Geodätisch-geophysikalische Arbeiten in der Schweiz

(Fortsetzung der Publikationsreihe
«Astronomisch-geodätische Arbeiten in der Schweiz»)

herausgegeben von der

Schweizerischen Geodätischen Kommission
(Organ der Akademie der Naturwissenschaften Schweiz)

Neunundachtzigster Band
Volume 89

Modeling and Processing Approaches for Integrated Inertial Navigation

Yannick Stebler

2013

Adresse der Schweizerischen Geodätischen Kommission:

Institut für Geodäsie und Photogrammetrie
Eidg. Technische Hochschule Zürich
ETH Zürich
8093 Zürich
Switzerland

Internet: http://www.sgc.ethz.ch

ISBN 978-3-908440-34-5

Redaktion des 89. Bandes:
Dr.Y. Stebler, J. Müller-Gantenbein, Prof. A. Geiger
Druck: Print-Atelier ADAG, Zürich

Vorwort

Obwohl die Kombination eines globalen Navigationssatellitensystems (GNSS) mit einem inertialen Navigationssystem kein neues Konzept ist, durchlebt diese Technik derzeit dank der Einführung von Mikrosystemen (MEMS) eine kleine Revolution. In der Tat ist die Anzahl möglicher Anwendungen beruhend auf der Inertialtechnologie umgekehrt proportional zur Grösse, zum Gewicht und zu den Kosten der Sensoren. Mit anderen Worten sind INS/GNSS Systeme dank der MEMS Technologie nicht zwingend besser als ihre Vorgänger, jedoch sind sie erheblich kleiner, leichter und billiger. Dies erklärt ihre vielfältige Anwendung, die weit über ihr ursprüngliches Gebiet der Navigation hinaus reicht.

Trotz ihrer Beliebtheit stellt die Einführung von auf MEMS basierenden Inertialsensoren eine beträchtliche Herausforderung hinsichtlich Modellierung und Schätzung ihres Fehlerverhaltens dar. Die Dissertation von Yannick Stebler liefert eine sauber formulierte und originelle Antwort auf die Problematik besserer Ansätze in Modellierung und Schätzung. Die Weiterentwicklung etablierter Werkzeuge wie das der spektralen Leistungsdichte (PSD), der Allan Varianz oder des EM-Algorithmus führte zur überraschenden Entdeckung eines neuen leistungsstarken Schätzers, der in einer Tandem-Arbeit zwischen dem Autor und seinem ehemaligen Fachkollegen Stéphane Guerrier (zur Zeit Doktorand an der Universität Genf) in die Praxis umgesetzt wurde. Die Methode wurde auf den Namen "Generalized Method of Wavelet Moments"(GMWM) getauft. Die GMWM Resultate zur Schätzung zusammengesetzter stochastischer Prozesse übertrumpfen die Resultate, die auf dem IEEE Standard der Allan Varianz basieren. Die Anpassung der Methode hat daher eine potentiell grosse Auswirkung auf verschiedenste Disziplinen ausgehend von den Inertialsensoren, über die Biologie und Wirtschaft bis hin zu den Oszillatoren (zum Beispiel Atom- oder Quartzuhren).

Hinter einer bedeutenden Entdeckung steht oft eine sehr einfache Idee. Dies trifft auch auf den GMWM Schätzer zu, wenn die Gleichungen im Definitionsbereich der Wavelets ausgedrückt werden. Seine Umsetzung ist jedoch sehr aufwendig und ist das Verdienst des Autors. Konkret hat er die Leistung der Fehlermodellierung von gewissen Sensoren (oft klein und billig, aber ungenau) an neue Grenzen gebracht. Jedenfalls geht die Tragweite dieser exzellenten Arbeit weit über die praktische Anwendung der integrierten Navigation, wie sie hier präsentiert wird, hinaus.

Dr. Jan Skaloud, MER
Labor für Topometrie
ETH Lausanne

Prof. Dr. Alain Geiger
ETH Zürich
Präsident der SGK

Préface

Bien que la combinaison de systèmes de positionnement par satellites (GNSS) et de systèmes de navigation inertielle (INS) ne soit pas nouvelle, cette technique subit actuellement une petite révolution grâce à l'arrivée de systèmes micro-électro-mécaniques (MEMS). En effet, le nombre d'applications possibles basées sur la technologie inertielle est inversement proportionnel à la taille, au poids et au coût de ses capteurs. En d'autres termes, grâce à la technologie MEMS, les systèmes INS/GNSS ne sont pas nécessairement plus performants que leurs prédécesseurs, mais sont devenus bien plus petits, plus légers et meilleur marché. Cela a eu pour conséquence d'étendre l'usage de tels systèmes bien au-delà de leur domaine d'origine qu'est la navigation.

Malgré leur popularité, l'arrivée des unités de navigation inertielle (IMU) basées sur les récepteurs MEMS représente un défi considérable en matière de modélisation et d'estimation de leurs erreurs. La thèse de Yannick Stebler apporte une réponse originale et bien formulée à la problématique de trouver de meilleures approches de modélisation et d'estimation. Le développement des outils existants tels que la densité spectrale de puissance (PSD), la variance d'Allan ou l'algorithme espérance-maximisation (EM) a conduit à la surprenante découverte d'un nouvel estimateur, baptisé méthode généralisée des moments d'ondelettes (GMWM). Sa mise en oeuvre résulte d'une étroite collaboration entre l'auteur et son ancien collègue Stéphane Guerrier (actuellement doctorant à l'Université de Genève). Pour l'estimation de modèles basés sur une somme de processus stochastiques, le GMWM fournit de meilleurs résultats que le procédé basé sur la variance d'Allan qui correspond actuellement au standard IEEE. Ainsi, cette nouvelle technique pourrait avoir un fort impact dans de nombreuses disciplines, allant des récepteurs inertiels à la biologie ou à l'économie, en passant par les oscillateurs (tels que les horloges, atomiques ou quartz).

Derrière une découverte importante, il y a souvent une idée très simple et c'est le cas de l'estimateur GMWM lorsque les équations sont exprimées dans le domaine des ondelettes. En revanche, son implémentation est très complexe et sa réussite revient à l'auteur. Concrètement, il a poussé la performance de la modélisation des erreurs de certains capteurs (souvent petits et bon marché, mais imprécis) vers de nouvelles limites. Toutefois, la portée de cet excellent travail s'étend bien au-delà son application à la navigation intégrée présentée ici.

Dr. Jan Skaloud, MER
Laboratoire de Topométrie
EPF Lausanne

Prof. Dr. Alain Geiger
ETH Zürich
Président de la CGS

Foreword

Although the approach of integrating Global Navigation Satellite Systems (GNSS) with Inertial Navigation Systems (INS) is not new, it currently undergoes a small revolution with the arrival of Micro-Electro-Mechanical Systems (MEMS). Indeed, the number of possible applications relying on inertial technology is inversely proportional to the size, weight and cost of its sensors. In other words, thanks to MEMS the INS/GNSS systems are not necessarily more accurate in comparison to its predecessors, but considerably smaller, lighter and cheaper. This explains its widespread use, well beyond its original field of navigation.

Despite its popularity, the introduction of Inertial Measurement Units (IMU) based on MEMS represents a considerable challenge in terms of modeling and estimating their behavior. The dissertation of Yannick Stebler is a well-formulated and original response to such a dual quest of finding better approaches in modeling and estimation. The process of extending well established tools such as Power Spectral Density (PSD), Allan variance or Expectation-Maximization (EM) lead to the surprising discovery of a new powerful estimator that was put in practice through a tandem work between the author and its former colleague Stéphane Guerrier (currently a doctoral student at the University of Geneva). The method was baptized as a Generalized Method of Wavelet Moments (GMWM) and the results for estimating compound stochastic processes obtained by the GMWM estimator are superior to the results of the well-accepted IEEE standard of Allan variance. For this reason, the adaptation of the new method promises a large potential impact across various disciplines ranging from inertial sensors, over biology and economy up to all sorts of oscillators (like e.g. atomic or other clocks).

The main idea behind a noble discovery is often simple and this applies also to the GMWM estimator when the equations are expressed in the wavelet domain. Its implementation is, however, very involved and it goes to the credit of the author. It is an excellent work with an influence reaching far beyond the discussed practical content of integrated navigation, where it pushed the performance of estimation with the given sensors (often small, inexpensive but imprecise) to new limits.

Dr. Jan Skaloud, MER
Geodetic Engineering Laboratory
EPF Lausanne

Prof. Dr. Alain Geiger
ETH Zürich
President of SGC

Abstract

The challenge of estimating the position, velocity and orientation in space in a precise and reliable way, at any time, with and without reception of satellite signals, is the core subject of this dissertation. To this end, the use of Bayesian filters which fuse outputs from autonomous inertial navigation with satellite positioning is a well-accepted and largely proven approach. The quality of integrated systems is mainly driven by the errors affecting the inertial sensors. This research intends to improve the navigation accuracy of INS/GNSS by proposing and investigating novel approaches at two levels. First, a new estimation framework is developed that allows to model complex composite stochastic processes. We consolidate the proposed estimator on a theoretical basis and validate it through simulations and experiments. Results show the ability of our method to estimate models for which other conventional approaches (e.g. Allan variance and likelihood-based estimators) fail, thereby supporting the challenging stage of navigation filter design. Second, we investigate filter designs accounting for inertial sensor redundancy at observation and state levels. The benefits brought by such filters in terms of navigation accuracy and adaptive modeling of sensor noise are discussed in the context of experiments. For that purpose, a redundant MEMS-based inertial navigation system was designed and operated on a vehicle. Compared to classical single-IMU based filters, we found a significant bounding of the position, velocity and attitude error when operating redundant inertial systems. Contrary to single-IMU/GNSS systems, the redundant configuration is able to self-evaluate the level of system noise and thus to catch the effects of the dynamics. The improved performance and robustness is attractive for many applications requiring reliable and accurate trajectory determination.

Keywords: navigation, inertial navigation system, stochastic process, Kalman filter, signal processing, error modeling, sensor redundancy, estimation methods, INS/GNSS integration, MEMS, IMU.

Contents

List of Figures

List of Tables

Conventions and Notation

Important Conventions

$F(\cdot)$	refers to a function
$F(x)$	refers to the value of the function $F(\cdot)$ at x
$\{x_k\}$	refers to a sequence of values indexed by the integer k
x_k	refers to the kth value of a sequence
$\hat{x}$	estimated quantity of x
$\bar{x}$	averaged quantity
$\dot{x}$	refers to the time derivative of x
$\ddot{x}$	refers to the second time derivative of x
$\mathbf{x}$	$(1 \times N)$ vector containing $\{x_k\}$ such that $\mathbf{x} = [x_k]_{k=1,\ldots,N}$
$\mathbf{x}^T$	refers to the transpose vector of $\mathbf{x}$
$\mathbf{X}$	refers to a matrix

Abbreviations used Frequently

ACF	autocorrelation function
ACS	autocorrelation sequence
AIC	Akaike information criterion
ALS	airborne laser scanning
AR	auto-regressive
ARMA	auto-regressive moving-average
BI	bias instability
BIC	Bayesian information criterion
CP	carrier phase
DCM	direction cosine matrix
DGPS	differential global positioning system
DIMU	distributed inertial measurement unit
DWT	discrete wavelet transform
ECEF	Earth-centered Earth-fixed
EKF	extended Kalman filter
EM	expectation-maximization

ENU	east-north-up
FDI	fault detection and isolation
GARCH	generalized auto-regressive conditional heteroskedasticity
GLS	generalized least-squares
GM	Gauss-Markov
GMM	generalized method of moments
GMWM	generalized method of wavelet moments
GNSS	global navigation satellite system
GPS	global positioning system
GST	Greenwich sideral time
IEEE	institute of electrical and electronics engineers
IMU	inertial measurement unit
INS	inertial navigation system
ITRF	international terrestrial reference frame
MA	moving-average
MEMS	micro-electro-mechanical system
MLE	maximum likelihood estimator
MODWT	maximal overlap discrete wavelet transform
MRS	Markov regime-switching
MSC	Markov switching criterion
NED	north-east-down
OS	operating system
PC	personal computer
PPS	pulse per second
PSD	power spectral density
QN	quantization noise
QUEST	quaternion estimation
RIMU	redundant inertial measurement unit
R-IMU	reference inertial measurement unit
RMSE	root mean squared error
R-RMSE	relative root mean squared error
RR	random ramp
RTS	Rauch-Tung-Striebel
RW	random walk
S-IMU	studied inertial measurement unit
TAR	threshold auto-regressive
TTL	transistor-transistor logic
UAV	unmanned air vehicle
WGS	world geodetic system
WMM	world magnetic model
WN	white noise

Non-Greek Notation used Frequently

a	semi-major axis of an ellipsoid
a_j	jth coefficient of an AR$(\cdot)$ or ARMA$(\cdot,\cdot)$ process
$\mathbf{a}^q$	leverarm between b-frame and GNSS antenna center expressed in q-frame
$\boldsymbol{a}_k^q$	vector containing specific force observations from all accelerometers composing the DIMU at digital time t_k and expressed in frame q
b	semi-minor axis of an ellipsoid
b_j	jth coefficient of an MA$(\cdot)$ or ARMA$(\cdot,\cdot)$ process
$\mathbf{b}_\bullet$	vector containing accelerometer or gyroscope measurement bias
c_j	jth coefficient of a GARCH$(\cdot,\cdot)$ process
$\mathbf{C}_p^q$	direction cosine matrix relating frame p to frame q
$\mathbf{C}_j(\alpha)$	elementary rotation matrix for rotation about axis j by an angle α
d_j	jth coefficient of a GARCH$(\cdot,\cdot)$ process
$(d_{i,j})_s$	order of AR$(\cdot)$ process at regime s for sensor i and axis j (in Markov regime switching model)
$\mathbb{E}[\cdot]$	expectation operator
e	first numerical eccentricity of an ellipsoid
e'	second numerical eccentricity of an ellipsoid
$F(\boldsymbol{\theta})$	model parameterized by $\boldsymbol{\theta}$
$\mathbf{F}(\cdot)$	$(p \times p)$ time-varying continuous coefficient matrix
f	frequency of a sinusoid
f_k	k/N or $k/N\Delta t$, the kth Fourier frequency
f_N	Nyquist frequency
$f(\cdot,\cdot)$	probability density function
$\mathbf{f}^q$	(3×1) specific force vector expressed in frame q
$\mathbf{g}^q$	vector containing gravitational acceleration expressed in frame q
$\breve{\mathbf{g}}^q$	local gravity vector expressed in frame q
$\boldsymbol{g}_k^q$	vector containing angular rate observations from all gyroscopes composing the DIMU at digital time t_k and expressed in frame q
$\mathbf{G}(\cdot)$	$(p \times r)$ time-varying continuous process noise coupling matrix
$H(\cdot)$	transfer function for $\{h_l\}$
$H^*(\cdot)$	complex conjugate transpose of transfer function $H(\cdot)$
$\widetilde{H}(\cdot)$	transfer function for $\{\widetilde{h}_l\}$
$H_j(\cdot)$	transfer function for $\{h_{j,l}\}$, with $H_1(\cdot) = H(\cdot)$
$\widetilde{H}_j(\cdot)$	transfer function for $\{\widetilde{h}_{j,l}\}$, with $\widetilde{H}_1(\cdot) = \widetilde{H}(\cdot)$
$\mathbf{H}(\cdot)$	$(l \times p)$ time-varying continuous measurement design matrix
$\mathbf{H}_k$	$(l \times p)$ discrete measurement design matrix for digital time t_k
h	ellipsoidal height
$\{h_l\}$	DWT wavelet filter
$\{\widetilde{h}_l\}$	MODWT wavelet filter
$\{h_{j,l}\}$	jth level DWT wavelet filter
$\{\widetilde{h}_{j,l}\}$	jth level MODWT wavelet filter
$\mathbf{I}_{n\times n}$	$(n \times n)$ identity matrix
J	largest DWT level for sample size $N = 2^J$ or number of scales
j	level (index) of scale usually (also used as generic index or $\sqrt{-1}$)

$\mathbf{K}_k$	Kalman gain matrix for digital time t_k
k	unitless index (discrete)
L	width of wavelet filter (unit scale)
L_j	width of jth level equivalent wavelet filter
$L(\cdot)$	likelihood function
$\mathbf{L}(\cdot)$	$(p \times r)$ time-varying continuous input coupling matrix
$\ell(\cdot)$	log-likelihood function
l	number of measurements (in dynamic systems)
M_j	number of nonboundary jth level MODWT coefficients
$\mathbf{m}^q$	vector containing Earth magnetic field expressed in frame q
N	sample size
$\mathcal{N}(\cdot, \cdot)$	Gaussian distribution function
$\mathbf{N}_\bullet$	skew-symmetric matrix accounting for cross-coupling errors
$\mathbf{0}_{n\times n}$	$(n \times n)$ null matrix
$\mathbf{P}_k^-$	predicted state covariance matrix for digital time t_k
$\mathbf{P}_k^+$	corrected state covariance matrix for digital time t_k
$\mathbf{P}_k^N$	smoothed state covariance matrix for digital time t_k
p	pitch angle or number of state variables or number of parameters
$p_{i,j}$	order of an AR$(\cdot)$ or ARMA$(\cdot, \cdot)$ process for sensor i on axis j
$p_{l,h}^{i,j}$	probability for moving from regime l to regime h for sensor i on axis j
$\mathbf{Q}(\cdot)$	$(r \times r)$ time-varying continuous system noise matrix
$\mathbf{Q}_k$	$(r \times r)$ discrete system noise matrix for digital time t_k
$\mathbf{q}_p^q$	(4×1) quaternion vector expressing rotation between frame p and frame q
$q_{i,j}$	order of an MA$(\cdot)$ or ARMA$(\cdot, \cdot)$ process for sensor i on axis j
R	number of IMUs composing the DIMU system
R_M	meridian radius of curvature
R_P	prime vertical radius of curvature
$\mathbf{R}(\cdot)$	$(l \times l)$ time-varying continuous measurement noise matrix
$\mathbf{R}_k$	$(l \times l)$ discrete measurement noise matrix for digital time t_k
$\mathbb{R}$	the entire real axis
$\mathbb{R}^+$	the positive real axis
$\mathbb{R}^p$	space of real-valued p dimensional vectors
r	roll angle or number of system noise coefficients
$r_{i,j}$	order of a GARCH$(\cdot, \cdot)$ process for sensor i on axis j
$\mathbf{r}^q$	position vector expressed in frame q
$\mathbf{r}_{p\to q}^p$	leverarm vector between p-frame and q-frame origins expressed in frame p
$S_Y(\cdot)$	power spectral density function of Y
$S_{XY}(\cdot)$	cross power spectral density function between X and Y
$\mathbf{S}_\bullet$	matrix containing scale factor errors
$(S_{i,j})_k$	regime at digital time t_k for sensor i on axis j
$S_{i,j}^\star$	number of regimes for sensor i on axis j
$s_{X,\tau}$	autocovariance sequence for process X at lag τ
$s_{i,j}$	order of a GARCH$(\cdot, \cdot)$ process for sensor i on axis j
t	unitless time index such that $t \in \mathbb{R}$
$\mathcal{U}(\cdot, \cdot)$	uniform distribution
$\mathbf{u}(\cdot)$	$(r \times 1)$ continuous deterministic input vector

$\mathbf{u}_k$	$(r \times 1)$ discrete deterministic input vector for digital time t_k
$\mathbf{V}_\bullet$	covariance matrix (in GMWM context)
$\mathbf{v}^q$	velocity vector expressed in frame q
$\mathbf{v}(\cdot)$	$(l \times 1)$ time-varying continuous measurement noise vector
$\mathbf{v}_k$	$(l \times 1)$ discrete measurement noise vector for digital time t_k
$W_{j,k}$	kth element of the jth level MODWT coefficients
$\{\overline{W}_{j,k}\}$	jth level MODWT coefficient sequence for the stochastic process $\{Y_k\}$
$\mathbf{W}$	weighting matrix in synthetic IMU computer
$\mathbf{w}(\cdot)$	$(r \times 1)$ time-varying continuous random forcing function
$\mathbf{w}_k$	$(r \times 1)$ discrete random forcing function for digital time t_k
$X(\cdot)$	continuous random variable
X_k	discrete random variable with $k \in \mathbb{Z}$
$\{x_k\}$	realization of process X_k with $k = 1, \ldots, N$
$\mathbf{x}(\cdot)$	$(p \times 1)$ continuous state vector
$\mathbf{x}_k$	$(p \times 1)$ vector of discrete state variables at digital time t_k
$\mathbf{x}_k^-$	predicted state vector for digital time t_k
$\mathbf{x}_k^+$	corrected state vector for digital time t_k
$\mathbf{x}_k^N$	smoothed state vector for digital time t_k
$\mathbf{x}^q$	vector $\mathbf{x}$ represented with respect to frame q
Y_k	discrete stochastic process with $k \in \mathbb{Z}$
$(Y_{i,j})_k$	discrete random variable corresponding to the signal recorded at digital time t_k by sensor i on the axis j
y	yaw angle
$\{y_k\}$	observed time series or realization of process Y_k with $k = 1, \ldots, N$
$\{(y_{i,j})_k\}$	realization of process $(Y_{i,j})_k$
$\{y_k^\star\}$	simulated sequence
$\mathbb{Z}$	set of integers
$\mathbb{Z}^+$	set of positive integers
$\mathbf{z}(\cdot)$	$(l \times 1)$ time-varying continuous measurement vector (in Kalman filtering theory)
$\mathbf{z}_k$	$(l \times 1)$ discrete measurement vector for digital time t_k (in Kalman filtering theory)

Greek Notation used Frequently

β	inverse correlation time of a first-order Gauss-Markov process
β_j	jth coefficient of the Adams-Bashforth or Adams-Moulton methods
δ	perturbation operator
$\delta(\tau)$	Dirac delta function
δ_τ	Kronecker delta function
Δt	sampling period
ε_k	kth element of a sequence of residuals
$(\varepsilon_{i,j})_k$	error affecting $(y_{i,j})_k$ at digital time t_k
$\boldsymbol{\varepsilon}^l$	attitude errors expressed with respect to l-frame
$(\gamma_{i,j})_k$	weight for sensor i and axis j computed at digital time t_k
$\mathbf{\Gamma}_k$	$(p \times r)$ discrete process noise coupling matrix for digital time t_k
λ	latitude
$\mathbf{\Lambda}_k$	$(p \times r)$ discrete input coupling matrix for digital time t_k
μ_X	expected value of the random variable X
$\mu_{j,k}$	true signal that should be observed at digital time t_k on axis j (in synthetic IMU computer)
$\nu^2(\tau_j)$	wavelet variance at scale τ_j (time independent)
$\hat{\nu}^2(\tau_j)$	unbiased MODWT estimator of wavelet variance at scale τ_j
$\hat{\nu}^2_\star(\tau_j)$	unbiased MODWT estimator of wavelet variance at scale τ_j computed on a simulated sequence
$\boldsymbol{\nu}^2$	vector containing the sequence $\{\nu^2(\tau_j) :\ j = 1, \ldots, J\}$
$\boldsymbol{\nu}^2(\boldsymbol{\theta})$	vector containing the wavelet variances generated by a model $F(\boldsymbol{\theta})$
$\boldsymbol{\omega}^q_{pq}$	(3×1) angular rate of the q-frame with respect to the p-frame expressed in frame q
$\mathbf{\Omega}^q_{pq}$	skew symmetric matrix representation of $\boldsymbol{\omega}^q_{pq}$, i.e. $\mathbf{\Omega}^q_{pq} = [\boldsymbol{\omega}^q_{pq}\times]$
ω_s	frequency of Schuler oscillation
π	$3.141592653589793\cdots$
$\mathbf{\Pi}_\bullet$	orthogonal projector
$\boldsymbol{\phi}$	$(1 \times J)$ binding function vector computed on a signal
$\boldsymbol{\phi}(\boldsymbol{\theta})$	$(1 \times J)$ binding function vector issued from a model $F(\boldsymbol{\theta})$
$\mathbf{\Phi}_k$	$(p \times p)$ discrete transition matrix between digital times t_{k-1} and t_k
ϕ	longitude
$\boldsymbol{\psi}$	vector containing the misalignment angles between p-frame and c-frame
$\boldsymbol{\psi}^p_{p\to q}$	vector containing the misalignment angles between p-frame and q-frame expressed in p-frame
$\rho_X(\tau)$	autocorrelation function for continuous process X at lag τ
$\rho_{X,\tau}$	autocorrelation sequence for discrete process X at lag τ
$\rho_{XY,\tau}$	crosscorrelation sequence between discrete process X and Y at lag τ
σ^2_X	variance of a random variable X
$\hat{\sigma}^2_x$	estimated variance of an observed sequence x_k issued from process X
$\sigma^2_Y(\tau)$	theoretical Allan variance of the signal Y at averaging time τ
$\hat{\sigma}^2_{\bar{y}}(\tau)$	estimated Allan variance of the observed signal $\{y_k\}$ at averaging time τ
$\boldsymbol{\sigma}$	vector containing standard deviations
$\mathbf{\Sigma}$	variance-covariance matrix

τ	lag index in autocorrelation sequence or scale or averaging time
τ_j	2^{j-1}, unitless scale of jth level wavelet coefficients ($j \geq 1$)
$\boldsymbol{\theta}$	$(p \times 1)$ vector containing parameters of model $F(\boldsymbol{\theta})$ or of a discrete state space model
$\boldsymbol{\theta}_c$	$(p \times 1)$ vector containing parameters of a continuous state space model
$\boldsymbol{\Theta}$	parameter space
χ^2_η	chi-square random variable with η degrees of freedom
$\boldsymbol{\vartheta}_{i,j}$	vector containing parameters of a GARCH$(\cdot,\cdot)$ process for sensor i and axis j
$\xi_{s\|k}^{i,j}$	probability at time t_k to be in regime s for sensor i and axis j
$\boldsymbol{\zeta}_{i,j}$	vector containing parameters of an ARMA$(\cdot,\cdot)$ process for sensor i and axis j

Other Mathematical Conventions and Symbols used Frequently

$\approx$	approximately equal to
$\text{cov}[\cdot,\cdot]$	covariance operator
$\text{cor}[\cdot,\cdot]$	correlation operator
$\det(\cdot)$	determinant operator
plim	probability limit operator
$\otimes$	quaternion product or Kronecker product
$\mathbf{a} \mapsto \mathbf{b}$	map between $\mathbf{a}$ and $\mathbf{b}$
$h(\cdot) \longleftrightarrow H(\cdot)$	Fourier transform pair
$\|\| \cdot \|\|$	norm
$[\boldsymbol{\omega}]_q$	quaternion equivalent of $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$, i.e. $[\boldsymbol{\omega}]_q = 0 + \omega_1 \cdot \mathbf{i} + \omega_2 \cdot \mathbf{j} + \omega_3 \cdot \mathbf{k}$
$[\mathbf{a}\times]$	skew-symmetric form of vector $\mathbf{a} = [a_1, a_2, a_3]^T$, i.e. $[\mathbf{a}\times] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$

Introduction

Background

Navigation by means of *Global Navigation Satellite Systems* (GNSS) is nowadays a standard approach for performing localization in outdoor environment. Conditions where satellite signals are partially or completely unavailable severely degrade the performance of such systems. Moreover, GNSS sensor bandwidth (typically below 10 Hz) may be too low for some application and the tracking of satellites can be difficult to maintain in vibrating conditions. Also, no information on attitude (e.g. for sensor orientation) is provided by GNSS. A well-accepted and largely proven approach for improving navigation in such situations is to integrate GNSS with inertial sensors. A conventional strapdown *inertial measurement unit* (IMU) is composed of a triad of usually orthogonally mounted accelerometers and gyroscopes observing specific force and angular rate or change, respectively. After initialization, these signals are integrated with respect to time to yield velocity, position and attitude at high rate (typically, higher than 50 Hz). This procedure is the core of a strapdown *inertial navigation system* (INS). The combination of inertial navigation with GNSS is usually performed through bayesian techniques among which the most popular is Kalman filtering. During periods of poor GNSS signal quality or the total absence of its reception, inertial navigation operates in coasting mode, i.e. the navigation states are determined independently from GNSS data. In such case, the overall navigation performance becomes strongly dependent on the errors corrupting inertial signals. These errors are integrated in the INS and their impact consequently grows with time. Correct error modeling and estimation of the systematic components is thus very important for improving navigation and correctly predicting its quality.

