingtips.com/2021/02/rough-ride-whole-body-vibrations-might-be-worse-for-you-than-you-think/.

[56] *International Roughness Index*. Wikipedia. 2021. URL: https://en.wikipedia.org/wiki/International_roughness_index.

[57] Leonid V. Moroz, Volodymyr V. Samotyy, and Oleh Y. Horyachyy. “Modified fast inverse square root and square root approximation algorithms: the method of switching magic constants”. In: *Computation* 9.2 (2021), p. 21. ISSN: 2079-3197. DOI: 10.3390/computation9020021. URL: https://www.mdpi.com/2079-3197/9/2/21 (visited on 11/01/2021).

[58] Luis Parrilla et al. “Table-free Seed Generation for Hardware Newton–Raphson Square Root and Inverse Square Root Implementations in IoT Devices”. In: *IEEE Internet of Things Journal* (2021), pp. 1–1. ISSN: 2327-4662. DOI: 10.1109/{JIOT}.2021.3116536. URL: https://ieeexplore.ieee.org/document/9552945/ (visited on 11/03/2021).

[59] M. Toljic, T. Brezina, and G. Emberger. “Influence of surface roughness on cyclists t velocity choices”. In: *Proceedings of the Institution of Civil Engineers - Municipal Engineer* 174.1 (2021), pp. 2–13. ISSN: 0965-0903. DOI: 10.1680/jmuen.18.00058. (Visited on 08/18/2021).

[60] *Directive 2002/44/EC of the European Parliament and of the Council of 25 June 2002 on the minimum health and safety requirements regarding the exposure of workers to the risks arising from physical agents (vibration) (sixteenth individual Directive within the meaning of Article 16(1) of Directive 89/391/EEC) - Joint Statement by the European Parliament and the Council*. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32002L0044.

[61] *The phenomenon of vibration in cycling: the Ortholite solution.* URL: http://engineerstalk.mavic.com/en/the-phenomenon-of-vibration-in-cycling-the-ortholite-solution/.