Objectives

This dissertation addresses the problem of estimating a navigation solution in terms of position, velocity and attitude using noisy discrete-time signals observed by inertial sensors.

Generally speaking, the design and operation of an INS requires two tasks:

- The *modeling* which is the process of approximating the underlying dynamics that generates the clean (true) signal;
- The *estimation* in which all observations over a certain period of time are used to approximate the current value of the underlying clean signal.

Both tasks are strongly interdependent. An accurate system model can be used for estimating the signal. Conversely, if the true signal is available, it can be exploited for constructing an accurate model of the system dynamics. However, neither the true model nor the clean signals are usually available in the context of inertial navigation.

With this respect, this work focuses on the development of innovative methods for both tasks by using two approaches:

1. The first is achieving correct *a priori* calibration of the individual accelerometers and gyroscopes. Inertial sensors are corrupted by errors (e.g. scale factors, biases, drifts) of deterministic and stochastic nature. Large part of the deterministic errors (e.g. axes misalignment, temperature effects) is compensated through physical models during calibration procedures. These error types and their calibration were investigated in many works (see e.g. Fong et al. [2008], Titterton and Weston [1997]) and will *not* be treated in this study. On the other hand, the stochastic errors contain components which have random behavior or are too complicated to model deterministically (e.g. dynamics-dependent errors, environmental changes, internal sensor noise). These types of errors are modeled using stochastic processes in the navigation filter. The questions of which stochastic processes to use for best describing the random part of the inertial sensors behavior and the determination of the process parameters are the challenging tasks of navigation filter design. The problem of estimating the stochastic process model (from noisy data) is the first objective of this thesis, and will be referred to herein as the *stochastic modeling* problem.
2. The second is the use of multiple IMUs mounted on the same strapdown platform. The benefits of using redundant sensors were demonstrated in several recent studies Stebler et al. [2011b], Waegli [2009], Waegli et al. [2010]. First, the noise level of the overall system can be reduced and defective sensors detected as well as isolated. This improves the accuracy of autonomous navigation, and hence system performance in GNSS-poor conditions Waegli et al. [2010]. Second, the gyroscope and accelerometer noise levels can be estimated directly from the data. This provides a better view on the reality from which the estimation process can benefit. This is indeed an interesting feature regarding the fact that sensor stochastic error modeling is often performed on error signal acquired in static conditions in spite of the supposition that the error behavior may vary as a function of environmental conditions applied to the sensors Guerrier [2009], such as the temperature, electrical power, magnetic fields or the dynamics. Moreover, if no reliable *a priori* calibration of the invidual sensors is available, the direct noise estimation capability of systems based on multiple IMUs enables performing a continuous estimation of the multiple IMU system noise level. Therefore, the second objective of the thesis is to develop models accounting for redundancy in the inertial observations and evaluate the different processing strategies.

Overview

In the first part of the thesis we provide all the necessary concepts related to inertial navigation. Particularly,

- Chapter 1 contains background material related to inertial navigation. Definitions of the reference frames and notations connected to navigation theory are provided. We develop the process of obtaining position, velocity and attitude information from inertial observations, and introduce the concept of Kalman filtering and smoothing at the end of the chapter.
- Chapter 2 discloses the main details of the Kalman filter developed in the framework of this research. We pay special attention in describing the filter prediction and navigation steps as well as alignment procedures.
- In Chapter 3, we introduce the most important concepts related to stochastic models and define the processes, together with associated notations that are used throughout the thesis.

In the second part of the thesis, we treat the stochastic modeling problem in the following chapters:

- Chapter 4 presents the classical methods of stochastic model estimation that are used within navigation community. The main limitations associated to these methods are explained at the end of the chapter. By this means, we provide the justification for the developments brought in the next two chapters.
- Chapter 5 proposes an innovative likelihood-based approach for estimating stochastic models for which the previously introduced methods fail. The algorithm and the related practical issues are described and its performance is tested on simulated as well as on real data.
- A completely new estimation framework is introduced in Chapter 6. The theory related to the construction of this estimator is developed and its performance demonstrated for situations where all the other methods considered in this work, including the method proposed in Chapter 5, are not able to provide an estimate.
- Chapter 7 is dedicated to the practical use of this new estimator presented in the preceding chapter. Several inertial sensors are calibrated for stochastic errors by means of the new estimator. The end of the chapter demonstrates the influence brought by external factors such as vibrations on the sensor error behavior. This provides a strong reason for employing redundant sensors operating simultaneously.

In the third part of the thesis, we investigate several means for handling the redundancy in the inertial observations:

- Chapter 8 concentrates on the algorithmic aspects related to the processing of redundant inertial observations. The end of this chapter is dedicated to the description of the experimental setup realized in the framework of this thesis.

- Chapter 9 is devoted to the presentation of results and discussion. Several tests conducted by car enabled to highlight the advantage of using multiple inertial sensors for systems of lower quality.

Finally, Chapter 10 draws conclusions from the conducted research work and gives recommendations for future investigations.

Part I

Preliminaries

Chapter 1

Fundamentals of Integrated Navigation

1.1 Introduction

As a prerequisite to further work, this chapter reviews the fundamentals of inertial navigation. The related fundamental theory needs to be provided. By that, we define the notations and conventions used in the thesis. The major part of this chapter is hence considered as background material which is extensively based on Farrell [2008], Hofmann-Wellenhof et al. [2003], Tomé [2002]. Additional references are mentioned throughout the text.

Since orientation of systems with respect to specific reference frames is fundamental in inertial navigation, the three main methods of attitude parameterization are summarized in Section 1.2. Section 1.3 defines the various important reference frames along with the associated transformations between them. In Section 1.4, we present the inertial sensor observation model. Then, strapdown inertial mechanization theory is reviewed in Section 1.5. The dynamic model in which inertial observations can be exploited to obtain position, velocity and attitude information is analysed in Section 1.7. Section 1.8 introduces the concept of state space augmentation which enables to account for sensor errors of time-correlated nature. Finally, the way how inertial observations can be combined with external measurements supplied by one or more independent sensors is briefly described in Section 1.9 (Kalman filter theory).

1.2 Attitude Representations

Navigation systems require the transformation of measured and computed quantities between different reference frames. When a vector is represented relative to a specific reference frame, the latter will be indicated by a superscript. For example, $\mathbf{x}^p$ is the vector $\mathbf{x}$ represented with respect to an arbitrary frame p, denoted p-frame. Several mathematical representations can be used to define the attitude of a body with respect to a coordinate reference frame. In the sequel, we consider three of them, namely the direction cosine matrix, Euler angles, and quaternions.

1.2.1 Direction Cosine Matrix

A *direction cosine matrix* (DCM) is a matrix which transforms a vector representation from a Cartesian frame p to another Cartesian frame q and is written as $\mathbf{C}_p^q$ where

$$\mathbf{x}^q = \mathbf{C}_p^q \mathbf{x}^p. \tag{1.2.1}$$

The matrix $\mathbf{C}_p^q$ is orthogonal and has the following properties:

$$\left(\mathbf{C}_p^q\right)^T \mathbf{C}_p^q = \mathbf{C}_p^q \left(\mathbf{C}_p^q\right)^T = \mathbf{I}_{3\times 3} \tag{1.2.2}$$

and

$$\det\left(\mathbf{C}_p^q\right) = 1 \tag{1.2.3}$$

where $\mathbf{I}_{3\times 3}$ is the 3×3 identity matrix, and $\det(\cdot)$ is the determinant operator. Note that the columns of $\mathbf{C}_p^q$ represent the unit vectors in p-frame axes projected along the q-frame axes. In other words, the element of the ith row and jth column of $\mathbf{C}_p^q$ is the cosine of the angle between the ith axis of the q-frame and the jth axis of the p-frame.

1.2.2 Euler Angles

The rotation from a p-frame to a q-frame can be performed by three sequential rotation matrices, noted $\mathbf{C}_j(\alpha_j)$. The elementary rotation matrices are defined as follows:

$$\begin{aligned}
\mathbf{C}_1(\alpha_1) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_1 & \sin\alpha_1 \\ 0 & -\sin\alpha_1 & \cos\alpha_1 \end{bmatrix}, \\
\mathbf{C}_2(\alpha_2) &= \begin{bmatrix} \cos\alpha_2 & 0 & -\sin\alpha_2 \\ 0 & 1 & 0 \\ \sin\alpha_2 & 0 & \cos\alpha_2 \end{bmatrix}, \\
\mathbf{C}_3(\alpha_3) &= \begin{bmatrix} \cos\alpha_3 & \sin\alpha_3 & 0 \\ -\sin\alpha_3 & \cos\alpha_3 & 0 \\ 0 & 0 & 1 \end{bmatrix},
\end{aligned} \tag{1.2.4}$$

for the *Euler angles* $\{\alpha_j : j = 1, 2, 3\}$. Each matrix $\mathbf{C}_j(\alpha_j)$ corresponds to an elementary rotation of an arbitrary Cartesian frame about the $\mathbf{x}_j^p$-axes ($j = 1, 2, 3$) by an angle α_j. The composed rotation is then given by the following product:

$$\mathbf{C}_p^q = \mathbf{C}_3(\alpha_3) \cdot \mathbf{C}_2(\alpha_2) \cdot \mathbf{C}_1(\alpha_1). \tag{1.2.5}$$

Similarly, the inverse transformation is yielded by

$$\mathbf{C}_q^p = \left(\mathbf{C}_p^q\right)^T = \mathbf{C}_1^T(\alpha_1) \cdot \mathbf{C}_2^T(\alpha_2) \cdot \mathbf{C}_3^T(\alpha_3). \tag{1.2.6}$$

Note that if frames p and q are both right-handed or both left-handed Cartesian coordinate frames, the Euler angles representation is equivalent to the DCM of Eq. (1.2.1).

1.2.3 Quaternions

Instead of describing rotations between two frames using the Euler angles $\{\alpha_j\}$, *quaternions* can be used with some advantages (see e.g. Altmann [1986]). A quaternion is a four-parameter attitude representation method based on Euler's theorem, which states that a transformation from one reference frame to another can be performed by a single rotation of magnitude $||\mathbf{u}||$ about the vector $\mathbf{u} = [u_X, u_Y, u_Z]^T$ Tomé [2002]. The four elements of the quaternion, denoted here by the symbol $\mathbf{q}$, are functions of the vector $\mathbf{u}$ and the magnitude of rotation:

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{||\mathbf{u}||}{2}\right) \\ \left(\frac{u_X}{||\mathbf{u}||}\right)\sin\left(\frac{||\mathbf{u}||}{2}\right) \\ \left(\frac{u_Y}{||\mathbf{u}||}\right)\sin\left(\frac{||\mathbf{u}||}{2}\right) \\ \left(\frac{u_Z}{||\mathbf{u}||}\right)\sin\left(\frac{||\mathbf{u}||}{2}\right) \end{bmatrix} \tag{1.2.7}$$

such that

$$||\mathbf{u}|| = \sqrt{u_X^2 + u_Y^2 + u_Z^2}. \tag{1.2.8}$$

A quaternion q can be defined as a hypercomplex number which is the sum of a scalar part, q_0, and a vector part, denoted as $\boldsymbol{q}$, i.e.

$$q = q_0 + \boldsymbol{q}. \tag{1.2.9}$$

The vector $\boldsymbol{q}$ is an ordinary (3×1) vector defined as

$$\boldsymbol{q} = q_1 \cdot \mathbf{i} + q_2 \cdot \mathbf{j} + q_3 \cdot \mathbf{k} \tag{1.2.10}$$

where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ denote the orthonormal basis which can be written as

$$\begin{aligned} \mathbf{i} &= [1, 0, 0] \\ \mathbf{j} &= [0, 1, 0] \\ \mathbf{k} &= [0, 0, 1]. \end{aligned} \tag{1.2.11}$$

Therefore, we have

$$q = q_0 + q_1 \cdot \mathbf{i} + q_2 \cdot \mathbf{j} + q_3 \cdot \mathbf{k} \tag{1.2.12}$$

with q_0, q_1, q_2, q_3 being real numbers satisfying

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1, \tag{1.2.13}$$

and $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ are such that

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1 \tag{1.2.14}$$

and

$$\begin{aligned} \mathbf{i} \cdot \mathbf{j} &= -\mathbf{j} \cdot \mathbf{i} = \mathbf{k} \\ \mathbf{j} \cdot \mathbf{k} &= -\mathbf{k} \cdot \mathbf{j} = \mathbf{i} \\ \mathbf{k} \cdot \mathbf{i} &= -\mathbf{i} \cdot \mathbf{k} = \mathbf{j}. \end{aligned} \tag{1.2.15}$$

The operations of addition, substraction, and multiplication by a scalar are done in the same manner as in vector algebra. The multiplication of $\mathbf{q}$ with another quaternion, $\mathbf{p} = [p_0, p_1, p_2, p_3]^T$, denoted by the symbol $\otimes$, is defined by

$$\begin{aligned}\mathbf{q} \otimes \mathbf{p} &= q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3 \\ &+ (q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2) \cdot \mathbf{i} \\ &+ (q_0p_2 + q_2p_0 - q_1p_3 + q_3p_1) \cdot \mathbf{j} \\ &+ (q_0p_3 + q_3p_0 + q_1p_2 - q_2p_1) \cdot \mathbf{k}.\end{aligned}$$

Extensive details about quaternion algebra can be found in Altmann [1986], Tomé [2002]. The relationship between the DCM, $\mathbf{C}_p^q$, and its corresponding attitude quaternion, noted $\mathbf{q}_p^q$, is

$$\mathbf{C}_p^q = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix}. \tag{1.2.16}$$

1.2.4 Infinitesimal Rotation

Suppose that the angles $\{\alpha_j\}$ in the elementary rotation matrices defined in Eq. (1.2.4) are infinitesimal, and let $\delta\boldsymbol{\alpha} = [\delta\alpha_1, \delta\alpha_2, \delta\alpha_3]^T$ denote the infinitesimal rotation between frame p and frame q. Then, $\cos(\delta\alpha_j) \approx 1$, $\sin(\delta\alpha_j) \approx \delta\alpha_j$, and $\delta\alpha_i\delta\alpha_j \approx 0$ for $i, j = 1, 2, 3$.

Hence, Eq. (1.2.4) becomes

$$\begin{aligned}\mathbf{C}_1(\delta\alpha_1) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \delta\alpha_1 \\ 0 & -\delta\alpha_1 & 1 \end{bmatrix}, \\ \mathbf{C}_2(\delta\alpha_2) &= \begin{bmatrix} 1 & 0 & -\delta\alpha_2 \\ 0 & 1 & 0 \\ \delta\alpha_2 & 0 & 1 \end{bmatrix}, \\ \mathbf{C}_3(\delta\alpha_3) &= \begin{bmatrix} 1 & \delta\alpha_3 & 0 \\ -\delta\alpha_3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.\end{aligned} \tag{1.2.17}$$

Note that these matrices are no longer orthogonal. Neglecting second- and higher-order terms, the composed transformation may be approximated by

$$\mathbf{C}_3(\delta\alpha_3) \cdot \mathbf{C}_2(\delta\alpha_2) \cdot \mathbf{C}_1(\delta\alpha_1) = \mathbf{I} - [\delta\boldsymbol{\alpha}\times] \tag{1.2.18}$$

where the notation $[\delta\boldsymbol{\alpha}\times]$ defines the (3×3) skew-symmetric matrix built from vector $\delta\boldsymbol{\alpha}$, i.e.

$$[\delta\boldsymbol{\alpha}\times] = \begin{bmatrix} 0 & -\delta\alpha_3 & \delta\alpha_2 \\ \delta\alpha_3 & 0 & -\delta\alpha_1 \\ -\delta\alpha_2 & \delta\alpha_1 & 0 \end{bmatrix}. \tag{1.2.19}$$

Furthermore, the order of rotation becomes irrelevant due to the structure of the infinitesimal rotation matrix.

1.2.5 Rotating Reference Frames

Reference frames may rotate arbitrarily with respect to one another. For example, the q-frame may rotate with respect to the p-frame. It is necessary to calculate derivatives of attitude representations for coordinate systems experiencing relative rotation.

Consider the vector $\boldsymbol{\omega}_{pq}^{q} = [\omega_1, \omega_2, \omega_3]^T$ representing the angular-rate vector of the q-frame with respect to the p-frame expressed in the q-frame. If the rotation rate between the p-frame and the q-frame is very small, it can be shown that the differential equation of orthogonal transformations is given by Farrell [2008]

$$\dot{\mathbf{C}}_{p}^{q} = \mathbf{C}_{p}^{q}\boldsymbol{\Omega}_{qp}^{p} \tag{1.2.20}$$

where $\boldsymbol{\Omega}_{qp}^{p} = [\boldsymbol{\omega}_{qp}^{p}\times]$.

Similarly, the time-derivative of the Euler angles $\{\alpha_j\}$ can be shown to be Tomé [2002]

$$\dot{\alpha}_1 = [\omega_2 \sin(\alpha_1) + \omega_3 \cos(\alpha_1)] \cdot \tan(\alpha_2) + \omega_1 \tag{1.2.21}$$
$$\dot{\alpha}_2 = \omega_2 \cos(\alpha_1) - \omega_3 \sin(\alpha_1) \tag{1.2.22}$$
$$\dot{\alpha}_3 = [\omega_2 \sin(\alpha_1) + \omega_3 \cos(\alpha_1)] \cdot \sec(\alpha_2). \tag{1.2.23}$$

Note that the Eq. (1.2.21-1.2.23) have a singularity at some orientations, specifically when α_2 equals ± 90 deg.

Finally, the quaternion $\mathbf{q}_{p}^{q}$ propagates in time in accordance to the following equation Tomé [2002]:

$$\dot{\mathbf{q}}_{b}^{q} = \frac{1}{2}\mathbf{q}_{p}^{q} \otimes \left[\boldsymbol{\omega}_{pq}^{q}\right]_{q} \tag{1.2.24}$$

with $[\boldsymbol{\omega}_{pq}^{q}]_q$ being the quaternion equivalent of $\boldsymbol{\omega}_{pq}^{q}$, i.e. $[\boldsymbol{\omega}_{pq}^{q}]_q = 0 + \omega_1 \cdot \mathbf{i} + \omega_2 \cdot \mathbf{j} + \omega_3 \cdot \mathbf{k}$.

1.3 Reference Frame Definitions

In this section, we define the reference frames and focus on the transformations between them.

1.3.1 The Inertial Frame (i-Frame)

The inertial frame, noted i-frame, is a nonaccelerating and nonrotating reference frame that is at rest or subject to a uniform translational motion. In such a frame, the laws of Newtonian mechanics are valid. Consider a point P located at a distance $\mathbf{r}$ from the origin of the i-frame. Let $\ddot{\mathbf{r}}^i$ and $\mathbf{F}^i$ be the acceleration and forces encountered by a body of mass m_i along the three axes of the i-frame, respectively. Newton's second law can be written as

$$\mathbf{F}^i = m_i \ddot{\mathbf{r}}^i. \tag{1.3.1}$$

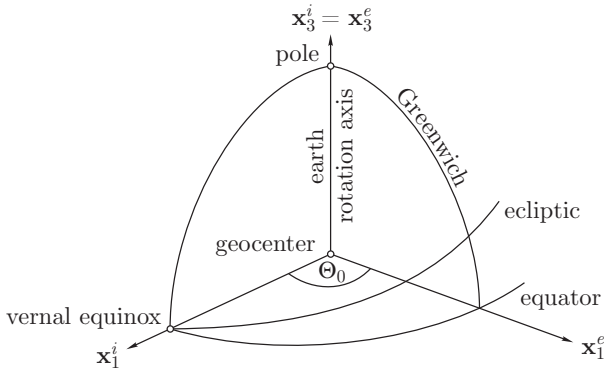

Figure 1.1: Equatorial coordinate systems (adapted from Hofmann-Wellenhof et al. [2003]).

If the body is subject to the gravitational acceleration represented by the vector $\breve{\mathbf{g}}^i$, Eq. (1.3.1) is extended to

$$m_i \ddot{\mathbf{r}}^i = \mathbf{F}^i + m_g \breve{\mathbf{g}}^i \tag{1.3.2}$$

where m_g is the gravitational mass. Applying Einstein's principle of equivalence leads to $m_i = m_g = m$. Replacing m_i and m_g by m in Eq. (1.3.2) and dividing both sides by m leads to

$$\ddot{\mathbf{r}}^i = \mathbf{f}^i + \breve{\mathbf{g}}^i \tag{1.3.3}$$

where $\mathbf{f}^i$ denotes the *specific force* vector (force per unit mass).

An inertial system may be approximated by a so-called *quasi-inertial system* that is inertial with respect to rotation and acceleration up to the measurement accuracy of the navigation instrument. A quasi-inertial system is a geocentric system with celestial (space-fixed) orientation. The accelerations occurring due to the fact that the Earth is orbiting the sun as well as the rotation of the solar system with respect to the center of galaxy can be neglected for all conventional inertial sensors as they are well below instrumental resolution and noise level. Hence the following definition of the i-frame can be formulated.

Definition 1.3.1. *An inertial frame is a celestial frame with origin at the center of mass of the Earth, such that the $\mathbf{x}_1^i$-axis points towards the vernal equinox and is, thus, the intersection line between the equatorial and the ecliptic plane, the $\mathbf{x}_3^i$-axis points towards the mean celestial pole, and the $\mathbf{x}_2^i$-axis completes the system to a 3D right-handed Cartesian system (see Figure 1.1).*

The rotation of the e-frame with respect to the i-frame is given by

$$\boldsymbol{\omega}_{ie}^e = \begin{bmatrix} 0 & 0 & \omega_{ie} \end{bmatrix}^T \tag{1.3.4}$$

where $\omega_{ie} = 7.292115 \cdot 10^{-5}$ rad/s is the magnitude of the rotation rate of the Earth[1].

[1]Strictly speaking, ω_{ie} is not constant in time. However, we assume that $\dot{\boldsymbol{\omega}}_{ie} \approx 0$, which is reasonable in the context of navigation.

1.3.2 The Earth Frame (e-Frame)

The Earth frame, denoted e-frame, is an *Earth-Centered Earth-Fixed* (ECEF) frame that is defined as follows.

Definition 1.3.2. *An Earth frame is an equatorial frame with origin at the center of mass of the Earth, such that the $\mathbf{x}_1^e$-axis points towards the Greenwich meridian, the $\mathbf{x}_3^e$-axis points towards the mean direction of the rotation axis of the Earth, and the $\mathbf{x}_2^e$-axis completes the system to a 3D right-handed Cartesian system.*

Examples of important realizations are the *International Terrestrial Reference Frame* (ITRF) and the *World Geodetic System* WGS-84 being the reference frame of the American *Global Positioning System* (GPS).

As shown in Figure 1.1, the transformation from the e-frame to the i-frame is achieved by a rotation about the $\mathbf{x}_3$-axis:

$$\mathbf{C}_e^i = \mathbf{C}_3(-\Theta_0) \tag{1.3.5}$$

where Θ_0 is the *hour angle*, i.e. the product of *Greenwich Sideral Time* (GST) and ω_{ie}. Referring to ECEF frames, a geocentric ellipsoid of revolution can be associated. Hence, a point can either be expressed in terms of *Cartesian coordinates* (x_1^e, x_2^e, x_3^e), or by its *ellipsoidal coordinates*, (ϕ, λ, h) with the latitude ϕ, the longitude λ, and the height h. The conversion from Cartesian to ellipsoidal coordinates can be done as follows (see Figure 1.2):

$$\mathbf{x}^e = \begin{bmatrix} x_1^e \\ x_2^e \\ x_3^e \end{bmatrix} = \begin{bmatrix} (R_P + h)\cos(\phi)\cos(\lambda) \\ (R_P + h)\cos(\phi)\sin(\lambda) \\ \left[R_P(1-e^2) + h\right]\sin\phi \end{bmatrix} \tag{1.3.6}$$

where R_P is the *prime vertical radius of curvature* given by

$$R_P = \frac{a}{\left(1 - e^2 \sin^2 \varphi\right)^{1/2}}. \tag{1.3.7}$$

The quantities a and e are the semi-major axis and first numerical eccentricity of the reference ellipsoid, respectively. The latter is given by

$$e^2 = \frac{a^2 - b^2}{a^2}. \tag{1.3.8}$$

with b the semi-minor axis.

The inversion of Eq. (1.3.6) directly yields

$$\lambda = \arctan\left(\frac{x_2^e}{x_1^e}\right) \tag{1.3.9}$$

while the computation of ϕ and h requires two auxiliary quantities

$$p = \sqrt{(x_1^e)^2 + (x_2^e)^2} = (R_P + h)\cos(\phi), \tag{1.3.10}$$

$$\psi = \arctan\left(\frac{x_3^e \cdot a}{p \cdot b}\right) \tag{1.3.11}$$

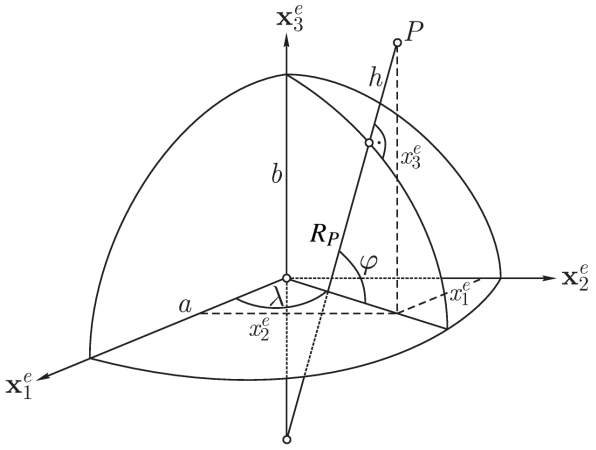

Figure 1.2: Cartesian and ellipsoidal coordinates (adapted from Hofmann-Wellenhof et al. [2003]).

where p is the orthogonal distance of $\mathbf{x}^e$ from the symmetry axis of the ellipsoid. The latitude is then computed by

$$\phi = \arctan\left(\frac{x_3^e + e'^2 b \sin^3(\psi)}{p - e^2 a \cos^3(\psi)}\right) \tag{1.3.12}$$

where e' is the second numerical eccentricity of the ellipsoid, given by

$$e'^2 = \frac{a^2 - b^2}{b^2}. \tag{1.3.13}$$

Once ϕ is computed, the height can be obtained from Eq. (1.3.10) by

$$h = \frac{p}{\cos(\phi)} - R_P. \tag{1.3.14}$$

1.3.3 The Local-level Frame (l-Frame)

The local-level frame, noted l-frame, usually serves as a direct reference with respect to geodetic observations. It can be defined anywhere on or near the surface of the Earth.

Definition 1.3.3. *A local-level frame is a local geodetic frame with arbitrary origin, e.g. a point on the Earth surface (topocenter), such that the $\mathbf{x}_1^l$-axis points to North, the $\mathbf{x}_2^l$-axis points to East, and the $\mathbf{x}_3^l$-axis points to the local nadir (down).*

The NED frame (North-East-Down) is right-handed. The nadir direction of a point is defined by its *astronomical latitude* and *longitude*, respectively noted Φ and Λ. If the astronomical coordinates are replaced by the ellipsoidal latitude ϕ and longitude λ, the $\mathbf{x}_3^l$-axis corresponds to the ellipsoidal normal. For the purpose of navigation, the difference between astronomical

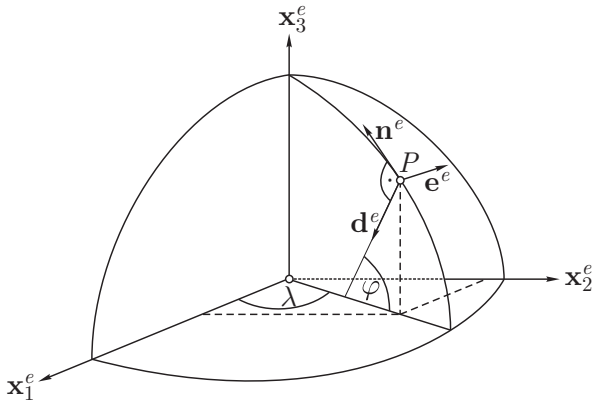

Figure 1.3: ECEF and local-level frame (adapted from Hofmann-Wellenhof et al. [2003]).

and ellipsoidal local-level frames can often be neglected (see Section 1.5.4). The resulting frame is called ellipsoidal tangential frame rather than local-level frame. Since the l-frame is singular at the poles where the North direction cannot be defined, the *wander frame* is used in practice (see e.g. Titterton and Weston [1997] for details about the wander frame).

To derive the orientation with respect to the right-handed ECEF frame (e-frame), the NED frame is chosen as l-frame (see Figure 1.3). Given a point P with ellipsoidal latitude ϕ and longitude λ, the frame $\mathbf{x}_j^l$-axes ($j = 1, 2, 3$) expressed in the e-frame are denoted as $\mathbf{n}^e$, $\mathbf{e}^e$, and $\mathbf{d}^e$ and are given by

$$\mathbf{d}^e = \begin{bmatrix} -\cos(\phi)\cos(\lambda) \\ -\cos(\phi)\sin(\lambda) \\ -\sin(\phi) \end{bmatrix}, \ \mathbf{n}^e = -\frac{\partial \mathbf{d}^e}{\partial \phi}, \ \mathbf{e}^e = -\frac{1}{\cos(\phi)}\frac{\partial \mathbf{d}^e}{\partial \lambda} \tag{1.3.15}$$

where $\mathbf{n}^e$ and $\mathbf{e}^e$ are deduced from $\mathbf{d}^e$ by partial derivatives. The rotation matrix $\mathbf{C}_l^e$ for the coordinate transformation from the l-frame to the e-frame is

$$\mathbf{C}_l^e = \begin{bmatrix} -\sin(\phi)\cos(\lambda) & -\sin(\lambda) & -\cos(\phi)\cos(\lambda) \\ -\sin(\phi)\sin(\lambda) & \cos(\lambda) & -\cos(\phi)\sin(\lambda) \\ \cos(\phi) & 0 & -\sin(\phi) \end{bmatrix}. \tag{1.3.16}$$

Note that since the origins of the e-frame and the l-frame are not identical, only difference vectors may be transformed if $\mathbf{C}_l^e$ is solely used without shift vector.

1.3.4 The Navigation Frame (n-Frame)

The navigation frame, noted n-frame, can be defined as follows.

Definition 1.3.4. *A navigation frame refers to the coordinate frame in which the INS is mechanized.*

In this study, the l-frame is chosen as n-frame for the mechanization of the strapdown INS. The reasons for this choice are Tomé [2002]:

- Navigation information is most commonly required in terms of North, East, Down velocity components and geographic coordinates which are directly obtained;
- The attitude of the vehicle with respect to l-frame is readily available;
- The representation of the Earth's gravity is simplified (e.g. deflections of the vertical are negligible and a normal gravity model can be used).

1.3.5 The Computer Frame (c-Frame)

The computer frame, noted c-frame, can be defined as follows.

Definition 1.3.5. *The computer frame is the frame that the INS computer assumes to be the true n-frame, i.e. the l-frame, with origin at the INS computed position.*

Hence, the c-frame is generated by the navigation computer from the measurement data. Because of numerical calculation errors, measurement errors and other errors, the c-frame will differ in orientation from the true l-frame by three small independent rotations Gosiewski and Ortyl [1998]. Consider $\delta\phi$ and $\delta\lambda$ the errors in the computed geodetic latitude and longitude, respectively. These errors can be projected onto the local North and vertical axes for a given latitude ϕ, yielding the following rotation vector:

$$\delta\boldsymbol{\theta} = \begin{bmatrix} -\delta\phi \\ \delta\lambda\cos(\phi) \\ -\delta\lambda\sin(\phi) \end{bmatrix}. \tag{1.3.17}$$

These rotations describe completely the difference in orientation of the computed c-frame in relation to the true l-frame, and the corresponding DCM can be written as

$$\mathbf{C}_l^c = \mathbf{I} - [\delta\boldsymbol{\theta}\times]. \tag{1.3.18}$$

Figure 1.4 schematizes the relationships between c-frame and l-frame (i.e. n-frame).

1.3.6 The Body Frame (b-Frame)

The body frame, noted b-frame, can be defined as follows.

Definition 1.3.6. *The body frame is a right-handed 3D Cartesian frame rigidly connected with the moving object (e.g. vehicle) and is used to determine the relative orientation or attitude of the object with respect to the chosen l-frame.*

Usually, the origin of the b-frame is situated at a specific point within the object (e.g. center of mass) and the axes coincide with the principal rotation axes of the object. The parameters used to describe the three-dimensional attitude are often denoted *roll*, *pitch*, and *yaw*. Here, we assume that the b-frame coincides with the orthogonal frame aligned with the roll, pitch, and heading axes of a vehicle, i.e. the $\mathbf{x}_1^b$-axis points forwards, the $\mathbf{x}_3^b$-axis points down, and the $\mathbf{x}_2^b$-axis completes the system to a 3D right-handed Cartesian system.

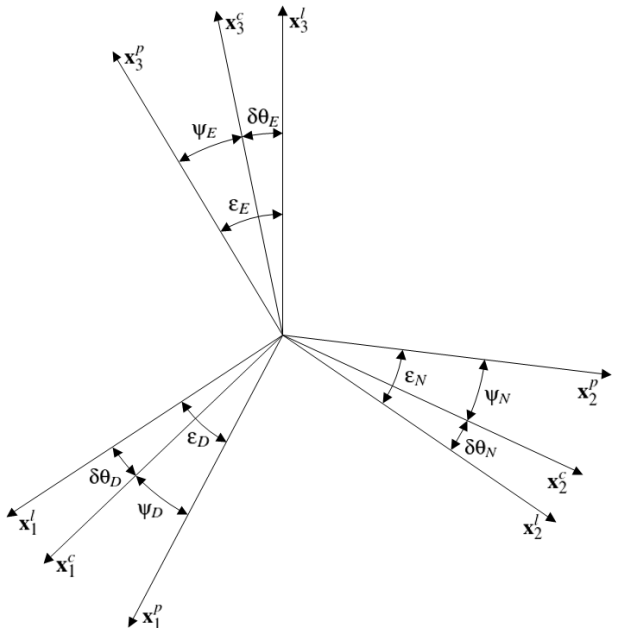

Figure 1.4: Relationships between l-frame (or n-frame), c-frame and p-frame.

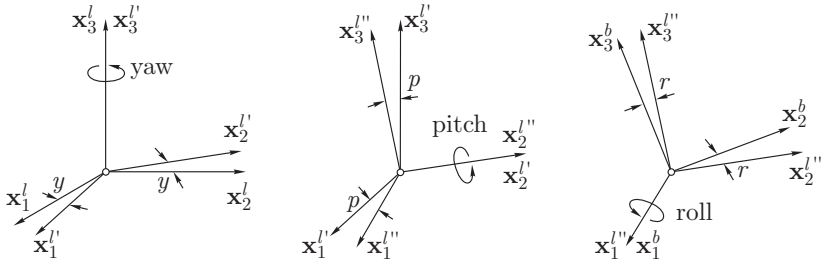

Figure 1.5: Rotations by yaw, pitch, and roll (adapted from Hofmann-Wellenhof et al. [2003]).

The transformation between the b-frame and the l-frame is usually described by three rotation angles about the $\mathbf{x}_j^l$-axes or the $\mathbf{x}_j^b$-axes ($j = 1, 2, 3$). Here, we define the attitude for the b-frame with respect to the l-frame by choosing the following order for the stepwise rotations (see Figure 1.5):

$$\mathbf{C}_l^b = \mathbf{C}_1(r) \cdot \mathbf{C}_2(p) \cdot \mathbf{C}_3(y) \tag{1.3.19}$$

where r, p and y are the roll, pitch and yaw angles, respectively.

1.3.7 The Platform Frame (p-Frame)

The platform frame, noted p-frame, is the "analytic-platform" frame which can be defined as follows.

Definition 1.3.7. *The platform frame is the frame in which the transformed accelerations and angular rates from the accelerometers and gyroscopes, respectively, are resolved.*

The inertial navigation computer estimates the rotational transformation , which under ideal conditions, transforms vectors from the b-frame to the true l-frame (or n-frame). However, because of computation errors, this function does not transform vectors to the l-frame but to the p-frame. Hence,

$$\hat{\mathbf{C}}_b^l = \mathbf{C}_b^p = \mathbf{C}_l^p \mathbf{C}_b^l. \tag{1.3.20}$$

From Figure 1.4, it can be seen that the angles $\boldsymbol{\psi} = [\psi_N, \psi_E, \psi_D]^T$ represent the difference in orientation of the p-frame with respect to c-frame. Hence, assuming small attitude errors, we have

$$\mathbf{C}_c^p = \mathbf{I} - [\boldsymbol{\psi}\times]. \tag{1.3.21}$$

The orientation error of the p-frame with respect to the true l-frame is refered to as the $\boldsymbol{\varepsilon}$ angle and is yielded by

$$\mathbf{C}_l^p = \mathbf{I} - [\boldsymbol{\varepsilon}\times]. \tag{1.3.22}$$

Since $\mathbf{C}_l^p = \mathbf{C}_c^p \mathbf{C}_l^c$, we have

$$\boldsymbol{\varepsilon} = \boldsymbol{\psi} + \delta\boldsymbol{\theta}. \tag{1.3.23}$$

1.4 Inertial Sensor Observation Model

Accelerometers and gyroscopes provide specific force and angular rate measurements, respectively noted as $\breve{\mathbf{f}}^b$ and $\breve{\boldsymbol{\omega}}_{ib}^b$. They are measured along and about three mutually orthogonal axes, forming an IMU. Both signals are corrupted by errors $\Delta\mathbf{f}^b$ and $\Delta\boldsymbol{\omega}_{ib}^b$ of deterministic and stochastic nature, yielding the following model

$$\breve{\mathbf{f}}^b = \mathbf{f}^b + \Delta\mathbf{f}^b \tag{1.4.1}$$
$$\breve{\boldsymbol{\omega}}_{ib}^b = \boldsymbol{\omega}_{ib}^b + \Delta\boldsymbol{\omega}_{ib}^b \tag{1.4.2}$$

with their respective error terms

$$\Delta\mathbf{f}^b = \mathbf{S}_a\mathbf{f}^b + \mathbf{N}_a\mathbf{f}^b + \mathbf{b}_a + \mathbf{w}_a \tag{1.4.3}$$
$$\Delta\boldsymbol{\omega}_{ib}^b = \mathbf{S}_g\boldsymbol{\omega}_{ib}^b + \mathbf{N}_g\boldsymbol{\omega}_{ib}^b + \mathbf{b}_g + \mathbf{w}_g \tag{1.4.4}$$

where $\mathbf{S}_\bullet$ is a diagonal matrix of scale factor errors, $\mathbf{N}_\bullet$ is a skew-symmetric matrix accounting for cross-coupling errors, $\mathbf{b}_\bullet$ is the measurement bias, and $\mathbf{w}_\bullet$ includes remaining errors which are treated as random errors (e.g. random bias), including correlated errors, random walk and white noise. Given estimates of $\Delta\hat{\mathbf{f}}^b$ and $\Delta\hat{\boldsymbol{\omega}}_{ib}^b$, the signal output by accelerometers and gyroscopes can be written as

$$\hat{\mathbf{f}}^b = \breve{\mathbf{f}}^b + \Delta\hat{\mathbf{f}}^b \tag{1.4.5}$$
$$\hat{\boldsymbol{\omega}}_{ib}^b = \breve{\boldsymbol{\omega}}_{ib}^b + \Delta\hat{\boldsymbol{\omega}}_{ib}^b. \tag{1.4.6}$$

All the calibration error components defined in model (1.4.3) and (1.4.4) can be defined as follows.

Definition 1.4.1. *The scale factor error,* $\mathbf{S}_\bullet$*, can be defined as the ratio of a change in the output to a change in the input intended to be measured, and is commonly expressed in parts per million (ppm), or as a percentage figure for the classes of sensors of lower performance Titterton and Weston [1997]. Scale factor errors may include a constant part which is dominant under normal operating conditions, and varying parts that also depend on the dynamics. The dominant part of these errors is usually determined in laboratory conditions, while the residual is lumped into the random errors.*

Definition 1.4.2. *Cross-coupling errors,* $\mathbf{N}_\bullet$*, arise through the non-orthogonality of the axes defining the sensor frame. They may be expressed as ppm or a percentage of the applied measured quantity. Under high dynamics, these errors become less significant and may be assigned to random errors.*

Definition 1.4.3. *The measurement bias,* $\mathbf{b}_\bullet$*, results from manufacturing imperfections in the sensors and may include constant and time-variable parts. The latter account for unmodeled effects (e.g. A/D quantization noise, residual temperature effects) and are usually modeled as stochastic noise.*

Each of the errors described will, in general, include systematic (fixed terms, and to a large extent temperature induced variations) and random effects (switch-on to switch-on and in-run variations) that can be compensated using deterministic and stochastic models, respectively.

1.5 Local-level Strapdown Mechanization

1.5.1 Acceleration in an Arbitrary Frame

Let $\mathbf{r}$ be the vector from the i-frame origin to a point P. Using the Coriolis theorem Thornton and Marion [2004], it can be shown that

$$\ddot{\mathbf{r}}^i = \mathbf{C}_a^i \left[2\boldsymbol{\Omega}_{ia}^a \mathbf{v}^a + \boldsymbol{\Omega}_{ia}^a \boldsymbol{\Omega}_{ia}^a \mathbf{r}^a + \dot{\boldsymbol{\Omega}}_{ia}^a \mathbf{r}^a + \ddot{\mathbf{r}}^a \right] \tag{1.5.1}$$

where $\mathbf{r}^a$ represents the vector from an arbitrary frame (a-frame) origin to P, and $\boldsymbol{\Omega}_{ia}^a = [\boldsymbol{\omega}_{ia}^a \times]$. Solving Eq. (1.5.1) for the second derivative of $\mathbf{r}^a$ and using Eq. (1.3.3) yields

$$\ddot{\mathbf{r}}^a = \mathbf{C}_b^a \mathbf{f}^b + \breve{\mathbf{g}}^a - 2\boldsymbol{\Omega}_{ia}^a \mathbf{v}^a - \left(\boldsymbol{\Omega}_{ia}^a \boldsymbol{\Omega}_{ia}^a + \dot{\boldsymbol{\Omega}}_{ia}^a \right) \mathbf{r}^a. \tag{1.5.2}$$

In the sequel, the navigation equations are derived for several choices of the a-frame before moving to the l-frame.

1.5.2 Velocity Dynamics in i-Frame

If the a-frame is selected to be the i-frame, Eq. (1.5.2) becomes

$$\ddot{\mathbf{r}}^i = \mathbf{C}_b^i \mathbf{f}^b + \breve{\mathbf{g}}^i(\mathbf{r}). \tag{1.5.3}$$

1.5.3 Velocity Dynamics in e-Frame

If the e-frame is chosen, Eq. (1.5.2) reduces to

$$\ddot{\mathbf{r}}^e = \mathbf{C}_b^e \mathbf{f}^b + \breve{\mathbf{g}}^e(\mathbf{r}) - 2\boldsymbol{\Omega}_{ie}^e \mathbf{v}^e - \boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e \tag{1.5.4}$$

where $\boldsymbol{\Omega}_{ie}^e = [\boldsymbol{\omega}_{ie}^e \times]$. The term $2\boldsymbol{\Omega}_{ie}^e \mathbf{v}^e$ is the *Coriolis acceleration* that appears when the vehicle moves with respect to a rotating reference frame, and $\boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e$ is the *centrifugal acceleration* experienced by the vehicle due to the Earth rotation. Typically, $\breve{\mathbf{g}}^e(\mathbf{r})$ and $-\boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e$ are combined to form the *local gravity vector* $\mathbf{g}^e$, i.e.

$$\mathbf{g}^e = \breve{\mathbf{g}}^e(\mathbf{r}) - \boldsymbol{\Omega}_{ie}^e \boldsymbol{\Omega}_{ie}^e \mathbf{r}^e. \tag{1.5.5}$$

Then, by substituting $\dot{\mathbf{v}}^e$ for $\ddot{\mathbf{r}}^e$ and using Eq. (1.5.5), the acceleration vector $\dot{\mathbf{v}}^e$ developed in the e-frame is

$$\dot{\mathbf{v}}^e = \mathbf{C}_b^e \mathbf{f}^b + \mathbf{g}^e - 2\boldsymbol{\Omega}_{ie}^e \mathbf{v}^e. \tag{1.5.6}$$

1.5.4 Velocity Dynamics in l-Frame

Eq. (1.5.6) can be transformed into the l-frame by using the following relationship:

$$\mathbf{v}_e^l = \mathbf{C}_e^l \mathbf{v}^e \tag{1.5.7}$$

where $\mathbf{v}_e^l = [v_N, v_E, v_D]^T$. Taking the time derivative of Eq. (1.5.7) yields

$$\dot{\mathbf{v}}_e^l = \mathbf{C}_b^l \left(\boldsymbol{\Omega}_{le}^e \mathbf{v}^e + \dot{\mathbf{v}}^e \right). \tag{1.5.8}$$

Removing $\dot{\mathbf{v}}^e$ from the previous equation by using Eq. (1.5.7) and after some manipulation, the final form of the navigation equation in the l-frame can be written as

$$\dot{\mathbf{v}}_e^l = \mathbf{C}_b^l \mathbf{f}^b + \mathbf{g}^l - \left(\boldsymbol{\Omega}_{el}^l + 2\boldsymbol{\Omega}_{ie}^l \right) \mathbf{v}_e^l \tag{1.5.9}$$

with $\boldsymbol{\Omega}_{el}^l = [\boldsymbol{\omega}_{el}^l \times]$, $\boldsymbol{\Omega}_{ie}^l = [\boldsymbol{\omega}_{ie}^l \times]$ where

$$\boldsymbol{\omega}_{ie}^l = \begin{bmatrix} \omega_{ie}\cos(\varphi) & 0 & -\omega_{ie}\sin(\varphi) \end{bmatrix}^T \text{ and } \boldsymbol{\omega}_{el}^l = \begin{bmatrix} \dot{\lambda}\cos(\varphi) & -\dot{\varphi} & -\dot{\lambda}\sin(\varphi) \end{bmatrix}^T. \tag{1.5.10}$$

Both $\dot{\varphi}$ and $\dot{\lambda}$ are members of $\dot{\mathbf{r}}_e^l$ which will be defined by Eq. (1.5.12) below. The $\mathbf{g}^l$ term is given by

$$\mathbf{g}^l = \begin{bmatrix} \xi g & -\eta g & g_0 + \Delta g \end{bmatrix}^T \tag{1.5.11}$$

where ξ and η are the *meridian* and *prime deflection of the vertical*, respectively. The term g is the sum of the gravity magnitude associated with the reference ellipsoid (*normal gravity* g_0) determined at height h, and the *gravity anomaly* Δg. The normal gravity g_0 can be calculated using models such as proposed by Committee [1997][2].

[2]The use of more detailed gravity models is needed only for inertial sensors of high quality. Moreover, the largest deflection encountered over the entire Earth is in the order of 1 arc-min. For increased accuracy, it would be desirable to compensate for the deflection and gravity anomaly terms. In the absence of detailed specifications, these terms are normally neglected and accepted as sources of error Jekeli [2000], Tomé [2002].

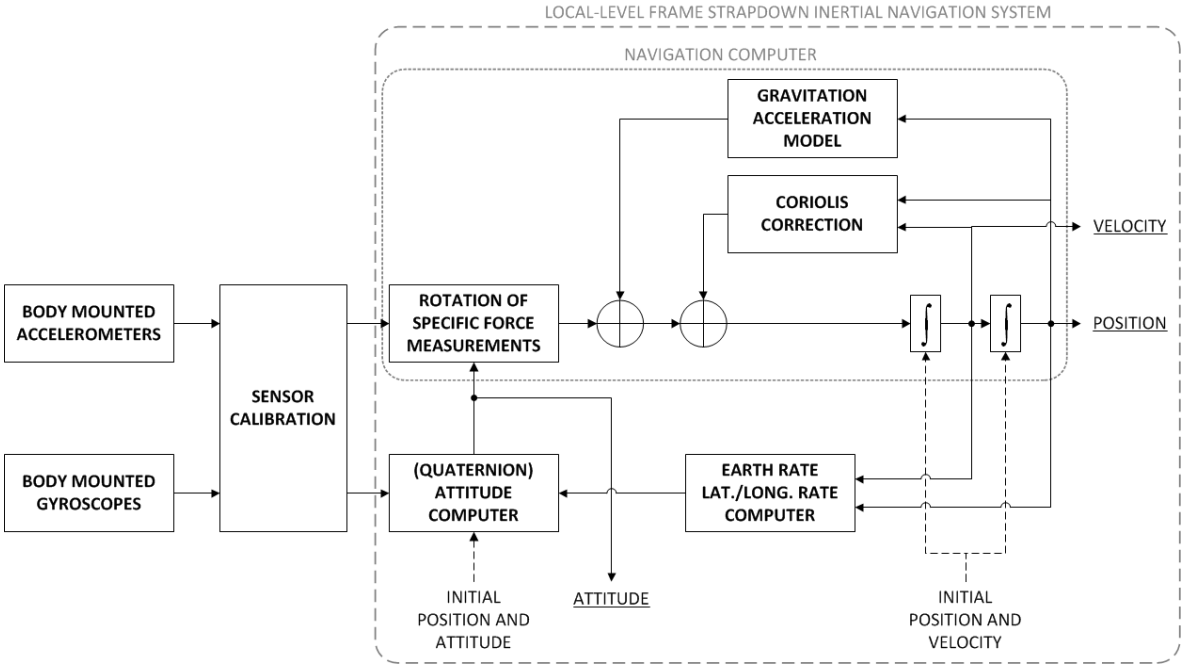

Figure 1.6: Strapdown inertial navigation system in l-frame.

1.5.5 Position Dynamics in l-Frame

The time derivative of the position states $\mathbf{r}_e^l$ can be linked to $\mathbf{v}_e^l$ through

$$\dot{\mathbf{r}}_e^l = \mathbf{D}^{-1}\mathbf{v}_e^l \tag{1.5.12}$$

where $\mathbf{r}_e^l = [\phi, \lambda, h]^T$, and the matrix

$$\mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{R_M+h} & 0 & 0 \\ 0 & \frac{1}{(R_P+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{1.5.13}$$

enables the velocity conversion from the cartesian (NED) to curvilinear coordinates. The *meridian radius of curvature* is given by

$$R_M = \frac{a\left(1-e^2\right)}{\left(1-e^2\sin^2(\varphi)\right)^{3/2}} \tag{1.5.14}$$

where a and e were defined in Section 1.3.2, and the prime vertical radius of curvature, R_P, was defined in Eq. (1.3.7).

1.5.6 Attitude Dynamics in l-Frame

In this study, the quaternion form is chosen for representing the attitude of a vehicle. The reasons are:

- It is computationally simpler than the DCM method;
- The singularity problem of the Euler angles representation is avoided.

The information about the vehicle's attitude with respect to the l-frame is contained in the quaternion $\mathbf{q}_b^l$. According to Eq. (1.2.24), its time propagation is described by

$$\dot{\mathbf{q}}_b^l = \frac{1}{2}\mathbf{q}_b^l \otimes \left[\boldsymbol{\omega}_{lb}^b\right]_q \tag{1.5.15}$$

where $\left[\boldsymbol{\omega}_{lb}^b\right]_q$ is the quaternion equivalent of

$$\boldsymbol{\omega}_{lb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_l^b \boldsymbol{\omega}_{il}^l \tag{1.5.16}$$

1.5.7 Summary

Finally, Eq. (1.5.12), (1.5.9) and (1.5.15) can be combined to form the state vector $\mathbf{x}$ in the l-frame:

$$\dot{\mathbf{x}}^l = \begin{bmatrix} \dot{\mathbf{r}}_e^l \\ \dot{\mathbf{v}}_e^l \\ \dot{\mathbf{q}}_b^l \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-1}\mathbf{v}_e^l \\ \mathbf{C}_b^l \mathbf{f}^b - \left(2\boldsymbol{\Omega}_{ie}^l + \boldsymbol{\Omega}_{el}^l\right)\mathbf{v}_e^l + \mathbf{g}^l \\ \frac{1}{2}\mathbf{q}_b^l \otimes \left[\boldsymbol{\omega}_{lb}^b\right]_q \end{bmatrix} \tag{1.5.17}$$

which expresses the first-order navigation equation in the l-frame. Figure 1.6 depicts the l-frame mechanization in block diagram form.

1.6 Dynamic Systems

1.6.1 Continuous-Time Systems Models

The model for a finite-dimensional *linear continuous-time system* with stochastic inputs can be written as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) + \mathbf{L}(t)\mathbf{u}(t) \\ \mathbf{z}(t) &= \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t) \end{aligned} \tag{1.6.1}$$

where $\mathbf{x}(t)$ is the $(p \times 1)$ system *state vector* at time $t \in \mathbb{R}^+$, $\mathbf{z}(t)$ is the $(l \times 1)$ *measurement vector*, $\mathbf{F}(t)$ is the $(p \times p)$ time-varying *dynamic coefficient matrix*, $\mathbf{G}(t)$ is the $(p \times r)$ time-varying *process noise coupling matrix*, $\mathbf{w}(t)$ is a $(r \times 1)$ *random forcing function*, $\mathbf{L}(t)$ is a $(p \times r)$ time-varying *input coupling matrix*, $\mathbf{u}(t)$ is a $(r \times 1)$ *deterministic input vector*, $\mathbf{v}(t)$ is a $(l \times 1)$ measurement noise vector and $\mathbf{H}(t)$ is the $(l \times p)$ time-varying *measurement design matrix* which maps the true state space $\mathbf{x}(t)$ into the observed space. The initial state $\mathbf{x}(0)$ is assumed to be a normal random vector with mean $\boldsymbol{\mu}_0$ and covariance $\mathbf{P}_0$. The mean and covariance of the random variables vectors $\mathbf{w}(t)$ and $\mathbf{v}(t)$ will be denoted

$$\begin{aligned} \mu_{\mathbf{w}}(t) &= \mathbb{E}\left[\mathbf{w}(t)\right] = \mathbf{0} \\ \operatorname{cov}\left(\mathbf{w}(t), \mathbf{w}(\tau)\right) &= \mathbf{Q}(t)\delta(t-\tau) \\ \mu_{\mathbf{v}}(t) &= \mathbb{E}\left[\mathbf{v}(t)\right] = \mathbf{0} \\ \operatorname{cov}\left(\mathbf{v}(t), \mathbf{v}(\tau)\right) &= \mathbf{R}(t)\delta(t-\tau). \end{aligned} \tag{1.6.2}$$

with $\delta(\cdot)$ the *Dirac delta function*. Thus, the parameter set associated with the continuous state space model is

$$\boldsymbol{\theta}_c = \{\mathbf{F}, \mathbf{L}, \mathbf{G}, \mathbf{H}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_0, \boldsymbol{P}_0\}. \tag{1.6.3}$$

1.6.2 Discrete-Time State Space Notation

The equivalent *linear discrete-time system* model for the digital time sequence $\{t_k : k \in \mathbb{Z}^+\}$ is

$$\begin{aligned} \mathbf{x}_k &= \boldsymbol{\Phi}_{k-1}\mathbf{x}_{k-1} + \boldsymbol{\Gamma}_{k-1}\mathbf{w}_{k-1} + \boldsymbol{\Lambda}_{k-1}\mathbf{u}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{1.6.4}$$

where $\boldsymbol{\Phi}_k$, the discrete form of $\mathbf{F}(t)$, will be called the *state transition matrix* and relates $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$. The matrices $\boldsymbol{\Gamma}_k$, $\boldsymbol{\Lambda}_k$ and $\mathbf{H}_k$ are the discrete forms of $\mathbf{G}(t)$, $\mathbf{L}(t)$ and $\mathbf{H}(t)$, respectively. The initial state $\mathbf{x}_0$ is assumed to be a normal random vector with mean vector $\boldsymbol{\mu}_0$ and $(p \times p)$ initial covariance matrix $\mathbf{P}_0$. The mean and covariance of the random variables vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ will be denoted

$$\begin{aligned} \mu_{\mathbf{w}} &= \mathbb{E}[\mathbf{w}_k] = \mathbf{0} \\ \operatorname{cov}(\mathbf{w}_k, \mathbf{w}_l) &= \mathbf{Q}_k\delta(t_k - t_l) \\ \mu_{\mathbf{v}} &= \mathbb{E}[\mathbf{v}_k] = \mathbf{0} \\ \operatorname{cov}(\mathbf{v}_k, \mathbf{v}_l) &= \mathbf{R}_k\delta(t_k - t_l) \end{aligned} \tag{1.6.5}$$

with $l \in \mathbb{Z}^+$. The vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is the set of time-invariant state space model parameters such that

$$\boldsymbol{\theta} = \{\boldsymbol{\Phi}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}, \mathbf{H}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_0, \mathbf{P}_0\}. \tag{1.6.6}$$

When $\mathbf{F}(t) = \mathbf{F}$ is a constant matrix, $\boldsymbol{\Phi}_{k-1}$ can be computed as

$$\boldsymbol{\Phi}_{k-1} = e^{\mathbf{F}\Delta t_k} \tag{1.6.7}$$

where $\Delta t_k = t_k - t_{k-1}$ is the *sampling period*. Usually, $\boldsymbol{\Phi}_{k-1}$ is approximated by retaining the first terms of the $e^{\mathbf{F}\Delta t_k}$ Taylor series expansion

$$\boldsymbol{\Phi}_{k-1} \approx \mathbf{I} + \mathbf{F}\Delta t_k + \frac{(\mathbf{F}\Delta t_k)^2}{2!} + \frac{(\mathbf{F}\Delta t_k)^3}{3!} + \dots \tag{1.6.8}$$

The matrix $\mathbf{Q}_{k-1}$ must account for the integrated effect of $\mathbf{w}(t)$ by the system dynamics over each sampling period. It can be shown (see e.g. Farrell [2008]) that the solution is

$$\mathbf{Q}_{k-1} = \int_{t_{k-1}}^{t_k} \boldsymbol{\Phi}(t_k, s)\mathbf{G}(s)\mathbf{Q}(s)\mathbf{G}^T(s)\boldsymbol{\Phi}^T(t_k, s)ds. \tag{1.6.9}$$

For a small time interval Δt_k, Eq. (1.6.9) can be approximated to

$$\mathbf{Q}_{k-1} \approx \mathbf{G}\mathbf{Q}\mathbf{G}^T\Delta t_k. \tag{1.6.10}$$

1.7 System Error Dynamics

Errors limit the accuracy of the navigation solution provided by an INS. These can be divided into three major groups:

- *Initialization errors*: information on initial conditions must be provided for position and velocity, while that of attitude can be possibly estimated. The imperfections in the initialization are propagated continuously from one estimate to the next;
- *Inertial sensor errors*: accelerometer and gyroscope signals are affected by different types of errors (see Section 1.4) which will cause navigation errors;
- *Computational errors*: approximations and simplifications made in the implementation of the navigation equations and round-off errors ultimately result in navigation errors.

How the first two types of errors affect the navigation solution with time is described by INS error propagation models. These models, developed in the post World War II era, were used first in the context of integration with satellite data, second for low-cost inertial sensors. Two main approaches are used for the derivation of INS error models:

- The *true frame approach* (also known as phi-angle or perturbation approach) perturbs the INS equations in the l-frame which has its origin at the true geographic location of the INS;
- The *computer frame approach* (also known as psi-angle approach) perturbs the INS equations in the c-frame which is the local-level north-pointing coordinate system with its origin at the location computed by the INS.

It has been shown that both approaches are equivalent and provide identical results Benson [1975]. The true frame perturbation approach will be used in this study for deriving the INS error model.

Eq. (1.5.17) can be written as a nonlinear system of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}_1 \left\{ \left[\mathbf{x}_1(t), \mathbf{x}_2(t) \right], t \right\} \tag{1.7.1}$$

where $\mathbf{x}_1(t)$ is the navigation state, and $\mathbf{x}_2(t)$ is the vector containing the input forcing functions, i.e. the inertial measurements $\mathbf{f}^b$ and $\boldsymbol{\omega}_{ib}^b$. Since $\mathbf{x}_1(t)$ is unknown, it must be replaced by its approximation:

$$\hat{\mathbf{x}}_1(t) = \mathbf{x}_1(t) - \delta\mathbf{x}_1(t) \tag{1.7.2}$$

where $\delta\mathbf{x}_1(t)$ are called *error states*. Therefore, Eq. (1.7.1) becomes

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}_1 \left\{ \left[\mathbf{x}_1(t) - \delta\mathbf{x}_1(t), \mathbf{x}_2(t) \right], t \right\}. \tag{1.7.3}$$

Expanding Eq. (1.7.3) in a first order Taylor series yields

$$\delta\dot{\mathbf{x}}_1(t) = \dot{\mathbf{x}}_1(t) - \dot{\hat{\mathbf{x}}}_1(t) = \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} \delta\mathbf{x}_1(t) = \mathbf{F}_1(t)\delta\mathbf{x}_1(t). \tag{1.7.4}$$

where $\mathbf{F}_1(t)$ is the dynamic coefficient matrix. Due to the errors corrupting inertial measurements (see Eq. (1.4.2)), $\delta\dot{\mathbf{x}}_2(t)$ is usually modeled using the following state space model (we will come to that in Section 1.8):

$$\delta\dot{\mathbf{x}}_2(t) = \mathbf{F}_2(t)\delta\mathbf{x}_2(t) + \mathbf{G}_2(t)\mathbf{w}(t) \tag{1.7.5}$$

with $\mathbf{w}(t)$ being Gaussian white noise. Finally, the *augmented state vector* can be written by combining Eq. (1.7.4) and (1.7.5), i.e.

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{1.7.6}$$

where

$$\delta\dot{\mathbf{x}}(t) = \begin{bmatrix} \delta\mathbf{x}_1(t) \\ \delta\mathbf{x}_2(t) \end{bmatrix}, \tag{1.7.7}$$

$$\mathbf{F}(t) = \begin{bmatrix} \mathbf{F}_1(t) & \mathbf{F}_{12}(t) \\ \mathbf{0} & \mathbf{F}_2(t) \end{bmatrix} \text{ and } \mathbf{G}(t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{G}_2(t) \end{bmatrix}. \tag{1.7.8}$$

1.7.1 Attitude Error Model

The derivation of the attitude errors is simplified if using the DCM representation. Let $\boldsymbol{\varepsilon}^l = [\varepsilon_N, \varepsilon_E, \varepsilon_D]^T$ be the vector containing the misalignment errors from navigation l-frame (see Figure 1.4). Then, $\mathbf{C}_b^l$ can be linked to its estimated quantity, denoted as $\hat{\mathbf{C}}_b^l$, through

$$\mathbf{C}_b^l = \left(\mathbf{I} + \mathbf{E}^l\right)\hat{\mathbf{C}}_b^l \tag{1.7.9}$$

where $\mathbf{E}^l = \left[\boldsymbol{\varepsilon}^l\times\right]$. The Euler angles can be estimated by solving

$$\dot{\hat{\mathbf{C}}}_b^l = \hat{\mathbf{C}}_b^l\hat{\boldsymbol{\Omega}}_{lb}^b \tag{1.7.10}$$

with $\hat{\boldsymbol{\Omega}}_{lb}^b = \left[\hat{\boldsymbol{\omega}}_{lb}^b\times\right]$. Two equations can be considered to obtain the rate of change of $\hat{\mathbf{C}}_b^l$. On one hand, differentiating Eq. (1.7.9) yields

$$\begin{aligned} \dot{\hat{\mathbf{C}}}_b^l &= \dot{\mathbf{C}}_b^l - \dot{\mathbf{E}}^l\mathbf{C}_b^l - \mathbf{E}^l\dot{\mathbf{C}}_b^l \\ &= \dot{\mathbf{C}}_b^l - \dot{\mathbf{E}}^l\mathbf{C}_b^l - \mathbf{E}^l\mathbf{C}_b^l\boldsymbol{\Omega}_{lb}^b. \end{aligned} \tag{1.7.11}$$

On the other hand, Eq. (1.7.10) can be rewritten as

$$\begin{aligned} \dot{\hat{\mathbf{C}}}_b^l &= \left(\mathbf{I} - \mathbf{E}^l\right)\mathbf{C}_b^l\left(\boldsymbol{\Omega}_{lb}^b - \delta\boldsymbol{\Omega}_{lb}^b\right) \\ &= \mathbf{C}_b^l\boldsymbol{\Omega}_{lb}^b - \mathbf{C}_b^l\delta\boldsymbol{\Omega}_{lb}^b - \mathbf{E}^l\mathbf{C}_b^l\boldsymbol{\Omega}_{lb}^b + \mathbf{E}^l\mathbf{C}_b^l\delta\boldsymbol{\Omega}_{lb}^b. \end{aligned} \tag{1.7.12}$$

From Eq. (1.7.11) and (1.7.12),

$$\dot{\mathbf{E}}^l \approx \mathbf{C}_b^l\delta\boldsymbol{\Omega}_{lb}^b\mathbf{C}_l^b \tag{1.7.13}$$

where second-order effects have been neglected. The vector form of this equation can be expressed as

$$\dot{\boldsymbol{\varepsilon}}^l = \mathbf{C}_b^l \delta\boldsymbol{\omega}_{lb}^b. \tag{1.7.14}$$

The angular velocity of the b-frame with respect to the l-frame is given by

$$\hat{\boldsymbol{\omega}}_{lb}^b = \hat{\boldsymbol{\omega}}_{ib}^b - \hat{\mathbf{C}}_l^b \hat{\boldsymbol{\omega}}_{il}^l. \tag{1.7.15}$$

Linearizing this equation and assuming that $\hat{\boldsymbol{\omega}}_{pq}^q = \boldsymbol{\omega}_{pq}^q - \delta\boldsymbol{\omega}_{pq}^q$ for any frame p and q gives

$$\delta\boldsymbol{\omega}_{lb}^b = \delta\boldsymbol{\omega}_{ib}^b - \mathbf{C}_l^b \delta\boldsymbol{\omega}_{il}^l + \mathbf{C}_l^b \mathbf{E}^l \boldsymbol{\omega}_{il}^l \tag{1.7.16}$$

where second order effects have been neglected. Substituting Eq (1.7.16) in Eq. (1.7.14) and keeping the vectorial notation yields the final form of the attitude errors time derivative:

$$\dot{\boldsymbol{\varepsilon}}^l = \mathbf{C}_b^l \delta\boldsymbol{\omega}_{ib}^b - \delta\boldsymbol{\omega}_{il}^l - \boldsymbol{\omega}_{il}^l \times \boldsymbol{\varepsilon}^l. \tag{1.7.17}$$

1.7.2 Velocity Error Model

From Eq. (1.5.9), we can write in vectorial form:

$$\begin{aligned} \dot{\hat{\mathbf{v}}}_e^n &= \hat{\mathbf{C}}_b^l \hat{\mathbf{f}}^b - \left(\hat{\boldsymbol{\omega}}_{ie}^l + \hat{\boldsymbol{\omega}}_{il}^l\right) \times \hat{\mathbf{v}}_e^l + \hat{\mathbf{g}}^l \\ &= \left(\mathbf{I} - \mathbf{E}^l\right) \mathbf{C}_b^l \left(\mathbf{f}^b - \delta\mathbf{f}^b\right) - \left(\boldsymbol{\omega}_{ie}^l - \delta\boldsymbol{\omega}_{ie}^l + \boldsymbol{\omega}_{il}^l - \delta\boldsymbol{\omega}_{il}^l\right) \times \left(\mathbf{v}_e^l - \delta\mathbf{v}_e^l\right) \\ &\quad + \mathbf{g}^l - \delta\mathbf{g}^l. \end{aligned} \tag{1.7.18}$$

Assuming that $\delta\mathbf{v}_e^l = \mathbf{v}_e^l - \hat{\mathbf{v}}_e^l$, the expression for the velocity errors time derivative is

$$\delta\dot{\mathbf{v}}_e^l = -\mathbf{f}^l \times \boldsymbol{\varepsilon}^l - \left(\boldsymbol{\omega}_{ie}^l + \boldsymbol{\omega}_{il}^l\right) \times \delta\mathbf{v}_e^l - \left(\delta\boldsymbol{\omega}_{ie}^l + \delta\boldsymbol{\omega}_{il}^l\right) \times \mathbf{v}_e^l + \mathbf{C}_b^l \delta\mathbf{f}^b + \delta\mathbf{g}^l \tag{1.7.19}$$

where $\mathbf{f}^l$ is the specific force vector resolved in the l-frame, $\delta\mathbf{f}^b$ are random specific force errors, and $\delta\mathbf{g}^l$ includes errors in the computed normal gravity due to position errors and the gravity anomaly.

1.7.3 Position Error Model

Eq. (1.5.12) can be rewritten as follows:

$$\mathbf{v}_e^l = \mathbf{D}\dot{\mathbf{r}}_e^l \tag{1.7.20}$$

whose linearized form is

$$\delta\mathbf{v}_e^l = \mathbf{D}\delta\dot{\mathbf{r}}_e^l + \delta\mathbf{D}\dot{\mathbf{r}}_e^l. \tag{1.7.21}$$

The matrix $\delta\mathbf{D}$ contains the effects of position errors. Therefore, the previous relation can be reformulated as

$$\delta\mathbf{v}_e^l = \mathbf{D}\delta\dot{\mathbf{r}}_e^l + \mathbf{D}_r \delta\mathbf{r}_e^l \tag{1.7.22}$$

with $\mathbf{D}_r$ given by:

$$\mathbf{D}_r = \begin{bmatrix} 0 & 0 & \dot{\phi} \\ -\dot{\lambda}\,(R_P + h)\sin(\phi) & 0 & \dot{\lambda}\cos(\phi) \\ 0 & 0 & 0 \end{bmatrix}. \tag{1.7.23}$$

Finally, solving Eq. (1.7.22) for $\delta\dot{\mathbf{r}}_e^l$ yields the final form of the position error model:

$$\delta\dot{\mathbf{r}}_e^l = \mathbf{D}^{-1}\delta\dot{\mathbf{v}}_e^l - \mathbf{D}^{-1}\mathbf{D}_r\delta\mathbf{r}_e^l. \tag{1.7.24}$$

1.7.4 True Frame INS Error Model

The combination of Eq. (1.7.17), (1.7.19) and (1.7.24) yields the first-order differential equations in accordance with Eq. (1.7.6):

$$\begin{bmatrix} \delta\dot{\mathbf{r}}_e^l \\ \delta\dot{\mathbf{v}}_e^l \\ \dot{\boldsymbol{\varepsilon}}^l \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-1}\delta\dot{\mathbf{v}}_e^l - \mathbf{D}^{-1}\mathbf{D}_r\delta\mathbf{r}_e^l \\ -\mathbf{f}^l \times \boldsymbol{\varepsilon}^l - \left(\boldsymbol{\omega}_{ie}^l + \boldsymbol{\omega}_{il}^l\right) \times \delta\mathbf{v}_e^l - \left(\delta\boldsymbol{\omega}_{ie}^l + \delta\boldsymbol{\omega}_{il}^l\right) \times \mathbf{v}_e^l + \mathbf{C}_b^l\delta\mathbf{f}^b + \delta\mathbf{g}^l \\ \mathbf{C}_b^l\delta\boldsymbol{\omega}_{ib}^b - \delta\boldsymbol{\omega}_{il}^l - \boldsymbol{\omega}_{il}^l \times \boldsymbol{\varepsilon}^l \end{bmatrix} \tag{1.7.25}$$

which can be further expanded in state space notation:

$$\delta\dot{\mathbf{x}}^l = \mathbf{F}\delta\mathbf{x}^l + \mathbf{G}\mathbf{w} \tag{1.7.26}$$

where $\delta\dot{\mathbf{x}}^l$ represents the error in the nominal navigation state. Thus,

$$\begin{bmatrix} \delta\dot{\mathbf{r}}_e^l \\ \delta\dot{\mathbf{v}}_e^l \\ \dot{\boldsymbol{\varepsilon}}^l \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{rr} & \mathbf{F}_{rv} & \mathbf{F}_{r\varepsilon} \\ \mathbf{F}_{vr} & \mathbf{F}_{vv} & \mathbf{F}_{v\varepsilon} \\ \mathbf{F}_{\varepsilon r} & \mathbf{F}_{\varepsilon v} & \mathbf{F}_{\varepsilon\varepsilon} \end{bmatrix} \begin{bmatrix} \delta\mathbf{r}_e^l \\ \delta\mathbf{v}_e^l \\ \boldsymbol{\varepsilon}^l \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{C}_b^l & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_b^l \end{bmatrix} \begin{bmatrix} \delta\mathbf{f}^b \\ \delta\boldsymbol{\omega}_{ib}^b \end{bmatrix}. \tag{1.7.27}$$

The expanded error vector is

$$\delta\mathbf{x}^l = \left[\begin{array}{ccc|ccc} \delta\phi & \delta\lambda & \delta h & \delta v_N & \delta v_E & \delta v_D \end{array}\right]^T, \tag{1.7.28}$$

$$\mathbf{w} = \left[\begin{array}{ccc|ccc} \delta f_1 & \delta f_2 & \delta f_3 & \delta\omega_1 & \delta\omega_2 & \delta\omega_3 \end{array}\right]^T. \tag{1.7.29}$$

Each subcomponents of the $\mathbf{F}$ matrix are each matrices in $\mathbb{R}^{3\times3}$:

$$\mathbf{F}_{rr} = \begin{bmatrix} 0 & 0 & \frac{-\dot{\phi}}{R_M+h} \\ \dot{\lambda}\tan(\phi) & 0 & \frac{-\dot{\lambda}}{R_P+h} \\ 0 & 0 & 0 \end{bmatrix} \tag{1.7.30}$$

$$\mathbf{F}_{rv} = \mathbf{D}^{-1} \tag{1.7.31}$$

$$\mathbf{F}_{r\phi} = \mathbf{0}_{3\times3} \tag{1.7.32}$$

$$\mathbf{F}_{vr} = \begin{bmatrix} -\left(2\omega_{ie}\cos(\phi) + \frac{\dot{\lambda}}{\cos(\phi)}\right)v_E & 0 & \frac{\dot{\lambda}\sin(\phi)}{R_P+h}v_E - \frac{\dot{\phi}}{R_M+h}v_D \\ 2\omega_{ie}\left(v_N\cos(\phi) - v_D\sin(\phi)\right) + \frac{\dot{\lambda}}{\cos(\phi)}v_N & 0 & \frac{-\dot{\lambda}}{R_P+h}\left(v_N\sin(\phi) + v_D\cos(\phi)\right) \\ 2\omega_{ie}v_E\sin(\phi) & 0 & \frac{\dot{\lambda}\cos(\phi)}{R_P+h}v_E + \frac{\dot{\phi}}{R_M+h}v_N + \frac{\partial g_0}{\partial h} \end{bmatrix}$$

$$\mathbf{F}_{vv} = \begin{bmatrix} \frac{v_D}{R_M+h} & -2\left(\omega_{ie} + \dot{\lambda}\right)\sin(\phi) & \dot{\phi} \\ \left(2\omega_{ie} + \dot{\lambda}\right)\sin(\phi) & \frac{v_N\tan(\phi)+v_D}{R_P+h} & \left(2\omega_{ie} + \dot{\lambda}\right)\cos(\phi) \\ -2\dot{\phi} & -2\left(\omega_{ie} + \dot{\lambda}\right)\cos(\phi) & 0 \end{bmatrix} \tag{1.7.33}$$

$$\mathbf{F}_{v\phi} = -[\mathbf{f}^l\times] \tag{1.7.34}$$

$$\mathbf{F}_{\phi r} = \begin{bmatrix} \omega_{ie}\sin(\phi) & 0 & \frac{\dot{\lambda}\cos(\phi)}{R_P+h} \\ 0 & 0 & \frac{-\dot{\phi}}{R_M+h} \\ \omega_{ie}\cos(\phi) + \frac{\dot{\lambda}}{\cos(\phi)} & 0 & \frac{-\dot{\lambda}\sin(\phi)}{R_P+h} \end{bmatrix} \tag{1.7.35}$$

$$\mathbf{F}_{\phi v} = \begin{bmatrix} 0 & \frac{-1}{R_P+h} & 0 \\ \frac{1}{R_M+h} & 0 & 0 \\ 0 & \frac{\tan(\phi)}{R_P+h} & 0 \end{bmatrix} \tag{1.7.36}$$

$$\mathbf{F}_{\phi\phi} = \begin{bmatrix} 0 & -\left(\omega_{ie}+\dot{\lambda}\right)\sin(\phi) & \dot{\phi} \\ \left(\omega_{ie}+\dot{\lambda}\right)\sin(\phi) & 0 & \left(\omega_{ie}+\dot{\lambda}\right)\cos(\phi) \\ -\dot{\phi} & -\left(\omega_{ie}+\dot{\lambda}\right)\cos(\phi) & 0 \end{bmatrix} \tag{1.7.37}$$

Three distinct frequencies characterize the navigation errors of this model:

- *Schuler oscillation*: the frequency of this oscillation is given by $\omega_s = \sqrt{g/R_0}$, with g and R_0 being the mean gravity acceleration and the mean distance of the vehicle to the Earth center, respectively. The period corresponds approximately to 84 minutes.
- *Foucault oscillation*: the frequency is given by $\omega_{ie}\sin(\phi)$, which corresponds to a period of about 30 hours for moderate latitudes.
- 24-hours oscillation: this oscillation is directly linked to the Earth's rotation.

These oscillations are important to consider in pure inertial navigation. However, their effect is mitigated by INS/GNSS integration.

1.7.5 Computer Frame INS Error Model

As already mentioned, the INS error model can also be developed with respect to the c-frame. Here, we refer the reader to Kong [2000], Scherzinger [1996] for its derivation and provide the relationships without a proof.

In the c-frame representation, $\mathbf{C}_c^e$, $\boldsymbol{\omega}_{ie}^c$ and $\boldsymbol{\omega}_{ic}^c$ are known without error because the position and velocity of the c-frame is known from the INS computer. Therefore, the following relationships link the true l-frame to the c-frame (see Figure 1.4):

$$\delta\mathbf{v}^l = \delta\mathbf{v}^c - \delta\boldsymbol{\theta}\times\mathbf{v}^c, \tag{1.7.38}$$

$$\delta\mathbf{g}^l = \delta\mathbf{g}^c - \delta\boldsymbol{\theta}\times\mathbf{g}^c, \tag{1.7.39}$$

$$\delta\boldsymbol{\omega}_{ie}^l = -\delta\boldsymbol{\theta}\times\boldsymbol{\omega}_{ie}^c. \tag{1.7.40}$$

The gravity error expressed in the c-frame can be shown to be

$$\delta\mathbf{g}^c \approx \begin{bmatrix} -\omega_s^2\delta r_N \\ -\omega_s^2\delta r_E \\ 2\omega_s^2\delta r_D \end{bmatrix}. \tag{1.7.41}$$

Therefore, the resulting computer frame error model is

$$\begin{bmatrix} \delta\dot{\mathbf{r}}^c \\ \delta\dot{\mathbf{v}}^c \\ \dot{\boldsymbol{\psi}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\omega}_{ec}^c \times \delta\mathbf{r}^c + \delta\mathbf{v}^c \\ \mathbf{f}^c \times \boldsymbol{\psi} - (2\boldsymbol{\omega}_{ie}^c + \boldsymbol{\omega}_{ec}^c) \times \delta\mathbf{v}^c + \delta\mathbf{g}^c + \mathbf{C}_b^p \delta\mathbf{f}^b \\ -(\boldsymbol{\omega}_{ie}^c + \boldsymbol{\omega}_{ec}^c) \times \boldsymbol{\psi} - \mathbf{C}_b^p \delta\boldsymbol{\omega}_{ib}^b \end{bmatrix}. \tag{1.7.42}$$

The differences between the true frame and computer frame approaches are mainly in the implementation aspects Shin [2005]:

- The dynamics of the position and velocity error in Eq. (1.7.42) do not depend on angular rate errors caused by the Earth rate and transport rate mis-resolved by the c-frame misalignment.
- The psi-angle dynamics (third line in Eq. (1.7.42)) is rate-stable, independent of all other INS errors and driven only by the gyroscope error terms $\delta\boldsymbol{\omega}_{ib}^b$ Shin [2005].
- The computer frame error model contains fewer terms and hence is easier to implement in a Kalman filter.

Details on the resulting state space model can be found in Kong [2000], Shin [2005].

1.8 State Space Augmentation

Given the estimates of the calibration factors $\Delta\hat{\mathbf{f}}^b$ and $\Delta\hat{\boldsymbol{\omega}}_{ib}^b$, it can be seen from Eq. (1.7.17) and (1.7.19) that the attitude and velocity errors are driven by the uncalibrated portion of gyroscope and accelerometer errors:

$$\delta\mathbf{f}^b = \Delta\mathbf{f}^b - \Delta\hat{\mathbf{f}}^b, \tag{1.8.1}$$
$$\delta\boldsymbol{\omega}_{ib}^b = \Delta\boldsymbol{\omega}_{ib}^b - \Delta\hat{\boldsymbol{\omega}}_{ib}^b. \tag{1.8.2}$$

In cases where an error source has a significant impact on the navigation error, and it is possible to estimate the parameters pertinent to describing the error sources, state augmentation can be used. Consider that instrument errors can be modeled as

$$\Delta\mathbf{f}^b = \mathbf{F}_{va}\mathbf{x}_a + \mathbf{G}_a\boldsymbol{\nu}_a \tag{1.8.3}$$
$$\Delta\boldsymbol{\omega}_{ib}^b = \mathbf{F}_{\varepsilon g}\mathbf{x}_g + \mathbf{G}_g\boldsymbol{\nu}_g \tag{1.8.4}$$

where $\boldsymbol{\nu}_a$ and $\boldsymbol{\nu}_g$ are Gaussian white noise vectors, and $\mathbf{x}_a$ and $\mathbf{x}_g$ are the vectors containing calibration parameters describing different types of errors (see Section 1.4). By considering Eq. (1.8.1) and (1.8.2), the dynamics of sensor errors can be written as

$$\delta\mathbf{f}^b = \mathbf{F}_{va}\delta\mathbf{x}_a + \mathbf{G}_{va}\boldsymbol{\nu}_a \tag{1.8.5}$$
$$\delta\boldsymbol{\omega}_{ib}^b = \mathbf{F}_{\varepsilon g}\delta\mathbf{x}_g + \mathbf{G}_{\varepsilon g}\boldsymbol{\nu}_g \tag{1.8.6}$$

where $\delta\mathbf{x}_a = \mathbf{x}_a - \hat{\mathbf{x}}_a$ and $\delta\mathbf{x}_g = \mathbf{x}_g - \hat{\mathbf{x}}_g$, with the associated differential equations:

$$\delta\dot{\mathbf{x}}_a = \mathbf{F}_{aa}\delta\mathbf{x}_a + \mathbf{G}_a\boldsymbol{w}_a \tag{1.8.7}$$
$$\delta\dot{\mathbf{x}}_g = \mathbf{F}_{gg}\delta\mathbf{x}_g + \mathbf{G}_g\boldsymbol{w}_g. \tag{1.8.8}$$

With the above definitions, the state augmented version of Eq. (1.7.27) becomes:

$$\begin{bmatrix} \delta\dot{\mathbf{r}}_e^l \\ \delta\dot{\mathbf{v}}_e^l \\ \dot{\boldsymbol{\varepsilon}}^l \\ \delta\dot{\mathbf{x}}_a \\ \delta\dot{\mathbf{x}}_g \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{rr} & \mathbf{F}_{rv} & \mathbf{F}_{r\varepsilon} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{vr} & \mathbf{F}_{vv} & \mathbf{F}_{v\varepsilon} & \mathbf{C}_b^l\mathbf{F}_{va} & \mathbf{0} \\ \mathbf{F}_{\varepsilon r} & \mathbf{F}_{\varepsilon v} & \mathbf{F}_{\varepsilon\varepsilon} & \mathbf{0} & \mathbf{C}_b^l\mathbf{F}_{\varepsilon g} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{aa} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{gg} \end{bmatrix} \begin{bmatrix} \delta\mathbf{r}_e^l \\ \delta\mathbf{v}_e^l \\ \boldsymbol{\varepsilon}^l \\ \delta\mathbf{x}_a \\ \delta\mathbf{x}_g \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_b^l & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_b^l & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_g \end{bmatrix} \begin{bmatrix} -\boldsymbol{\nu}_a \\ \boldsymbol{\nu}_g \\ \boldsymbol{w}_a \\ \boldsymbol{w}_g \end{bmatrix} \tag{1.8.9}$$

The choice of $\mathbf{F}_{aa}$ and $\mathbf{F}_{gg}$ and the associated spectral densities of $\boldsymbol{w}_a$ and $\boldsymbol{w}_g$ (together with their shaping matrices $\mathbf{G}_a$ and $\mathbf{G}_g$) is specific to a sensor error model with the aim to represent an accurate characterization of the error behaviour. Chapter 3 will describe typical stochastic processes (or shaping filters) which can be used in $\delta\mathbf{x}_a$ and $\delta\mathbf{x}_g$ to model gyroscope and accelerometer time-correlated errors.

1.9 Extended Kalman Filter

Generally, a non-controlled standalone INS provides solutions affected by time-dependent, time-growing errors of usually large magnitude (with respect to GNSS positioning). For bounding these drifts, external measurements are used. This combination of multiple sensor types is achieved by an *extended Kalman filter* (EKF).

The nonlinear system and measurement relationship of the filter for discrete times $\{t_k : k \in \mathbb{Z}^+\}$ is

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) + \boldsymbol{\Gamma}_{k-1}\mathbf{w}_{k-1} \tag{1.9.1}$$
$$\mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{v}_k \tag{1.9.2}$$

where $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are known functions, $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero mean Gaussian white noise vectors with covariance matrix $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. $\boldsymbol{\Gamma}_k$ represents the coupling between $\mathbf{x}_k$ and $\mathbf{w}_k$, and $\mathbf{z}_k$ is the measurement vector. At the estimation time t_k, the system is linearized around the previous state estimate. The state and conditional covariance matrix $\mathbf{P}_k^-$ are extrapolated using the transition matrix $\boldsymbol{\Phi}_{k-1}$:

$$\mathbf{x}_k^- = \mathbf{f}\left(\mathbf{x}_{k-1}^+, \mathbf{u}_{k-1}\right) \tag{1.9.3}$$
$$\mathbf{P}_k^- = \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}^+\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{\Gamma}_{k-1}\mathbf{Q}_{k-1}\boldsymbol{\Gamma}_{k-1}^T \tag{1.9.4}$$

where $\mathbf{\Phi}_{k-1} = \frac{\partial \mathbf{f}\left(\mathbf{x}^*_{k-1}, \mathbf{u}_{k-1}\right)}{\partial \mathbf{x}}$ with the approximate $\mathbf{x}^*_{k-1}$ usually chosen to be $\mathbf{f}\left(\mathbf{x}^+_{k-1}, \mathbf{u}_{k-1}\right)$. The "$-$" and "$+$" superscripts respectively denote the predicted and updated variable. The Kalman gain, and the updated state vector and covariance matrix are evaluated as

$$\mathbf{K}_k = \mathbf{P}^-_k \mathbf{H}^T_k \left(\mathbf{H}_k \mathbf{P}^-_k \mathbf{H}^T_k + \mathbf{R}_k\right)^{-1}, \tag{1.9.5}$$

$$\mathbf{P}^+_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}^-_k, \tag{1.9.6}$$

$$\mathbf{x}^+_k = \mathbf{x}^-_k + \mathbf{K}_k \left[\mathbf{z}_k - \mathbf{h}\left(\mathbf{x}^-_k\right)\right], \tag{1.9.7}$$

such that $\mathbf{H}_k = \frac{\partial \mathbf{h}\left(\mathbf{x}^*_k\right)}{\partial \mathbf{x}}$. Note that the difference $\mathbf{z}_k - \mathbf{h}\left(\mathbf{x}^-_k\right)$ is known as the *innovation sequence*, while $\mathbf{z}_k - \mathbf{h}\left(\mathbf{x}^+_k\right)$ is called the *residual sequence*. The initial conditions are

$$\mathbf{x}_0 = \mathbb{E}\left[\mathbf{x}(0)\right], \tag{1.9.8}$$

$$\mathbf{P}_0 = \mathbb{E}\left[\left(\mathbf{x}(0) - \mathbf{x}_0\right)\left(\mathbf{x}(0) - \mathbf{x}_0\right)^T\right], \tag{1.9.9}$$

and it is assumed that $\mathbb{E}\left[\mathbf{w}_k \mathbf{v}^T_j\right] = 0,\ \forall j, k \in \mathbb{Z}^+$. Note that in the case of a continuous system model (see Eq. (1.6.1)), $\mathbf{F}(t) = \mathbf{F} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*}$.

1.10 Optimal Smoothing

In post-processing, the navigation performance can be improved by filtering in forward and backward direction with respect to time. Both solutions can then be combined by optimal smoothing. Two optimal smoothing algorithms are considered herein:

- The *fixed-interval smoother* combines forward and backward filtered solutions in the least-squares sense Shin [2005], Waegli [2009]:

$$\begin{aligned} \mathbf{P}^N_k &= \left[(\mathbf{P}^{-1}_f)_k + (\mathbf{P}^{-1}_b)_k\right]^{-1}, \\ \mathbf{x}^N_k &= (\mathbf{x}_f)_k + \mathbf{P}^N_k (\mathbf{P}^{-1}_b)_k \left[(\mathbf{x}_b)_k - (\mathbf{x}_f)_k\right], \end{aligned} \tag{1.10.1}$$

  where $(\mathbf{P}_\bullet)_k$ and $(\mathbf{x}_\bullet)_k$ are, respectively, the covariance matrix and state vector at digital times $\{t_k : k = 1, \ldots, N\}$ issued from the forward (f index) and backward (b index) filtering, and $\mathbf{P}^N_k$ and $\mathbf{x}^N_k$ are the smoothed covariance matrix and state vector, respectively.

- The *Rauch-Tung-Striebel* (RTS) algorithm Gelb [1974] which handles data processed in a single direction (forward or backward):

$$\mathbf{P}^N_k = \mathbf{P}^+_k + \mathbf{J}_k \cdot \left[\mathbf{P}^N_{k+1} - \mathbf{P}^-_{k+1}\right] \cdot \mathbf{J}^T_k, \tag{1.10.2}$$

$$\mathbf{x}^N_k = \mathbf{x}^+_k + \mathbf{J}_k \cdot \left[\mathbf{x}^N_{k+1} - \mathbf{x}^-_{k+1}\right] \cdot \mathbf{J}^T_k, \tag{1.10.3}$$

  with $\mathbf{J}_k = \mathbf{P}^+_k \cdot \mathbf{\Phi}^T_k \cdot \left(\mathbf{P}^-_{k+1}\right)^{-1}$.

Chapter 2

Navigation Filter Implementation

2.1 Introduction

In this thesis, a loosely coupled EKF was adopted mainly because of its simplicity of implementation and its flexibility. However, the developed concepts can be transfered to other nonlinear filters (e.g. unscented Kalman filter and particle filters), or more sophisticated integration strategies such as closely or tightly coupled schemes when performing INS/GNSS integration.

This chapter discloses the main design details and intricacies of the navigation software developed in the frame of this thesis. The state vector is introduced in Section 2.2, followed by a detailed explanation of the algorithms used in the two fundamental stages of the Kalman filter procedure, namely the *filter prediction* (Section 2.3) and *filter update* (Section 2.4) stages. The initialization and the alignment procedures are explained in Section 2.5 and 2.6.

2.2 State Vector Definition

The complete state vector was defined in Eq. (1.8.9) and is detailed below:

$$\delta\mathbf{x} = \begin{bmatrix} \delta\dot{\mathbf{r}}_e^l \\ \delta\dot{\mathbf{v}}_e^l \\ \dot{\boldsymbol{\varepsilon}}^l \\ \delta\dot{\mathbf{x}}_a \\ \delta\dot{\mathbf{x}}_g \end{bmatrix} \tag{2.2.1}$$

where the navigation error states are

$$\delta\mathbf{r}_e^l = \begin{bmatrix} \delta\phi & \delta\lambda & \delta h \end{bmatrix}^T, \tag{2.2.2}$$

$$\delta\mathbf{v}_e^l = \begin{bmatrix} \delta v_N & \delta v_E & \delta v_D \end{bmatrix}^T, \tag{2.2.3}$$

$$\boldsymbol{\varepsilon}^l = \begin{bmatrix} \varepsilon_N & \varepsilon_E & \varepsilon_D \end{bmatrix}^T. \tag{2.2.4}$$

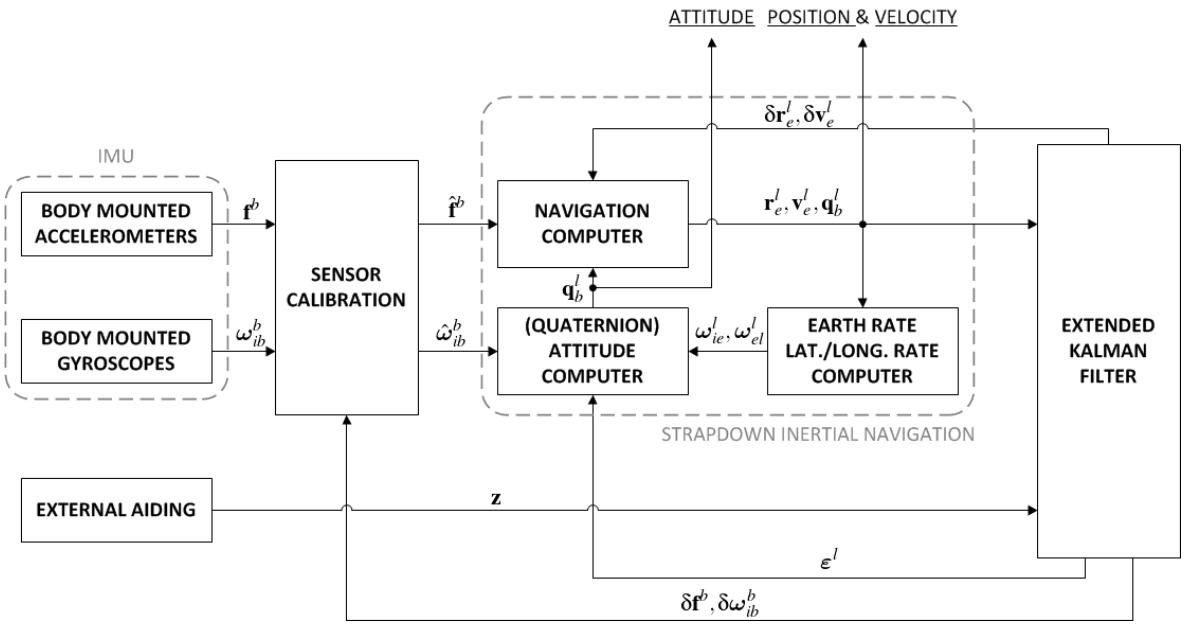

Figure 2.1: Detailed loosely-coupled INS integration approach.

The states $\delta\mathbf{x}_a$ and $\delta\mathbf{x}_g$ account for accelerometer (a index) and gyroscope (g index) biases, respectively. The implemented filter supports flexible introduction of multiple stochastic error models characterizing time-correlated errors in sensors. Therefore,

$$\delta\mathbf{x}_a = \begin{bmatrix} \delta\mathbf{f}_1^b \\ \delta\mathbf{f}_2^b \\ \vdots \\ \delta\mathbf{f}_{J_a}^b \end{bmatrix} \quad \text{and} \quad \delta\mathbf{x}_g = \begin{bmatrix} \delta\boldsymbol{\omega}_{ib,1}^b \\ \delta\boldsymbol{\omega}_{ib,2}^b \\ \vdots \\ \delta\boldsymbol{\omega}_{ib,J_g}^b \end{bmatrix} \tag{2.2.5}$$

where each (3×1) subvector $\delta\mathbf{f}_j^b$ for $j = 1, \ldots, J_a$, and $\delta\boldsymbol{\omega}_{ib,j}^b$ for $j = 1, \ldots, J_g$, represents respectively the accelerometer and gyroscope errors along the $(\mathbf{x}_1^b, \mathbf{x}_2^b, \mathbf{x}_3^b)$ axes, i.e.

$$\delta\mathbf{f}_j^b = \begin{bmatrix} \delta f_{1,j} \\ \delta f_{2,j} \\ \delta f_{3,j} \end{bmatrix} \quad \text{and} \quad \delta\boldsymbol{\omega}_{ib}^b = \begin{bmatrix} \delta\omega_{1,j} \\ \delta\omega_{2,j} \\ \delta\omega_{3,j} \end{bmatrix}. \tag{2.2.6}$$

Note that for simplification, it will be assumed that $J_a = J_g = J$ in the sequel. In addition, for the sake of readibility, only one process per sensor is assumed. Therefore, $J = 1$ and we have

$$\delta\mathbf{x}_a = \delta\mathbf{f}_1^b = \delta\mathbf{f}^b \tag{2.2.7}$$

and

$$\delta\mathbf{x}_g = \delta\boldsymbol{\omega}_{ib,1}^b = \delta\boldsymbol{\omega}_{ib}^b \tag{2.2.8}$$

2.3 Filter Prediction Stage

For implementing a strapdown inertial navigation system, the navigation model given in Eq. (1.5.17) needs to be solved. Since the underlying differential equations do not allow for an

analytical solution, numerical integration algorithms need to be used. The problem can be formulated as:

$$\begin{cases} \dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \text{ for } t \in [t_0, t_f] \\ \mathbf{x}(t_0) & = & \mathbf{x}_0 \end{cases} \tag{2.3.1}$$

where $\mathbf{x} \in \mathbb{R}^p$ is the (navigation) state vector and $\mathbf{u} \in \mathbb{R}^m$ is the (inertial) observation vector. Methods such as the Euler or Runge-Kutta methods are usually employed to compute a numerical solution to this problem Jekeli [2000]. In this thesis, two integration algorithms are considered. They are described in the following sections.

2.3.1 Classical Euler Integration Algorithm

The references Savage [1998a,b] precisely describe the design of an inertial navigation computer. Here, the slightly simplified version of Tomé [2002] is summarized.

Attitude Integration

Eq. (1.5.15) can be further developed as follows:

$$\dot{\mathbf{q}}_b^l = \frac{1}{2}\mathbf{q}_b^l \otimes \left[\boldsymbol{\omega}_{ib}^b\right]_q - \frac{1}{2}\left[\boldsymbol{\omega}_{il}^l\right]_q \otimes \mathbf{q}_b^l. \tag{2.3.2}$$

Consider the digital time sequence $\{t_k : k \in \mathbb{Z}^+\}$. It can be shown that after discretization, the orientation of the i-frame with respect to l-frame at epoch t_{k-1} can be propagated to the next epoch t_k by applying:

$$(\mathbf{q}_b^l)_{k-1|k} = (\mathbf{q}_b^l)_{k-1|k-1} \otimes (\mathbf{q}_b^b)_{k-1|k} \tag{2.3.3}$$

$$(\mathbf{q}_b^l)_{k|k} = (\mathbf{q}_l^l)_{k|k-1} \otimes (\mathbf{q}_b^l)_{k-1|k}. \tag{2.3.4}$$

The quaternion $(\mathbf{q}_b^b)_{k-1|k}$ expresses the rotation of the b-frame between two successive epochs which is derived from the gyroscopes. It can be written as

$$(\mathbf{q}_b^b)_{k-1|k} = \begin{bmatrix} \cos\left(\frac{||\mathbf{u}_k||}{2}\right) \\ \left(\frac{\mathbf{u}_k}{||\mathbf{u}_k||}\right)\sin\left(\frac{||\mathbf{u}_k||}{2}\right) \end{bmatrix} \tag{2.3.5}$$

where $||\mathbf{u}_k||$ is the norm of vector $\mathbf{u}_k = [u_1, u_2, u_3]^T$ that is obtained from the gyroscopes as follows:

$$\mathbf{u}_k = \mathbf{u}(t_k) = \int_{t_{k-1}}^{t_k} \dot{\mathbf{u}}(t)dt \approx (\boldsymbol{\omega}_{ib}^b)_k \cdot \Delta t_k \tag{2.3.6}$$

assuming no coning motion and a constant angular velocity, $(\boldsymbol{\omega}_{ib}^b)_k$, within the sampling period $\Delta t_k = t_k - t_{k-1}$. The quaternion $(\mathbf{q}_l^l)_{k-1|k}$ accounts for the angular rate of the l-frame relative to the nonrotating i-frame, $(\boldsymbol{\omega}_{il}^l)_k$, and can be expressed as a function of the rotation vector $\mathbf{v}_k$:

$$(\mathbf{q}_l^l)_{k|k-1} = \begin{bmatrix} \cos\left(\frac{||\mathbf{v}_k||}{2}\right) \\ -\left(\frac{\mathbf{v}_k}{||\mathbf{v}_k||}\right)\sin\left(\frac{||\mathbf{v}_k||}{2}\right) \end{bmatrix}. \tag{2.3.7}$$

The negative sign on $\mathbf{v}_k$ accounts for the opposite phase sense of $(\mathbf{q}_l^l)_{k|k-1}$, which describes the relative motion of the l-frame at instant t_{k-1} with respect to instant t_k. The vector $\mathbf{v}_k$ can be calculated as

$$\mathbf{v}_k = \mathbf{v}(t_k) \approx \int_{t_{k-1}}^{t_k} \boldsymbol{\omega}_{il}^l(t)dt \approx (\boldsymbol{\omega}_{il}^l)_k \cdot \Delta t_k \tag{2.3.8}$$

which is valid if $(\boldsymbol{\omega}_{il}^l)_k$ is considered as small and slowly changing over a typical t_{k-1} to t_k update cycle.

The $\mathbf{q}_b^l$, $\mathbf{q}_b^b$ and $\mathbf{q}_l^l$ quaternions are propagated at the same rate in this algorithm. However, Savage [1998b] provides a more general algorithm in which $\mathbf{q}_b^l$ is not necessarily propagated with the same rate as $\boldsymbol{\omega}_{ib}^b$ or $\boldsymbol{\omega}_{il}^l$.

Velocity Integration

The digital velocity integration algorithm is direcly formulated from Eq. (1.5.9) as

$$(\mathbf{v}_e^l)_k = (\mathbf{v}_e^l)_{k-1} + (\Delta\mathbf{v}_\mathbf{f}^l)_k + (\Delta\mathbf{v}_{\mathbf{g}/c}^l)_k \tag{2.3.9}$$

where

$$(\Delta\mathbf{v}_\mathbf{f}^l)_k = \int_{t_{k-1}}^{t_k} \mathbf{C}_b^l \mathbf{f}^b dt \approx (\mathbf{C}_b^l)_{k|k-1} \cdot (\mathbf{f}^b)_k \cdot \Delta t_k \tag{2.3.10}$$

is the specific force increment during the period $\Delta t_k = t_k - t_{k-1}$, and

$$\begin{aligned}(\Delta\mathbf{v}_{\mathbf{g}/c}^l)_k &= \int_{t_{k-1}}^{t_k} \left[\mathbf{g}^l - \left(2\boldsymbol{\omega}_{ie}^l + \boldsymbol{\omega}_{el}^l\right) \times \mathbf{v}_e^l\right] dt \\ &\approx \left[(\mathbf{g}^l)_k - \left(2(\boldsymbol{\omega}_{ie}^l)_k + (\boldsymbol{\omega}_{el}^l)_k\right) \times (\mathbf{v}_e^l)_k\right] \cdot \Delta t_k\end{aligned} \tag{2.3.11}$$

is the gravity/Coriolis velocity increment. Note that $(\mathbf{C}_b^l)_{k|k-1}$ is the cosine matrix equivalent of $(\mathbf{q}_b^l)_{k|k-1}$ which can be deduced from Eq. (2.3.4) as follows:

$$(\mathbf{q}_b^l)_{k|k-1} = (\mathbf{q}_l^l)_{k|k-1} \otimes (\mathbf{q}_b^l)_{k-1|k-1}. \tag{2.3.12}$$

Position Integration

The digital integration algorithm for propagating $\mathbf{r}_e^l$ is given by

$$(\mathbf{r}_e^l)_k = (\mathbf{r}_e^l)_{k-1} + \frac{1}{2} \cdot (\mathbf{D}^{-1})_{k-\frac{1}{2}} \left[(\mathbf{v}_e^l)_k + (\mathbf{v}_e^l)_{k-1}\right] \cdot \Delta t_k \tag{2.3.13}$$

where $(\mathbf{D}^{-1})_{k-\frac{1}{2}}$ (see Eq. (1.5.13)) can be calculated using a simple linear extrapolation:

$$(\mathbf{D}^{-1})_{k-\frac{1}{2}} \approx (\mathbf{D}^{-1})_{k-1} + \frac{1}{2}\left[(\mathbf{D}^{-1})_{k-1} - (\mathbf{D}^{-1})_{k-2}\right]. \tag{2.3.14}$$

2.3.2 Prediction-Correction Integration Algorithm

In the classical integration scheme, the filter predicts the state from one filter time t_{k-1} to the next time t_k by integrating the navigation states. Such type of computation in which the estimation of $\mathbf{x}_k$ depends only on one state $\mathbf{x}_{k-1}$ but with a same time interval between them belongs to the class of *one-step* integrators. If an external aiding observation is available at time t_k, an optimal state is computed by blending the predicted state and the observation. In practice however, the sampling of the inertial sensors and the aiding sensors may not necessarily occur at the same time. Also, the IMU sampling frequency may not be perfectly constant over time, or even worse, some data gaps may occur. This makes the integration step size, noted as $\Delta t_k = t_k - t_{k-1}$, not constant and forbids the application of classical multistep integrators like the Runge-Kutta algorithm. This scenario was handled by Rosales and Colomina [2005] who proposed to use *multistep* and *predictor-corrector* methods for integrating the navigation equations. For the sake of completeness, the fundamental idea behind these two methods is given in the sequel. A detailed derivation can be found in Rosales and Colomina [2005].

Consider a continuous function $\mathbf{f}(\cdot)$. Then it is shown in Rosales and Colomina [2005] that for $r \geq 2$,

$$\hat{\mathbf{x}}_{k+r} = \hat{\mathbf{x}}_{k+r-1} + \Delta t \sum_{j=0}^{r} \beta_j \mathbf{f}\left(\hat{\mathbf{x}}_{k+j}, \mathbf{u}_{k+j}, t_{k+j}\right) \tag{2.3.15}$$

where $\{\beta_j : j = 0, 1, \ldots, r\}$ are coefficients that can be obtained by solving the integral

$$\beta_j = \int_{s=0}^{1} \prod_{i=0, i\neq j}^{r} \frac{s - (r - j - 1)}{i - j} ds. \tag{2.3.16}$$

By setting $\beta_r \neq 0$ and $\beta_0 = 0$ in Eq. (2.3.15), one gets the *Adams-Moulton* methods. If $\beta_r = 0$ and $\beta_0 \neq 0$, the *Adams-Bashforth* methods are obtained. Eq. (2.3.15) shows that the case of the Adams-Moulton method is an implicit problem which can be solved by means of the following iterative scheme:

$$\hat{\mathbf{x}}_{k+r}^{(m+1)} = \hat{\mathbf{x}}_{k+r-1} + \Delta t \left\{ \beta_r \mathbf{f}\left(\hat{\mathbf{x}}_{k+r}^{(m)}, \mathbf{u}_{k+r}, t_{k+r}\right) + \sum_{j=1}^{r-1} \beta_j \mathbf{f}\left(\hat{\mathbf{x}}_{k+j}^{(m)}, \mathbf{u}_{k+j}, t_{k+j}\right) \right\} \tag{2.3.17}$$

where $m = 0, 1, 2, \ldots$ is the iteration index. The initial state value $\hat{\mathbf{x}}_{k+r}^{(0)}$ can first be computed using the Adams-Bashforth formula (prediction step), and then $\hat{\mathbf{x}}_{k+r}$ is corrected using the Adams-Moulton iterative scheme (correction step).

In Rosales and Colomina [2005] it is shown how to construct a *variable step-size multistep predictor-corrector* in which the values of the coefficients $\{\beta_j\}$ account for a non-constant sampling period $\Delta t_{k+1} = t_{k+1} - t_k$. In summary the variable step-size multistep predictor and corrector can be written as

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \Delta t_k \sum_{j=1}^{r} (\beta_{r-j})_k \mathbf{f}(\hat{\mathbf{x}}_{k-j}, \mathbf{u}_{k-j}, t_{k-j}) \tag{2.3.18}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \Delta t_k \sum_{j=0}^{r-1} (\beta_{r-j})_k \mathbf{f}(\hat{\mathbf{x}}_{k-j}, \mathbf{u}_{k-j}, t_{k-j}), \tag{2.3.19}$$

respectively.

For example for the particular case of $r = 4$ which was treated by Rosales and Colomina [2005], the variable step-size coefficients for the predictor are:

$$\begin{aligned}
(\beta_0)_k &= \frac{\zeta_k}{12\Delta t_{k-1}\left(\Delta t_{k-1}+\Delta t_{k-2}\right)\left(\Delta t_{k-1}+\Delta t_{k-2}+\Delta t_{k-3}\right)} \\
(\beta_1)_k &= -\frac{\xi_k}{12\Delta t_{k-1}\Delta t_{k-2}\left(\Delta t_{k-2}+\Delta t_{k-3}\right)}\Delta t_k \\
(\beta_2)_k &= \frac{3\Delta t_k^2+4\Delta t_k\left(2\Delta t_{k-1}+\Delta t_{k-2}+\Delta t_{k-3}\right)+6\Delta t_{k-1}\left(\Delta t_{k-1}+\Delta t_{k-2}+\Delta t_{k-3}\right)}{12\Delta t_{k-2}\Delta t_{k-3}\left(\Delta t_{k-1}+\Delta t_{k-2}\right)}\Delta t_k \\
(\beta_3)_k &= -\frac{3\Delta t_k^2+4\Delta t_k\left(2\Delta t_{k-1}+\Delta t_{k-2}\right)+6\Delta t_{k-1}\left(\Delta t_{k-1}+\Delta t_{k-2}\right)}{12\Delta t_{k-3}\left(\Delta t_{k-2}+\Delta t_{k-3}\right)\left(\Delta t_{k-1}+\Delta t_{k-2}+\Delta t_{k-3}\right)}\Delta t_k
\end{aligned}$$

with

$$\begin{aligned}
\zeta_k &= 3\Delta t_k^3+4\Delta t_k^2\left(3\Delta t_{k-1}+2\Delta t_{k-2}+\Delta t_{k-3}\right) \\
&\quad +6\Delta t_k\left[\Delta t_{k-1}\left(3\Delta t_{k-1}+4\Delta t_{k-2}+2\Delta t_{k-3}\right)+\Delta t_{k-2}\left(\Delta t_{k-2}+\Delta t_{k-3}\right)\right] \\
&\quad +12\Delta t_{k-1}\left[\Delta t_{k-1}\left(\Delta t_{k-1}+2\Delta t_{k-2}+\Delta t_{k-3}\right)+\Delta t_{k-2}\left(\Delta t_{k-2}+\Delta t_{k-3}\right)\right] \\
\xi_k &= 3\Delta t_k^2+4\Delta t_k\left(2\Delta t_{k-1}+2\Delta t_{k-2}+\Delta t_{k-3}\right) \\
&\quad +6\Delta t_{k-1}\left(\Delta t_{k-1}+2\Delta t_{k-2}+\Delta t_{k-3}\right)+6\Delta t_{k-2}\left(\Delta t_{k-2}+\Delta t_{k-3}\right)
\end{aligned}$$

and for the corrector:

$$\begin{aligned}
(\beta_1)_k &= \frac{3\Delta t_k^2+4\Delta t_k\left(2\Delta t_{k-1}+\Delta t_{k-2}\right)+6\Delta t_{k-1}\left(\Delta t_{k-1}+\Delta t_{k-2}\right)}{12\left(\Delta t_k+\Delta t_{k-1}\right)\left(\Delta t_k+\Delta t_{k-1}+\Delta t_{k-2}\right)} \\
(\beta_2)_k &= \frac{\Delta t_k^2+2\Delta t_k\left(2\Delta t_{k-1}+\Delta t_{k-2}\right)+6\Delta t_{k-1}\left(\Delta t_{k-1}+\Delta t_{k-2}\right)}{12\Delta t_{k-1}\left(\Delta t_{k-1}+\Delta t_{k-2}\right)} \\
(\beta_3)_k &= \frac{3\Delta t_k-2\left(2\Delta t_k+\Delta t_{k-1}+\Delta t_{k-2}\right)}{12\Delta t_{k-1}\Delta t_{k-2}\left(\Delta t_k+\Delta t_{k-1}\right)}\Delta t_k^2 \\
(\beta_4)_k &= \frac{\Delta t_k+2\Delta t_{k-1}}{12\Delta t_{k-2}\left(\Delta t_{k-1}+\Delta t_{k-2}\right)\left(\Delta t_k+\Delta t_{k-1}+\Delta t_{k-2}\right)}\Delta t_k^2.
\end{aligned}$$

Now that all the elements are defined, we can construct the variable step-size multistep predictor-corrector navigation computer.

Attitude Integration

We define the digital attitude field as

$$\mathbf{f}(\mathbf{x}_k,\mathbf{u}_k,t_k)=\frac{1}{2}(\mathbf{q}_b^l)_k\otimes\left[(\boldsymbol{\omega}_{lb}^b)_k\right]_q \tag{2.3.20}$$

where $(\mathbf{q}_b^l)_k=(\mathbf{q}_b^l)_{k|k-1}=\mathbf{q}_b^l(t_k)$. The digital quaternion $(\mathbf{q}_b^l)_k$ can be predicted from the previous epoch t_{k-1} using the Adams-Bashforth method:

$$(\mathbf{q}_b^l)_k^- = (\mathbf{q}_b^l)_{k-1}^+ + \Delta t_k\sum_{j=1}^{r}\frac{1}{2}(\beta_{r-j})_{k-j}(\mathbf{q}_b^l)_{k-j}^+\otimes\left[(\boldsymbol{\omega}_{lb}^b)_{k-j}^+\right]_q \tag{2.3.21}$$

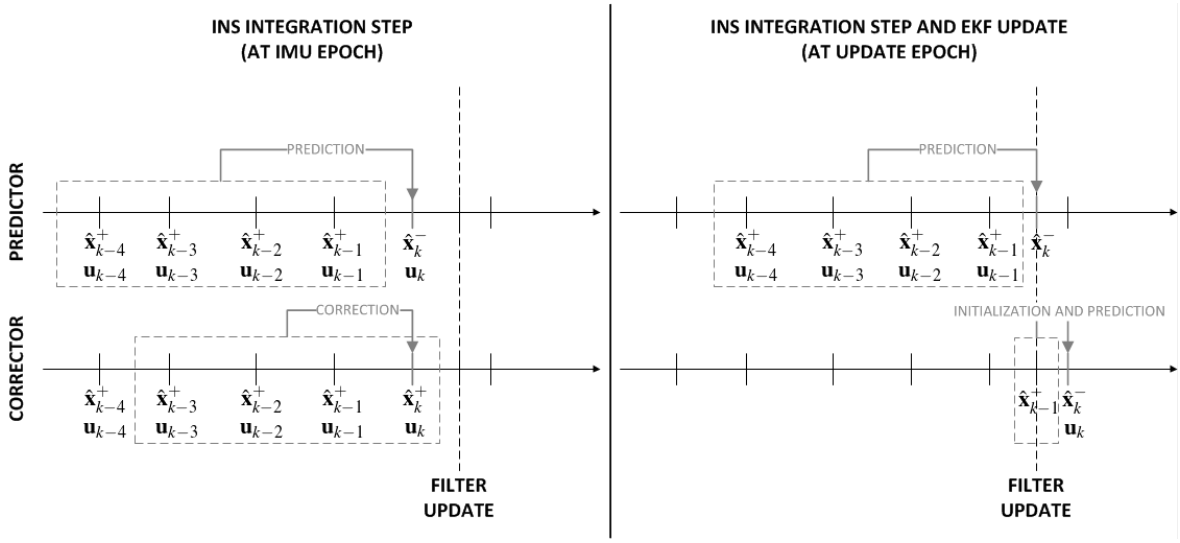

Figure 2.2: Principle of the multistep variable step-size prediction-correction integration algorithm. The left part shows the integration step when no update has to be performed (i.e. IMU observations are available at all epochs but possibly at different time-intervals). The right part shows the situation where an update has to be done (the field has to be interpolated at the epoch with missing IMU observation).

where

$$(\boldsymbol{\omega}_{lb}^{b})_k = (\boldsymbol{\omega}_{ib}^{b})_k - (\mathbf{C}_{l}^{b})_k(\boldsymbol{\omega}_{il}^{l})_k \tag{2.3.22}$$

and

$$(\boldsymbol{\omega}_{il}^{l})_k = (\boldsymbol{\omega}_{ie}^{l})_k + (\boldsymbol{\omega}_{el}^{l})_k. \tag{2.3.23}$$

The cosine matrix $(\mathbf{C}_{l}^{b})_k$ can be retrieved from $(\mathbf{q}_{b}^{l})_k$. Then, the correction step can be performed by the Adams-Moulton formula:

$$(\mathbf{q}_{b}^{l})_k^{+} = (\mathbf{q}_{b}^{l})_{k-1}^{+} + \Delta t_k \left\{ \frac{1}{2}(\beta_r)_k (\mathbf{q}_{b}^{l})_k^{-} \otimes \left[(\boldsymbol{\omega}_{lb}^{b})_k^{-} \right]_q + \sum_{j=1}^{r-1} \frac{1}{2}(\beta_{r-j})_{k-j} (\mathbf{q}_{b}^{l})_{k-j}^{+} \otimes \left[(\boldsymbol{\omega}_{lb}^{b})_{k-j}^{+} \right]_q \right\} \tag{2.3.24}$$

From Eq. (1.5.10), it can be seen that $(\boldsymbol{\omega}_{il}^{l})_k = \mathbf{f}\left((\mathbf{r}_{e}^{l})_k, (\mathbf{v}_{e}^{l})_k\right)$. Hence, $(\boldsymbol{\omega}_{lb}^{b})_k^{-}$ can be computed using $(\mathbf{r}_{e}^{l})_k^{-}$, $(\mathbf{v}_{e}^{l})_k^{-}$ and $(\mathbf{q}_{b}^{l})_k^{-}$ from the predicted state.

Velocity Integration

We define the digital velocity field as

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, t_k) = (\mathbf{C}_{b}^{l})_k (\mathbf{f}^{b})_k - \left[2(\boldsymbol{\omega}_{ie}^{l})_k + (\boldsymbol{\omega}_{el}^{l})_k \right] \times (\mathbf{v}_{e}^{l})_k + (\mathbf{g}^{l})_k \tag{2.3.25}$$

where $(\mathbf{v}_e^l)_k = \mathbf{v}_e^l(t_k)$, $(\mathbf{g}^l)_k = \mathbf{g}^l(t_k)$ and $(\mathbf{C}_b^l)_k = (\mathbf{C}_b^l)_{k|k-1}$. The digital velocity vector $(\mathbf{v}_e^l)_k$ can be predicted from the previous epoch t_{k-1} using the Adams-Bashforth method:

$$(\mathbf{v}_e^l)_k^- = (\mathbf{v}_e^l)_{k-1}^+ + \Delta t_k \sum_{j=1}^{r} (\beta_{r-j})_{k-j} (\mathbf{f}^l)_{k-j}^+ + \sum_{j=1}^{r} (\beta_{r-j})_{k-j} (\Delta \mathbf{v}_{\mathbf{g}/c}^l)_{k-j}^+ \tag{2.3.26}$$

where $(\Delta \mathbf{v}_{\mathbf{g}/c}^l)_k$ can be obtained from Eq. (2.3.11), and $(\mathbf{f}^l)_k = (\mathbf{C}_b^l)_k \mathbf{f}_k^b$. Then, the correction step can be computed using the Adams-Moulton formula:

$$(\mathbf{v}_e^l)_k^+ = (\mathbf{v}_e^l)_{k-1}^+ + \Delta t_k \left\{ (\beta_r)_k \left[(\mathbf{f}^l)_k^- + (\Delta \mathbf{v}_{\mathbf{g}/c}^l)_k^- \right] + \sum_{j=1}^{r-1} (\beta_{r-j})_{k-j} (\mathbf{f}^l)_{k-j}^+ + \sum_{j=1}^{r-1} (\beta_{r-j})_{k-j} (\Delta \mathbf{v}_{\mathbf{g}/c}^l)_{k-j}^+ \right\} \tag{2.3.27}$$

Again, $(\mathbf{r}_e^l)_k^-$, $(\mathbf{v}_e^l)_k^-$ and $(\mathbf{q}_b^l)_k^-$ can be used to compute $(\mathbf{f}^l)_k^-$ and $(\Delta \mathbf{v}_{\mathbf{g}/c}^l)_k^-$.

Position Integration

We define the digital position field as

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, t_k) = (\mathbf{D}^{-1})_k (\mathbf{v}_e^l)_k \tag{2.3.28}$$

such that $(\mathbf{r}_e^l)_k = \mathbf{r}_e^l(t_k)$. The digital position vector $(\mathbf{r}_e^l)_k$ can be predicted from the previous epoch t_{k-1} using the Adams-Bashforth method:

$$(\mathbf{r}_e^l)_k^- = (\mathbf{r}_e^l)_{k-1}^+ + \Delta t_k \sum_{j=1}^{r} (\beta_{r-j})_{k-j} (\mathbf{D}^{-1})_{k-j}^+ (\mathbf{v}_e^l)_{k-j}^+ \tag{2.3.29}$$

where $(\mathbf{D}^{-1})_k$ can be computed using $(\mathbf{r}_e^l)_k$ and $(\mathbf{v}_e^l)_k$ in Eq. (1.5.13). Then, the correction step can be obtained using the Adams-Moulton formula:

$$(\mathbf{r}_e^l)_k^+ = (\mathbf{r}_e^l)_{k-1}^+ + \Delta t_k \left\{ (\beta_r)_k (\mathbf{D}^{-1})_k^- (\mathbf{v}_e^l)_k^- + \sum_{j=1}^{r-1} (\beta_{r-j})_{k-j} (\mathbf{D}^{-1})_{k-j}^+ (\mathbf{v}_e^l)_{k-j}^+ \right\}. \tag{2.3.30}$$

Filter Update

Assume that an external observation is available at digital time t_k. The navigation state is predicted from epoch t_{k-1} to t_k using Eq. (2.3.21), (2.3.25) and (2.3.29). At this stage, two cases may happen:

1. The inertial observations at time t_k, i.e. $\mathbf{f}_k^b$ and $(\boldsymbol{\omega}_{ib}^b)_k$ exist. Then the correction step is performed by applying Eq. (2.3.24), (2.3.27) and (2.3.30), the field is computed for epoch t_k, and the filter is updated. After the update, the fields $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, t_k)$ for attitude, velocity and position are recomputed using the updated state vector before pursuing with the integration (see left part of Figure 2.2).

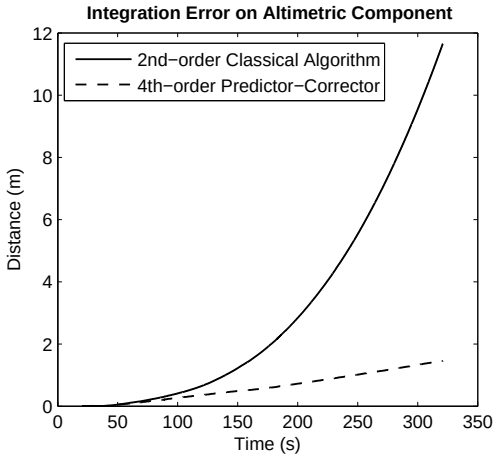

Figure 2.3: Integration error comparison between the classical and the Adams-Bashforth integration algorithm.

2. There are no inertial observations at time t_k, i.e. $\mathbf{f}_k^b$ and $(\boldsymbol{\omega}_{ib}^b)_k$ are missing. In this case, $\mathbf{x}_k^-$ can be computed but not $\mathbf{f}(\mathbf{x}_k^-, \mathbf{u}_k, t_k)$. This renders any further integration step impossible. A solution to this problem is to estimate the field at update epoch by rearranging Eq. (2.3.19) like

$$\mathbf{f}(\mathbf{x}_k^-, t_k) = \frac{1}{(\beta_r)_k}\left(\frac{\mathbf{x}_k^- - \mathbf{x}_{k-1}^+}{\Delta t_k} - \sum_{j=1}^{r-1}(\beta_{r-j})_k \mathbf{f}(\mathbf{x}_{k-j}^+, \mathbf{u}_{k-j}, t_{k-j})\right). \tag{2.3.31}$$

   By this way, the filter can pursuie by integrating the state from t_k to t_{k+1} at the next filter step (see right part of Figure 2.2).

Note that the order of the predictor-corrector is reset to one after each update. This is to avoid computing a prediction/correction with large discontinuities due to state corrections in the previous field.

Finally, the advantage of using higher-order integrators is depicted in the right panel of Figure 2.3. The full line represents the integration error occuring when processing noise-free simulated inertial observations (here Δt_k was simulated as a constant) with the classical integrator while the dashed line is the error occuring when using the fourth-order variable step-size multistep prediction-correction algorithm.

2.4 Filter Update Stage

When an external measurement arrives, the difference $\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-)$ is performed, the Kalman gain $\mathbf{K}_k$ is computed using $\mathbf{P}_k^-$, $\mathbf{H}_k$ and $\mathbf{R}_k$, and the state vector is updated using Eq. (1.9.7).

In the sequel, the measurement models $\mathbf{h}(\cdot)$ together with the $\mathbf{H}_k$ and $\mathbf{R}_k$ matrices are derived for different kinds of external measurements.

2.4.1 Absolute Position Update

In a loosely coupled scheme, GNSS receivers provide absolute curvilinar position measurements, denoted as $\mathbf{z_r} = [z_\phi, z_\lambda, z_h]^T$, which can be used to update the filter. Let $\mathbf{a}^b$ be the lever-arm vector of the GNSS antenna location with respect to the origin of the l-frame expressed in the b-frame. Then,

$$\mathbf{a}^l = \mathbf{C}_b^l \mathbf{a}^b = \begin{bmatrix} a_N & a_E & a_D \end{bmatrix}^T. \tag{2.4.1}$$

The measurement model can be written as

$$\mathbf{z_r} = \mathbf{r}_e^l + \mathbf{D}^{-1}\mathbf{C}_b^l \mathbf{a}^b + \mathbf{v_r} \tag{2.4.2}$$

where $\mathbf{v_r}$ is the measurement noise such that $\mathbf{v_r} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_r})$. Linearizing Eq. (2.4.1) yields

$$\mathbf{H_r} = \begin{bmatrix} \mathbf{I}_{3\times3} + \mathbf{D}_a \mid \mathbf{0}_{3\times3} \mid -\mathbf{D}^{-1}[\mathbf{a}^l\times] \mid \mathbf{0}_{3\times6} \end{bmatrix} \tag{2.4.3}$$

where the auxiliary matrix $\mathbf{D}_a$ is given by

$$\mathbf{D}_a = \begin{bmatrix} 0 & 0 & 0\frac{-a_N}{(R_M+h)^2} \\ \frac{a_E \tan(\phi)}{(R_P+h)\cos(\phi)} & 0 & \frac{-a_E}{(R_P+h)^2\cos(\phi)} \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.4.4}$$

Assuming that $\boldsymbol{\sigma_r} = [\sigma_{r_N}, \sigma_{r_E}, \sigma_{r_D}]^T$ is the position uncertainty expressed in Cartesian coordinates (NED), the measurement noise matrix can be written as

$$\mathbf{R_r} = \begin{bmatrix} \left(\frac{\sigma_{r_N}}{R_M+h}\right)^2 & 0 & 0 \\ 0 & \left(\frac{\sigma_{r_E}}{(R_P+h)\cos(\phi)}\right)^2 & 0 \\ 0 & 0 & \sigma_{r_D}^2 \end{bmatrix}. \tag{2.4.5}$$

2.4.2 Absolute Velocity Update

Considering that velocity measurements expressed in l-frame, denoted as $\mathbf{z_v} = [z_{v_N}, z_{v_E}, z_{v_D}]^T$, are available from a GNSS receiver, the associated measurement model is defined as

$$\mathbf{z_v} = \mathbf{v}_e^l + \boldsymbol{\omega}_{lb}^l \times \left(\mathbf{C}_b^l \mathbf{a}^b\right) + \mathbf{v_v} \tag{2.4.6}$$

where $\mathbf{v_v}$ is the measurement noise such that $\mathbf{v_v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_v})$, and

$$\boldsymbol{\omega}_{lb}^l = \mathbf{C}_b^l \boldsymbol{\omega}_{ib}^b - \boldsymbol{\omega}_{il}^l. \tag{2.4.7}$$

Linearizing Eq. (2.4.6) yields

$$\mathbf{H_v} = \begin{bmatrix} [\mathbf{a}^l\times]\mathbf{D}_1 \mid \mathbf{I}_{3\times3} + [\mathbf{a}^l\times]\mathbf{D}_2 \mid [\mathbf{a}^l\times]\left(\boldsymbol{\Omega}_{il}^l + 2\boldsymbol{\Omega}_{lb}^b\right) \mid \mathbf{0}_{3\times3} \mid -[\mathbf{a}^l\times]\mathbf{C}_b^l \end{bmatrix} \tag{2.4.8}$$

where $\mathbf{\Omega}_{il}^{l} = [\boldsymbol{\omega}_{il}^{l}\times]$, $\mathbf{\Omega}_{lb}^{l} = [\boldsymbol{\omega}_{lb}^{l}\times]$. The auxiliary matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ are given by

$$\mathbf{D}_1 = \begin{bmatrix} -\omega_{ie}\sin(\phi) & 0 & \frac{-\dot{\lambda}\cos(\phi)}{R_P+h} \\ 0 & 0 & \frac{\dot{\phi}}{R_M+h} \\ -\left(\omega_{ie}\cos(\phi) + \frac{\dot{\lambda}}{\cos(\phi)}\right) & 0 & \frac{\dot{\lambda}\sin(\phi)}{R_P+h} \end{bmatrix}, \tag{2.4.9}$$

$$\mathbf{D}_2 = \begin{bmatrix} 0 & \frac{1}{R_P+h} & 0 \\ \frac{-1}{R_M+h} & 0 & 0 \\ 0 & \frac{-\tan(\phi)}{R_P+h} & 0 \end{bmatrix}. \tag{2.4.10}$$

The measurement covariance matrix is given by

$$\mathbf{R}_{\mathbf{v}} = \begin{bmatrix} \sigma_{v_N}^2 & 0 & 0 \\ 0 & \sigma_{v_E}^2 & 0 \\ 0 & 0 & \sigma_{v_D}^2 \end{bmatrix} \tag{2.4.11}$$

such that $\boldsymbol{\sigma_v} = [\sigma_{v_N}, \sigma_{v_E}, \sigma_{v_D}]^T$ are the standard deviations associated to $\mathbf{z}_{\mathbf{v}}$.

2.4.3 Update using Non-holonomic Constraints

If we assume that the vehicle does not jump off the ground or slide lateraly, its velocity towards the $\mathbf{x}_2^b$-axis and the $\mathbf{x}_3^b$-axis is almost zero when $\mathbf{x}_1^b$ is aligned with the forward motion. In this respect, two pseudomeasurements can be built for updating the filter:

$$\mathbf{z}_{\mathbf{h}} = \begin{bmatrix} v_2^b \\ v_3^b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.4.12}$$

The b-frame velocity, $\mathbf{v}^b$, can be estimated using

$$\hat{\mathbf{v}}^b = \mathbf{h}(\hat{\mathbf{x}}) = \left(\hat{\mathbf{C}}_b^l\right)^T \hat{\mathbf{v}}_e^l. \tag{2.4.13}$$

It can be shown Shin and El-Sheimy [2001] that perturbating Eq. (2.4.13) yields, to the first order,

$$\begin{bmatrix} z_{v_1} \\ \mathbf{z}_{\mathbf{h}} \end{bmatrix} - \mathbf{h}(\hat{\mathbf{x}}) \approx \mathbf{C}_l^b \delta\mathbf{v}_e^l - \mathbf{C}_l^b[\mathbf{v}_e^l\times]\boldsymbol{\varepsilon}^l + \mathbf{v}_{\mathbf{h}} \tag{2.4.14}$$

with $\mathbf{v}_{\mathbf{h}} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathbf{h}})$. Taking the second and third row of the above equation gives the linearized model

$$\mathbf{H}_{\mathbf{h}} = \left[\begin{array}{c|ccc|ccc|c} \mathbf{0}_{1\times3} & c_{12} & c_{22} & c_{32} & -v_D c_{22} + v_E c_{32} & v_D c_{12} - v_N c_{32} & -v_E c_{12} + v_N c_{22} & \mathbf{0}_{1\times6} \\ \mathbf{0}_{1\times3} & c_{13} & c_{23} & c_{33} & -v_D c_{23} + v_E c_{33} & v_D c_{13} - v_N c_{33} & -v_E c_{13} + v_N c_{23} & \mathbf{0}_{1\times6} \end{array}\right]. \tag{2.4.15}$$

The measurement noise matrix $\mathbf{R}_{\mathbf{h}}$ is a (2×2) matrix defined as

$$\mathbf{R}_{\mathbf{h}} = \begin{bmatrix} \sigma_{v_2}^2 & 0 \\ 0 & \sigma_{v_3}^2 \end{bmatrix} \tag{2.4.16}$$

such that $\boldsymbol{\sigma}_{\mathbf{h}} = [\sigma_{v_2}, \sigma_{v_3}]^T$ are the standard deviations associated to $\mathbf{z}_{\mathbf{h}}$.

2.4.4 Zero-Velocity Update

If the vehicle is stopped, all velocity components of the vehicle are zero. Therefore, the following pseudomeasurements can be constructed:

$$\mathbf{z_z} = \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.4.17}$$

Hence, the model can be written as

$$\mathbf{h}(\hat{\mathbf{x}}) = \hat{\mathbf{v}}_e^l \tag{2.4.18}$$

and its linearization yields

$$\mathbf{H_z} = \left[\begin{array}{c|ccc} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times9} \end{array} \right]. \tag{2.4.19}$$

The measurement noise matrix $\mathbf{R_z}$ is defined as

$$\mathbf{R_z} = \begin{bmatrix} \sigma_{z_N}^2 & 0 & 0 \\ 0 & \sigma_{z_E}^2 & 0 \\ 0 & 0 & \sigma_{z_D}^2 \end{bmatrix} \tag{2.4.20}$$

such that $\boldsymbol{\sigma_z} = [\sigma_{z_N}, \sigma_{z_E}, \sigma_{z_D}]^T$ are the standard deviations associated to $\mathbf{z_z}$.

2.5 Static Coarse Alignment

Since an INS is a dead-reckoning system, the initial attitude matrix $\mathbf{C}_b^l$ at time t_0 has to be either known or determined. The *coarse alignment* is a procedure to estimate attitude parameters approximately. When the IMU is not moving with respect to Earth surface, the inertial sensors observe natural quantities such as local gravity, Earth rotation or local magnetic field. For that, coarse alignment has to be performed while the vehicle is not moving. Therefore, before describing the implemented static coarse alignment algorithms, a developed non-moving period detection algorithm is presented.

2.5.1 Automatic Detection of Non-moving Periods

Discriminating moving from non-moving periods may be of interest for mainly two reasons:

- Several conventional coarse alignment techniques like *leveling* and *gyrocompassing* require signals acquired in non-moving conditions to be operational.
- In many applications like car navigation, the vehicule may stop regularly. If pure inertial navigation continues during these non-moving periods, large estimation errors may occur since noise and residual time-correlated errors will be integrated. This is especially true when using low-grade sensors or sensors mounted on vibrating platforms (like a car). Therefore, the filter should be aware of such scenarios by changing its integration strategy and bounding the error growth.

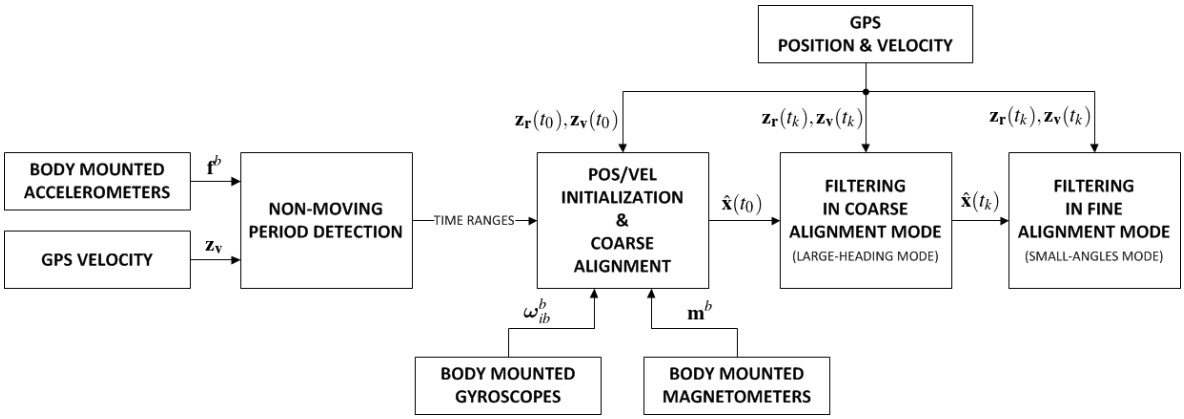

Figure 2.4: INS coarse and fine alignment procedure assuming large initial heading error.

Once non-moving periods are detected, a filtering strategy needs to be set. For instance, the filter is reset with coarse alignment algorithm or fed with pseudomeasurements like zero-velocity updates (see Figure 2.4). Regarding this, consider the set of static segments $\{S_i : i = 1, \ldots, I\}$ with each having a time span of duration $D_i \geq 0$. If $D_i < T$ with $T \geq 0$ a user-defined threshold, then pseudomeasurements (e.g. zero-velocity, position fix) updates are activated. If $D_i \geq T$, the EKF covariance matrix $\mathbf{P}_k$ is reset and the filter is put in coarse alignment. The determination of the static segments $\{S_i\}$ has been implemented in a fully automatic way. In the sequel, we propose two different algorithms which are selected depending on wether or not GNSS observations are available.

GNSS-based Detector

If GNSS-based velocity measurements $\mathbf{z_v}$ are available, a simple thresholding rule on the measured speed can be setup. The value for the threshold depends on the noise level of the velocity measurements, and must be carefully chosen according to the propagated error when computing the speed signal. Figure 2.5 shows the speed signal computed from the measured GPS velocity for a car trajectory. The gray circles are the speed values belonging to the detected static segments $\{S_i\}$, and the black triangles correspond to speed values in dynamic segments. In this case, the decision threshold was set at 0.25 m/s.

Inertial-based Detector

Specific force and angular rate measurements could directly be used as signals on which the inertial-based detector can work. We propose to adapt a feature extraction methodology initially developed for audio signal processing like silence removal or content-based segmentation Giannakopoulos [2009]. The general idea of the non-moving periods detector is a three-stage processing: first, feature sequences from which moving and non-moving periods can be discriminated are computed; second, decision thresholds are dynamically estimated; third, the sequences are detected and extracted.

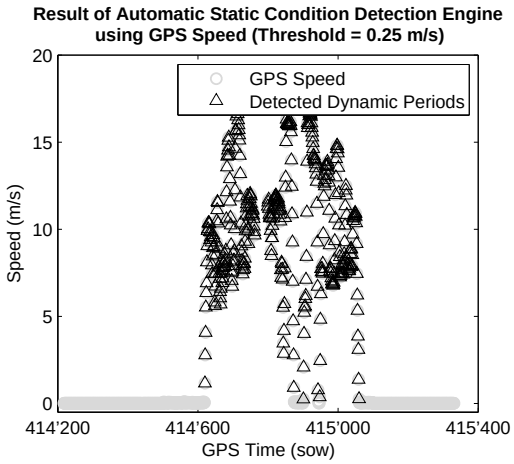

Figure 2.5: Identified non-moving GNSS epochs using the GNSS-based detector. The speed samples belonging to moving and non-moving periods are marked as black triangles and gray circles, respectively. The decision threshold was set at a speed of 0.25 m/s.

Feature Extraction Let $\{y_k : k = 1 \ldots, N\}$ be samples of the jerk signal. A short-term processing technique is adopted for calculating the chosen inertial signal feature. Since jerk signals coming from platforms subject to static and dynamic condition are non-stationary, we divide the signal in non-overlapping or overlapping short-term frames. For each of these frames, which can be assumed as "quasistationary", we calculate the feature using windowing technique. The latter is achieved by convolving $\{y_k\}$ with a window sequence $\{w_k, k = 1, \ldots, L\}$ of length L:

$$(y_i)_k = y_k w_{k-i \cdot S},\ i = 1, 2, \ldots, M \tag{2.5.1}$$

where $\{(y_i)_k : k = 1, \ldots, N\}$ is the sample sequence of the ith frame, S is the frame step (window shift). We use the rectangular window sequence which is defined as:

$$w_l = \begin{cases} 1, & 0 \leq l \leq L-1 \\ 0, & \text{elsewhere} \end{cases} \tag{2.5.2}$$

Each frame y_i has been windowed, and hence is time limited. Since $(y_i)_k$ is time limited to L nonzero samples, we can sample $(y_i)_k$ at these values, yielding the sequence $\{(y_i)_l : l = 1, \ldots, L\}$. The L value should be chosen large enough for the feature calculation stage to be reliable, and short enough for the assumed stationarity to hold. The shift value S is associated to the level of overlap. For each frame, the feature κ_i can be computed, yielding the sequence $\{\kappa_i : i = 1, \ldots, M\}$ with length $M = \frac{N-S}{L} + 1$. In audio processing, time domain features are usually combined with features that contain frequency-related information. In our context, we propose to adopt the same methodology by choosing the two following features for $\{\kappa_i\}$:

- *Signal energy* (time-domain). For each frame y_i, the energy can be calculated using

$$E_i = \frac{1}{L} \sum_{l=1}^{L} |(y_i)_l|^2 . \tag{2.5.3}$$

This feature is often used for detecting silent periods in audio signals, which is a very similar problematic to detecting non-moving periods in jerk signals. In the upper left panel of Figure 2.6, the energy sequence of the jerk signal issued from the same dataset as already used in Section 2.5.1 is shown. Comparing this to Figure 2.5 obviously reveals that large variations in the energy signal are strongly related to moving periods. The choice of this feature is motivated mainly for its simplicity and its excellent performance in cases were the level of sensor noise is not very high (as it is the case for tactical- and navigation-grade IMUs).

- *Spectral centroid* (frequency-domain). Let $\{(Y_i)_l : l = 1, \ldots, L\}$ be the sequence of Discrete Fourier Transform coefficients of the ith short-term frame $\{(y_i)_l\}$, i.e.

$$\{(y_i)_l\} \longleftrightarrow Y_i(\cdot). \tag{2.5.4}$$

Then the *spectral centroid*, C_i, of y_i is defined as the center of gravity of its spectrum, i.e.,

$$C_i = \frac{\sum_{l=1}^{L} l \cdot (Y_i)_l}{\sum_{l=1}^{L} (Y_i)_l} \tag{2.5.5}$$

which is a measure of the spectral position. In the upper right panel of Figure 2.6, the spectral centroid sequence is shown again for the same jerk signal. It is obvious that for non-moving periods, the spectral centroid sequence has very low amplitude. The choice of this feature was motivated by the fact that a signal containing components coming from platform vibrations has a larger spectral centroid. This is due to the fact that signals caused by vibrations contain higher frequencies than sensor noise.

Note that typical values for the window length L were set such that between 0.5 and 1.0 seconds of inertial samples were considered in each frame. The window shift S was chosen to yield 50% overlap between successive frames.

Non-moving Periods Segments Detection Once both feature sequences are computed, a simple threshold-based algorithm is applied for extracting the non-moving periods segments. The empirical histogram is evaluated for both feature sequences. To yield better results, a smoothing can be performed by low-pass filtering the feature sequence prior computing the histogram. This is shown as the black lines in the upper two panels of Figure 2.6. Then, the first two local maxima of each histogram, denoted as $M_{1,s}$ and $M_{2,s}$ for $s = 1, 2$, are detected and the two thresholds T_s computed as a weighted average

$$T_s = \frac{p \cdot M_{1,s} + M_{2,s}}{p + 1} \quad \text{for} \quad s = 1, 2 \tag{2.5.6}$$

where weight p is user-defined. The thresholds are depicted as black horizontal lines in the two upper panels of Figure 2.6. The two feature sequences are then thresholded, and the segments are constructed by merging successive frames for which the respective feature values (for both feature sequences) are below the computed thresholds.

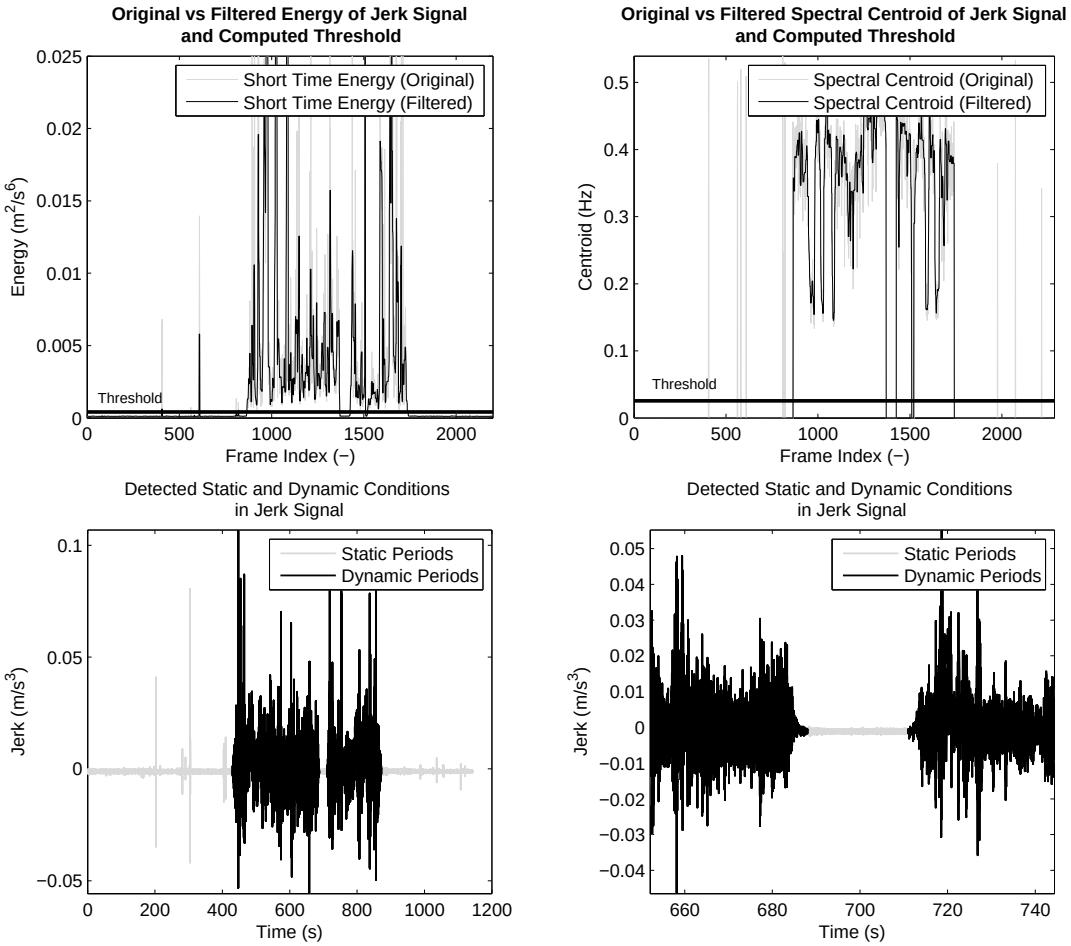

Figure 2.6: Identified non-moving periods using the inertial-based detector. The upper panels show the linear jerk signal energy (left panel) and spectral centroid (right panel) from which the decision threshold is computed (black horizontal lines). Both the original (gray curves) and low-pass filtered (black curves) feature sequences are drawn. The lower panels depict the final result on the complete jerk signal (left panel) and a close view on a period during which the car was stopped (right panel). The epochs detected as belonging to moving and non-moving periods are drawn in black and gray, respectively. Note that the jerk peaks in the detected non-moving periods are due to parasite movements of the vehicle which can be filtered out.

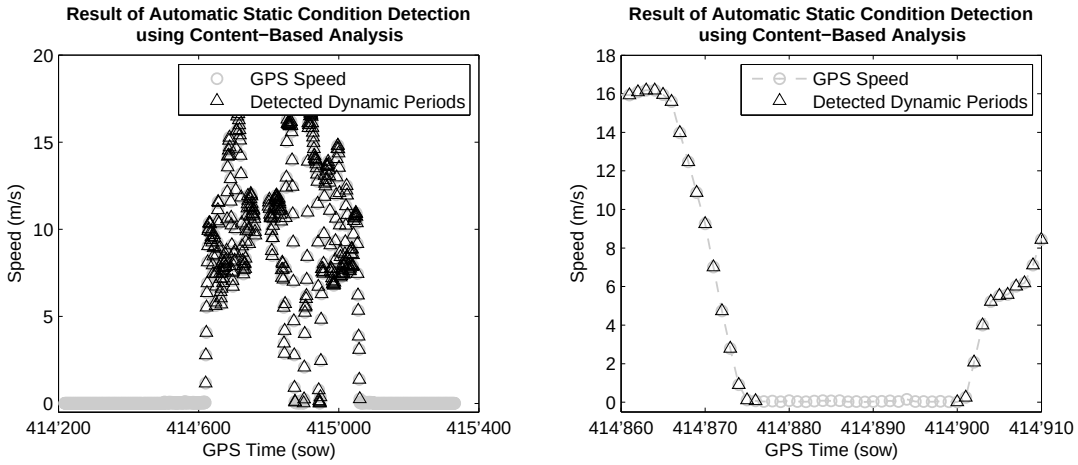

Figure 2.7: Identified non-moving GNSS epochs using the inertial-based detector. The left panel shows the complete speed signal in which samples detected in non-moving and moving periods are coded as gray circles and black triangles, respectively. The right panel provides a close view on a period during which the car was stopped.

Results The result of the feature-based detection is illustrated in the lower panels of Figure 2.6. The panels show the jerk signal in which the non-moving periods are colored in gray, while the regions containing dynamics are colored in black. In both panels of Figure 2.7, the GPS speed measurements corresponding to detected non-moving and moving periods are color-coded as gray circles and black triangles, respectively. The right panel provides a close view on a detected non-moving period during which the car was stopped at a red light.

2.5.2 Static Self-Alignment using Accelerometers and Gyroscopes

In this approach, the coarse alignment is based on the principle that the accelerometers sense only gravity $\mathbf{g}^l$, while the gyroscopes sense only the Earth rate in the b-frame, $\boldsymbol{\omega}_{ie}^b$. For that, it is assumed that initial position $\mathbf{r}_e^l(t_0)$ is known and that $\mathbf{v}_e^l(t_0) = 0$ and $\dot{\mathbf{v}}_e^l(t_0) = 0$. Such an alignment procedure can be formulated by the following relationship:

$$\begin{bmatrix} -\mathbf{f}^b & \boldsymbol{\omega}_{ib}^b & -\mathbf{f}^b \times \boldsymbol{\omega}_{ib}^b \end{bmatrix}_{t_k} = \mathbf{C}_b^l(t_0) \begin{bmatrix} -\mathbf{g}^l & \boldsymbol{\omega}_{ie}^l & -\mathbf{g}^l \times \boldsymbol{\omega}_{ie}^l \end{bmatrix}_{t_0}. \tag{2.5.7}$$

Therefore, the initial attitude matrix can be estimated by

$$\mathbf{C}_b^l(t_0) = \begin{bmatrix} -\mathbf{f}^b & \boldsymbol{\omega}_{ib}^b & -\mathbf{f}^b \times \boldsymbol{\omega}_{ib}^b \end{bmatrix}_{t_k} \cdot \begin{bmatrix} -\mathbf{g}^l & \boldsymbol{\omega}_{ie}^l & -\mathbf{g}^l \times \boldsymbol{\omega}_{ie}^l \end{bmatrix}_{t_0}^{-1} \tag{2.5.8}$$

Although it is clear from Eq. (2.5.8) that the coarse self-alignment could be done at each epoch t_k, more accurate results can be obtained by averaging the data over the detected non-moving period, while estimating residual systematic errors. Note that the matrix on the right hand side of Eq. (2.5.8) could be noninvertible if the observed angular rotation $\boldsymbol{\omega}_{ie}^l$ is close to zero (i.e. close to poles).

2.5.3 Static Alignment using Accelerometers and Magnetometers

In low-grade (e.g. MEMS-based) gyroscopes, the noise level may typically be superior to 0.1 deg/s/$\sqrt{\text{Hz}}$ and the residual systematic errors as high as several deg/s. In such conditions, the Earth rate cannot be sensed, and therefore the classical self-alignment technique presented in the previous section cannot be applied. For this reason, Waegli [2009] proposed a modified coarse alignment method using MEMS-based magnetometers. The principle is similar to the alignment of a pair of vectors but magnetic field is used instead of Earth rotation. Such an alignment method can be written as

$$\left[\begin{array}{ccc} -\mathbf{f}^b & \mathbf{m}^b & -\mathbf{f}^b \times \mathbf{m}^b \end{array} \right]_{t_k} = \mathbf{C}_b^l(t_0) \left[\begin{array}{ccc} -\mathbf{g}^l & \mathbf{m}^l & -\mathbf{g}^l \times \mathbf{m}^l \end{array} \right]_{t_0} . \tag{2.5.9}$$

where $\mathbf{m}^b$ are the magnetic measurements obtained from the magnetometers, and $\mathbf{m}^l$ is the Earth magnetic field deduced from any global or local reference model (e.g. the *World Magnetic Model* (WMM) described in Anonymous). The initial attitude can be estimated by

$$\mathbf{C}_b^l(t_0) = \left[\begin{array}{ccc} -\mathbf{f}^b & \mathbf{m}^b & -\mathbf{f}^b \times \mathbf{m}^b \end{array} \right]_{t_k} \cdot \left[\begin{array}{ccc} -\mathbf{g}^l & \mathbf{m}^l & -\mathbf{g}^l \times \mathbf{m}^l \end{array} \right]_{t_0}^{-1} . \tag{2.5.10}$$

Again, $\mathbf{f}^b$ and $\mathbf{m}^b$ observations can be averaged over a non-moving period to suppress the noise level. Note that the matrix on the right hand side of Eq. (2.5.10) could be noninvertible if the local magnetic field vector $\mathbf{m}^l$ is close to zero (i.e. close to magnetic poles).

2.5.4 Quaternion Estimation Algorithm

The *quaternion estimation* (QUEST) algorithm exploits the Earth's gravity and magnetic field by minimizing the following cost function $J(\cdot)$ Wahba [1965], Waegli [2009]:

$$J\left(\mathbf{q}_l^b\right) = \frac{1}{2} \sum_{j=1}^{2} w_k \left(\mathbf{b}_j - \mathbf{C}_l^b \mathbf{n}_j \right)^2 \tag{2.5.11}$$

with the condition $\left(\mathbf{q}_l^b\right)^T \otimes \mathbf{q}_l^b = 1$, and w_k being weights. The term $\mathbf{b}_j$ with $j = 1, 2$ expresses a measurement in the b-frame (i.e. magnetic observation $\mathbf{m}^b$ or specific force $\mathbf{f}^b$), and $\mathbf{n}_j$ a corresponding reference value in the l-frame (i.e. $\mathbf{m}^l$ or $\mathbf{g}^l$). Such a minimization problem is known as *Wahba's problem* Wahba [1965] and has the advantage of directly providing unambiguous orientation.

An experimental evaluation of the orientation initialization performance based on the QUEST algorithm and the method presented in Section 2.5.3 using MEMS-based sensors was done by Waegli [2009] .

2.6 INS Fine Alignment

Once the coarse alignment is performed, the *fine alignment* refines the approximate attitude by estimating misalignment angles in the EKF. Generally, fine alignment is achieved in-motion assuming known initial attitude. A detailed review on the alignment methods can be found in Kong [2000]. Here, we describe briefly the implemented algorithms.

Basically, most in-motion alignment methods assume small initial misalignments, as it is the case in models (1.7.25) and (1.7.42). However, the previous section mentioned that low-grade IMUs have generally lower resolution than the observable quantity due to the Earth rate. In such cases, the self-alignment will not work and external sensors such as magnetometers are required. If they are not available or if the local magnetic deviations are too large (e.g. on a machine), the initial misalignment may be large and the small angle assumption made in models (1.7.25) and (1.7.42) is no longer valid. Note that similar conditions may occur also for tactical-grade gyroscopes if the (static) averaging time is too short. We implemented two approaches able to deal with large uncertainty in initial attitude.

2.6.1 Quaternion-based Large-angle Misalignment Algorithm

In the context of Skaloud et al. [2009], Tomé and Yalak [2008], an in-motion alignment technique compatible with large misalignment angles has been developed. The algorithm is based on the true frame approach and corrects the attitude quaternion at each EKF update stage. The *a priori* estimated attitude quaternion $(\hat{\mathbf{q}}_b^l)_k^-$ for digital times $\{t_k : k \in \mathbb{Z}^+\}$ is updated using the *a posteriori* orientation errors estimate $(\hat{\boldsymbol{\varepsilon}}^l)_k^+$ provided by the quaternion mechanization Tomé and Yalak [2008]:

$$(\hat{\mathbf{q}}_b^l)_k^+ = \left[(\hat{\boldsymbol{\varepsilon}}^l)_k^+\right]_q \otimes (\hat{\mathbf{q}}_b^l)_k^- \tag{2.6.1}$$

where $\left[(\hat{\boldsymbol{\varepsilon}}^l)_k^+\right]_q$ is the quaternion equivalent of the estimated attitude error angles, $(\hat{\boldsymbol{\varepsilon}}^l)_k^+$. After coarse alignment, the filter is put in fine alignment mode using the following nonlinear large attitude form for $\left[(\hat{\boldsymbol{\varepsilon}}^l]_k^+\right)_q$:

$$\left[(\hat{\boldsymbol{\varepsilon}}^l)_k^+\right]_q = \begin{bmatrix} \cos\left(\frac{||\boldsymbol{\varepsilon}^l||}{2}\right) \\ \frac{\boldsymbol{\varepsilon}^l}{||\boldsymbol{\varepsilon}^l||}\sin\left(\frac{||\boldsymbol{\varepsilon}^l||}{2}\right) \end{bmatrix}. \tag{2.6.2}$$

When the $\hat{\boldsymbol{\varepsilon}}^l$ angles diminish to one degree, the algorithm switches to the small angle approximation in which $\left((\hat{\boldsymbol{\varepsilon}}^l)_k^+\right)_q$ is approximated by

$$\left[(\hat{\boldsymbol{\varepsilon}}^l)_k^+\right]_q \approx \begin{bmatrix} \sqrt{1 - ||(\hat{\boldsymbol{\varepsilon}}^l)_k^+||^2} \\ \frac{1}{2}(\hat{\boldsymbol{\varepsilon}}^l)_k^+ \end{bmatrix}. \tag{2.6.3}$$

In practice, the initial tilt angles are approximated in the coarse alignment stage and the initial heading is set in the $(-\pi, +\pi)$ range (see Figure 2.4). Most of the time, the large attitude errors are corrected in a very short period of time (typically a few iterations).

2.6.2 Model for Large Misalignment Errors based on c-Frame Approach

In Kong et al. [1999], a general nonlinear c-frame approach that assumes large misalignment angles has been proposed. They generalized the computer frame navigation model described in Eq. (1.7.42), yielding the large attitude navigation error model:

$$\begin{bmatrix} \delta\dot{\mathbf{r}}^c \\ \delta\dot{\mathbf{v}}^c \\ \dot{\boldsymbol{\psi}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\omega}_{ec}^c \times \delta\mathbf{r}^c + \delta\mathbf{v}^c \\ \left(\mathbf{I} - \mathbf{C}_p^c\right)\mathbf{f}^p - (2\boldsymbol{\omega}_{ie}^c + \boldsymbol{\omega}_{ec}^c) \times \delta\mathbf{v}^c + \delta\mathbf{g}^c + \mathbf{C}_b^p\delta\mathbf{f}^b \\ (\mathbf{I} - \mathbf{C}_c^p)\cdot(\boldsymbol{\omega}_{ie}^c + \boldsymbol{\omega}_{ec}^c) - \mathbf{C}_b^p\delta\boldsymbol{\omega}_{ib}^b \end{bmatrix} \tag{2.6.4}$$

with

$$\mathbf{C}_p^c = \mathbf{C}_1(\psi_N) \cdot \mathbf{C}_2(\psi_E) \cdot \mathbf{C}_3(\psi_D). \tag{2.6.5}$$

Let

$$\begin{array}{llll} s_N & = \sin(\psi_N), & c_N & = \cos(\psi_N) \\ s_E & = \sin(\psi_E), & c_E & = \cos(\psi_E) \\ s_D & = \sin(\psi_D), & c_D & = \cos(\psi_D) \end{array} \tag{2.6.6}$$

Then, Eq. (2.6.5) becomes

$$\mathbf{C}_p^c = \begin{bmatrix} c_E c_D - s_E s_N s_D & -c_N s_D & s_E c_D + c_E s_N s_D \\ c_E s_D + s_E s_N c_D & c_N c_D & s_E s_D - c_E s_N c_D \\ -s_E c_N & s_N & c_E c_N \end{bmatrix}$$

Two special cases are considered in Kong et al. [1999]:

- If ψ_N, ψ_E and ψ_D are small, then by using the infinitesimal rotation approximations defined in Section 1.2.4, we get $\mathbf{C}_p^c \approx \mathbf{I} + [\boldsymbol{\psi}\times]$ and $\mathbf{C}_c^p \approx \mathbf{I} - [\boldsymbol{\psi}\times]$. Substituting both approximations into Eq. (2.6.4) yields the small angle model of Eq. (1.7.42).
- If ψ_D is large and ψ_N and ψ_E are both small, we are in the case of large uncertainty in heading and low uncertainties in tilt angles. With this assumption, $\mathbf{C}_p^c$ is approximated by

$$\mathbf{C}_p^c = \begin{bmatrix} c_D & -s_D & \psi_E c_D + \psi_N s_D \\ s_D & -c_D & \psi_D s_D - \psi_N c_D \\ -\psi_E & \psi_N & 1 \end{bmatrix}, \tag{2.6.7}$$

  where, again, the approximations defined in Section 1.2.4 were used. Substituting this matrix $\mathbf{C}_p^c$ into Eq. (2.6.4) yields the large heading uncertainty model.

In practice, the platform leveling is achieved using coarse alignment techniques in which tilt angles are generally quite well approximated (even when using low-grade IMUs), and the heading is arbitrarily fixed in the $(-\pi, +\pi)$ range (see Figure 2.4). Then, the EKF is operated based on model (2.6.4) in which approximation (2.6.7) is used, and at each filter update, the $\hat{\boldsymbol{\psi}}^l$ estimation is used to rotate the c-frame to coincide with the p-frame. As stated by Kong et al. [1999], the large heading errors are usually compensated in a short period of time. When the misalignment angles are below one degree, the small angle model defined in Eq. (1.7.42), or its equivalent true frame model (1.7.25), is used.

Chapter 3

Stochastic Processes

3.1 Introduction

This chapter provides background information necessary to understand the notations, developments and analyses in forthcoming chapters. It is assumed that the reader is already familiar with the basic concepts and therefore only the essential elements are provided. The content of this chapter is extensively based on Gelb [1974], Hamilton [1994], and some additional references mentioned throughout the text.

When acquiring or designing sensors, one naturally searches for the best sensors that achieve the requirements in terms of aimed performance and costs. If the sensor manufacturers remove, to the best extent possible, all predictable errors corrupting the sensor's output (e.g cross-coupling errors, temperature dependent errors), there are generally still remaining noncompensated errors. These must be accomodated within the design of an integrated navigation system by appropriate modeling of the stochastic process.

The concepts of stochastic process together with associated fundamental elements are defined in Sections 3.2 and 3.3. The broad-band white noise is handled in Section 3.4 before moving to the properties of linear systems driven by random signals in Section 3.5. Finally, we define in Section 3.6 all the stochastic processes that will be used throughout the thesis.

3.2 Definition

Let $t \in \mathcal{T}$ denote the time. A stochastic process is defined as a real-valued function $X(\cdot)$ such that for any fixed t, $X(t)$ is a random variable. In particular, if $\mathcal{T} \subseteq \mathbb{R}$, then $X(t)$ is a *continuous-time stochastic process*. If $\mathcal{T} \subseteq \mathbb{Z}$, t is a sequence of times indexed by variable k, noted $\{t_k : k \in \mathbb{Z}\}$, and $X(\cdot)$ becomes a sequence of random variables, $\{X_k : k \in \mathbb{Z}\}$, recorded at a sampling interval Δt assumed to have time units (e.g. seconds).

3.3 Stationarity, Autocorrelation Sequence and Power Spectral Density Function

A discrete-time process $\{X_k : k \in \mathbb{Z}\}$ is said to be (weakly) stationary if it satisfies the following two properties:

1. $\mathbb{E}[X_k] = \mu_X$ for all $k \in \mathbb{Z}$; i.e. the expected value of all components X_k in $\{X_k\}$ is time invariant; and
2. $\text{cov}[X_k, X_{k+\tau}] = s_{X,\tau}$ for all $k, \tau \in \mathbb{Z}$; i.e. the covariance between any two components X_k and $X_{k+\tau}$ is independent of time.

The sequence $\{s_{X,\tau} : \tau \in \mathbb{Z}\}$ is called the *autocovariance sequence* which is symmetric about $\tau = 0$ in the sense that $s_{X,-\tau} = s_{X,\tau}$ for all τ. The *autocorrelation sequence* (ACS), noted $\{\rho_{X,\tau} : \tau \in \mathbb{Z}\}$, can be defined as

$$\rho_{X,\tau} = \text{cor}[X_k, X_{k+\tau}] = \frac{s_{X,\tau}}{s_{X,0}}. \tag{3.3.1}$$

Under the condition that

$$\sum_{\tau=-\infty}^{\infty} s_{X,\tau}^2 < \infty, \tag{3.3.2}$$

the following relationship can be written:

$$S_X(f) = \Delta t \sum_{\tau=-\infty}^{\infty} s_{X,\tau} e^{-i2\pi f \tau \Delta t} \quad \text{for} \quad |f| \leq f_N \equiv \frac{1}{2\Delta t} \tag{3.3.3}$$

where f_N i the *Nyquist frequency*, $\Delta t = t_k - t_{k-1}$ is the sampling interval, and $S_X(\cdot)$ is the *Power Spectral Density* (PSD) function. Standard Fourier analysis shows that $\{s_{X,\tau}\}$ is the inverse Fourier transform of $S_X(\cdot)$:

$$s_{X,\tau} = \int_{-f_N}^{f_N} S_X(f) e^{i2\pi f \tau \Delta t} df, \quad \text{for} \quad \tau \in \mathbb{Z}. \tag{3.3.4}$$

In particular, for $\tau = 0$, the following fundamental result can be calculated:

$$\int_{-f_N}^{f_N} S_X(f) df = s_{X,0} = \text{var}[X_k], \tag{3.3.5}$$

meaning that $S_X(\cdot)$ decomposes the process variance with respect to frequency.

Note that we will designate $\rho_X(\tau)$ as the continuous equivalent of the ACS defined in Eq. (3.3.1), i.e.

$$\text{cor}[X(t), X(t+\tau)] = \rho_X(\tau) \tag{3.3.6}$$

where $t, \tau \in \mathbb{R}$, and will be referred to as the *autocorrelation function (ACF* (ACF).

3.4 White Noise

A stationary process of particular interest is the continuous white noise process, $W(t)$. By definition, the PSD of $W(t)$ is constant:

$$S_W(f) = q \tag{3.4.1}$$

where $q \in \mathbb{R}^+$ is the amplitude of the continuous process PSD[1]. Then, by Eq. (3.3.1), the ACF is

$$\rho_W(\tau) = q\delta(\tau). \tag{3.4.2}$$

Despite the fact that a white noise process can have any probability distribution, the Gaussian distribution is often assumed.

The equivalent discrete-time white noise process $\{W_k : k \in \mathbb{Z}\}$ is the average of $W(t)$ over a small Δt, i.e.

$$W_k = \int_{t_{k-1}}^{t_k} W(s)ds, \quad \text{with } \Delta t = t_k - t_{k-1}, \tag{3.4.3}$$

and its ACS is

$$\rho_{W,\tau} = \sigma_W^2 \delta_\tau \tag{3.4.4}$$

where δ_τ is the dimensionless *Kronecker delta function*[2]. If the navigation system is based on sensor error models which account for all known or predictable effects, then the remaining residuals can be accurately modeled as the output of a discrete linear system driven by Gaussian white noise input W_k. In such as case, $W_k \sim \mathcal{N}\left(0, \sigma_{WN}^2\right)$.

3.5 Statistical Properties of Linear Systems with Random Inputs

As discussed in Chapter 1, sensor biases are often modeled via the state augmentation approach. Thus, linear stochastic systems forced by Gaussian white noise vector $W(t)$ and providing output $X(t)$ are of special interest.

Consider a (deterministic) time-invariant, causal, linear system with stochastic input $W(t)$ and output $X(t)$. Then,

$$X(t) = \int_{-\infty}^{t} h(t-\lambda)W(\lambda)d\lambda \tag{3.5.1}$$

can be seen as a filter in which $h(\cdot)$ is called the *impulse response* of the linear system. Using our notation for Fourier transform pairs, we can write

$$h(\cdot) \longleftrightarrow H(\cdot) \tag{3.5.2}$$

[1] Since $\delta(\tau)$ has units of $1/\text{sec} = \text{Hz}$, q has the units of W squared, divided by Hz.

[2] The dimension of σ_W^2 are the dimensions of W_k squared.

where $H(\cdot)$ is called the *transfer function*. If $W(t)$ is wide sense stationary, manipulation of Eq. (3.5.1) leads to relationships between the continuous correlation function of $W(t)$ and $X(t)$ Farrell [2008]:

$$\rho_{WX}(\tau) = \int_{-\infty}^{\infty} \rho_W(\tau+\lambda)h(\lambda)^T d\lambda \tag{3.5.3}$$

$$\rho_X(\tau) = \int_{-\infty}^{\infty} h(\lambda)\rho_{WX}(\tau-\lambda)d\lambda. \tag{3.5.4}$$

Note that $\rho_{WX}(\tau)$ is the continuous *cross-correlation function* defined by

$$\text{cor}\left[W(t), X(t+\tau)\right] = \rho_{WX}(\tau) \tag{3.5.5}$$

where $t, \tau \in \mathbb{R}$. The Fourier Transforms of Eq. (3.5.3) and (3.5.4) yield

$$S_{WX}(f) = S_W(f)H(f)^* \tag{3.5.6}$$

$$S_X(f) = H(f)S_W(f)H(f)^* \tag{3.5.7}$$

where $H(\cdot)^*$ is the complex conjugate transpose of $H(\cdot)$, and $S_{WX}(\cdot)$ is the *cross-PSD function*. Eq. (3.5.7) can be exploited for finding a linear system realization $H(\cdot)$ which produces a given colored noise process $X(t)$ from a white noise input $W(t)$ Farrell [2008].

3.6 Stochastic Error Models

3.6.1 Random Constant

A random constant can be generated as the output of an integrator with no input, but with an initial condition $X(t_0)$, $t_0 \in \mathbb{R}$ and variance $P(0) = P_{X_0}$ such that $X(t_0) \sim \mathcal{N}(\mu_{X_0}, P_{X_0})$. The controlling differential equation is

$$\dot{X}(t) = 0 \tag{3.6.1}$$

starting from $X(t_0)$. Since the samples are constant in time, the ACF is given by

$$\rho_X(\tau) = P_{X_0} + \mu_{X_0}^2 \quad \text{for} \quad \tau \in \mathbb{R}, \tag{3.6.2}$$

and the PSD is

$$S_X(f) = \left(P_{X_0} + \mu_{X_0}^2\right)\delta(f). \tag{3.6.3}$$

The equivalent discrete process is the sequence $\{X_k : k \in \mathbb{Z}\}$ such that

$$X_{k+1} = X_k \tag{3.6.4}$$

considering $X_0 \sim \mathcal{N}\left(\mu_{X_0}, \sigma_{RC}^2\right)$. Some parts of the sensor error (e.g. scale factor, switch-on to switch-on nonrepeatability biases) can be modeled using random constant processes.

3.6.2 Random Walk

A random walk is the output of an integrator driven by Gaussian white noise:

$$\dot{X}(t) = W(t) \quad \text{for} \quad t \in \mathbb{R}. \tag{3.6.5}$$

Integrating this equation yields

$$X(t) = \int_{t_0}^{t} W(s)ds \tag{3.6.6}$$

with initial condition $X(t_0)$ and ACF $\rho_W(\tau) = q \cdot \delta(\tau)$, $t_0, \tau \in \mathbb{R}$. The random walk process is not stationary. Thus, the ACF depends on both involved time instants (not only on the time interval):

$$\rho_X(\tau) = \begin{cases} q \cdot (t - t_0) & \text{if } t + \tau \geq t > t_0 \\ q \cdot (t + \tau - t_0) & \text{if } t > t + \tau > t_0 \end{cases} \tag{3.6.7}$$

where q is the amplitude of the PSD of $W(t)$. The variance of the random walk grows linearly with time, implying

$$\text{var}\left[X(t)\right] = q \cdot (t - t_0). \tag{3.6.8}$$

The equivalent discrete process is the sequence $\{X_k : k \in \mathbb{Z}\}$ such that

$$X_{k+1} = X_k + W_k \tag{3.6.9}$$

with $W_k \sim \mathcal{N}\left(0, \sigma_{RW}^2\right)$, and $\sigma_{RW}^2 = q \cdot \Delta t$. Random walk processes are frequently used in inertial navigation due to the time integration of sensor data. In case of an accelerometer, white noise corrupting $\mathbf{f}^b$ samples propagates as a random walk when the measurements are integrated to obtain velocity. In case of a gyroscope, white noise in $\boldsymbol{\omega}_{ib}^b$ results in a random walk of the attitude obtained by integrating the gyroscope measurements.

3.6.3 Exponentially Time-Correlated (First-Order Gauss-Markov) Process

These models are first order lags driven by Gaussian white noise $W(t)$ of strength q described by

$$\dot{X}(t) = -\beta X(t) + W(t) \quad \text{for} \quad t \in \mathbb{R}, \tag{3.6.10}$$

where $q = 2\sigma_{GM}^2\beta$ with $\beta \in \mathbb{R}^+$ assumed constant in time. The initial condition $X(t_0)$ can be chosen to make the process stationary. The associated ACF decreases exponentially, i.e.

$$\rho_X(\tau) = \sigma_{GM}^2 e^{-\beta|\tau|} \quad \text{for} \quad \tau \in \mathbb{R}, \tag{3.6.11}$$

where $\sigma_{GM}^2 = \text{var}\left[X(t_0)\right]$. The PSD is given by

$$S_X(f) = \frac{2\sigma_{GM}^2\beta}{4\pi^2 f^2 + \beta^2}. \tag{3.6.12}$$

The discrete version is the sequence $\{X_k : k \in \mathbb{Z}\}$ described by

$$X_{k+1} = e^{-\beta \Delta t} X_k + W_k \tag{3.6.13}$$

such that $W_k \sim \mathcal{N}(0, q_k)$ with

$$q_k = \frac{q}{2\beta}\left[1 - e^{-2\beta\Delta t}\right] \tag{3.6.14}$$

and $q/(2\beta) = \sigma^2_{GM}$. Gauss-Markov processes are frequently used to approximate a wide variety of empirically observed band-limited (wide or narrow-band) noises. The parameter β reflects the degree of correlation and can be seen as the inverse correlation time $t_c = 1/\beta$. If t_c is large, the signal is highly correlated in time. If t_c is near zero, the process looks like a random constant. If t_c is small, the signal quickly decorrelates and the process becomes similar to white noise.

3.6.4 Quantization Noise

Quantization noise is due to the digital nature of the inertial sensor outputs. The PSD of this particular process is given by

$$S_X(f) = 4\Delta t Q^2 \sin^2\left(\frac{\pi f}{\Delta t}\right) \tag{3.6.15}$$

where Q is the quantization noise coefficient. The differential equation description of quantization noise is expressed as follows Han and Wang:

$$X(t) = Q \cdot \sqrt{\Delta t} \cdot \dot{U}(t) \tag{3.6.16}$$

where $U(t)$ is a unit white noise. Therefore, the corresponding discrete sequence $\{X_k : k \in \mathbb{Z}\}$ can be generated by discretizing Eq. (3.6.16).

3.6.5 Random Rate Ramp

The random rate ramp is a function which grows linearly with time. The growth rate is a random variable with given probability density. Two states are necessary to control such process:

$$\begin{aligned} \dot{X}_1(t) &= X_2(t) \\ \dot{X}_2(t) &= 0 \end{aligned} \tag{3.6.17}$$

where $X_1(t)$ is the random ramp process, $X_2(t)$ is the slope of the ramp, and $t \in \mathbb{R}$. The mean square value of $X_1(t)$ grows parabolically with time, i.e. $\mathbb{E}\left[X_1^2(t)\right] = \mathbb{E}\left[X_2^2(0)\right] \cdot t^2$.

The discrete sequence $\{X_k : k \in \mathbb{Z}\}$ can be obtained by

$$\begin{aligned} X_{1,k+1} &= X_{1,k} + X_{2,k} \cdot \Delta t \\ X_{2,k+1} &= X_{2,k} \end{aligned} \tag{3.6.18}$$

for $k \in \mathbb{Z}$ such that $X_{1,0} \sim \mathcal{N}\left(\mu_{X_{1,0}}, \sigma^2_{RR,1}\right)$ and $X_{2,0} \sim \mathcal{N}\left(\mu_{X_{2,0}}, \sigma^2_{RR,2}\right)$. In the sequel, we will denote the ramp slope as $c_{RR} = X_2$. Note that the random ramp is not a stationary process.

3.6.6 Bias Instability

The bias instability, also known as 1/f or *flicker noise*, models bias fluctuations in signals. The PSD of a bias instability process $X(t)$ is

$$S_X(f) = \begin{cases} \left(\frac{B^2}{2\pi}\right)\frac{1}{f} & \text{if } f \leq f_0 \\ 0 & \text{if } f > f_0 \end{cases} \tag{3.6.19}$$

where B is the bias instability coefficient and f_0 is the cutoff frequency Std 1293-1998 [1998]. Numerous approaches have been developed over the years to approximate 1/f noise. The most common are ARMA-based models, wavelet analysis and first-order Gauss-Markov processes. One way of obtaining a discrete sequence $\{X_k : k \in \mathbb{Z}\}$ of fluctuating bias is using

$$X_{k+1} = \begin{cases} W_{k+1} & \text{if } \mathrm{mod}(t_{k+1}, T_{BI}) = 0 \\ X_k & \text{otherwise} \end{cases} \tag{3.6.20}$$

with T_{BI} the period of the fluctuations, and $W_k \sim \mathcal{N}(0, \sigma_{BI}^2)$. Electronics and other components susceptible to random flickering generate noise of low-frequency nature which can be modeled using a bias instability process.

Table 3.1: PSD and time representation of the stochastic processes used in this thesis. Note that *iid* stands for *independent and identically distributed*.

Process	**Abbr.**	**Parameter**	**Power Spectral Density**	**Differential Equation** **Discrete State-Space Model**	**Note**
White Noise	WN	σ^2_{WN}	$S_X(f) = \sigma^2_{WN}$	$\dot{X}(t) = W(t)$ $X_k = W_k$	Gaussian white noise $W_k \overset{iid}{\sim} \mathcal{N}\left(0, \sigma^2_{WN}\right)$
Random Walk	RW	σ^2_{RW}	$S_X(f) = \frac{\sigma^2_{RW}}{(2\pi f)^2}$	$\dot{X}(t) = W(t)$ $X_{k+1} = X_k + W_k$	– $W_k \overset{iid}{\sim} \mathcal{N}\left(0, \sigma^2_{RW}\right)$
Gauss-Markov	GM	$\beta,\ \sigma^2_{GM}$	$S_X(f) = \frac{2\sigma^2_{GM}\beta}{(2\pi f)^2+\beta^2}$	$\dot{X}(t) = -\beta X(t) + W(t)$ $X_{k+1} = e^{-\beta\Delta t} X_k + W_k$	– $W_k \overset{iid}{\sim} \mathcal{N}(0, q)$ with $q = \sigma^2_{GM}\left(1 - e^{2\beta\Delta t}\right)$
Bias Instability	BI	$T_{BI},\ \sigma^2_{BI}$	$S_X(f) = \begin{cases} \left(\frac{\sigma^2_{BI}}{2\pi f}\right) & \text{if } f \le f_0; \\ 0 & \text{if } f > f_0. \end{cases}$	– $X_{k+1} = \begin{cases} W_{k+1} & \text{if mod}\,(t_{k+1}, T_{BI}) = 0; \\ X_k & \text{otherwise.} \end{cases}$	– $W_k \overset{iid}{\sim} \mathcal{N}\left(0, \sigma^2_{BI}\right)$
Quantization Noise	QN	Q	$S_X(f) = 4\Delta t Q^2 \sin^2\left(\frac{\pi f}{\Delta t}\right)$, if $f < \frac{\Delta t}{2}$	$\dot{X}(t) = Q \cdot \sqrt{\Delta t} \cdot \dot{U}(t)$ $X_{k+1} = \sqrt{Q} \cdot (U_{k+1} - U_k)$	$U(t) = \sqrt{12} \cdot \tilde{U}(t)$ $\tilde{U}(t) \sim \mathcal{U}(0,1)$
Random Ramp	RR	c_{RR}	$S_X(f) = \frac{c_{RR}}{(2\pi f)^3}$	$\dot{X}_1(t) = X_2(t)$ $\dot{X}_2(t) = 0$ $X_{1,k+1} = X_{1,k} + X_{2,k} \cdot \Delta t$ $X_{2,k+1} = X_{2,k}$	$c_{RR} = X_2$ –

Part II

Stochastic Modeling

Chapter 4

Traditional Model Estimation Techniques

4.1 Introduction

The purpose of this chapter is to discuss the instrument error terms in $\Delta\mathbf{f}^b$ and $\Delta\boldsymbol{\omega}_{ib}^b$ defined in Eq. (1.4.3) and (1.4.4). Sensors biases are usually modeled as stochastic processes using states $\mathbf{x}_a$ and $\mathbf{x}_g$ in the augmented part of the navigation filter state space model. The choice of the processes and the following determination of their parameters is part of the filter design procedure.

In the sequel, the justification of the first part of this research is brought in Section 4.2 which demonstrates why designing a navigation filter is a non-trivial task. Section 4.3 sets the assumptions on which the next chapters rely. The subsequent Sections 4.4, 4.5, 4.6 and 4.7, present the classical methods used within the navigation community for designing filters. Finally, Section 4.8 concludes with the main limitations of these standard methods and provides the motivation for the next two chapters.

4.2 The Challenge of Filter Design

4.2.1 Problem Statement

Consider the sequence $\{y_k : k = 1, \ldots, N\}$ representing the observed one-dimensional non-compensated error signal of an accelerometer or a gyroscope. This sequence can be seen as a realization of an univariate Gaussian time series $\{Y_k : k \in \mathbb{Z}\}$ that we assume to be stationary[1] and to which the conditional distribution $F\left(Y_k | Y_{k-1}, Y_{k-2}, \ldots, \boldsymbol{\theta}\right)$ with parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^p$ is associated. Therefore,

$$Y_k \sim F\left(Y_k | Y_{k-1}, Y_{k-2}, \ldots, \boldsymbol{\theta}\right). \tag{4.2.1}$$

[1] Y_k can also be non-stationary but with stationary backward differences of order d. The first-order backward difference of Y_k is $Y_k^{(1)} = Y_k - Y_{k-1}$ and the backward difference of order d is $Y_k^{(d)} = Y_k^{(d-1)} - Y_{k-1}^{(d-1)}$.

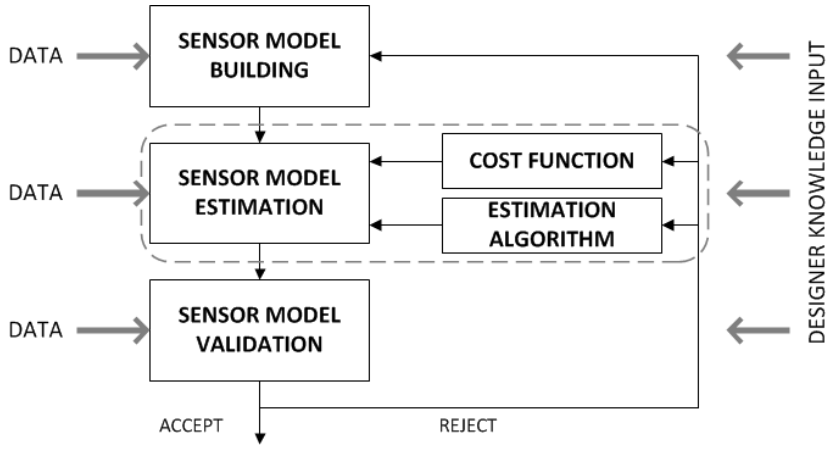

Figure 4.1: The filter design loop. In this thesis, the main focus is set on the elements enclosed by the dashed line (adapted from Nelson [2000]).

Navigation filter design requires an answer to three questions which may have significant impact on filter performance:

1. Which processes should be considered for building $F(\cdot)$ that best describes the underlying dynamics of the observed sensor error sequence $\{y_k\}$?
2. How can the parameters $\boldsymbol{\theta}$ of the resulting model $F(\cdot)$ be accurately estimated?
3. How can the model be validated?

The first question refers to the *model building* problem, which aims at finding a plausible model to be estimated. In the filter design context, the tradeoff between the accuracy of the model with respect to the application and the increased computational load involved by the augmented state must be taken into account by the designer at this stage.

The second question tackles the *model estimation* problem, which becomes nontrivial when multiple processes are mixed. At this stage, a model estimation algorithm together with a cost function must be chosen in order to select a model from the set and estimate its parameters. This will be the core topic of this and the following two chapters.

The *model validation* step is a very challenging task in inertial navigation for several reasons. First, only one realization of the trajectory is generally observed, making any statistical analysis difficult or even impossible. Second, highly variable conditions (e.g. temperature changes or vibrations) may largely affect the behavior of the sensors and thus change the underlying error dynamics. Finally, an inadequate set of observations $\{y_k\}$ and/or too many states in $F(\cdot)$ may result in a problem referred to as the *observability* problem. Although the system dynamics and the observation models should reflect the real navigation situation by including a representative number of states, the interrelationships within the dynamic model as well as the external observations may affect the observability of these states Jekeli [2000]. In this thesis, we validate models at observation level by comparing the noise structure between observed signals and synthetic signals generated under $F(\hat{\boldsymbol{\theta}})$, and at state level by analysing navigation performance with respect to a reference when inertial navigation is operating in coasting mode.

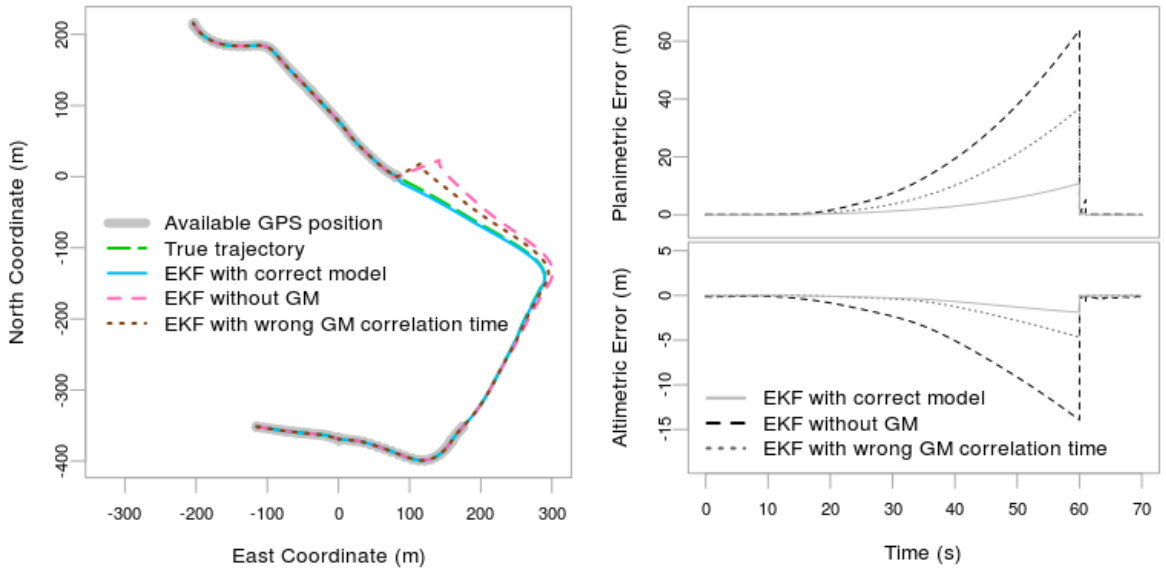

Figure 4.2: Simulation scenario demonstrating the impact of the inertial sensor calibration quality on the filtered position obtained by the EKF using different models (see Table 4.1). The results are presented on a two-dimensional planimetric frame (left panel), and separately along planimetric and altimetric axes (right panel).

The three steps must inevitably be used within a filter design loop as represented in Figure 4.1. At each step, some prior knowledge from the designer (e.g. type of application, expected sensor behavior) must be taken into account.

4.2.2 Importance of Filter Design

The quality of the sensor error modeling will directly influence the final estimated navigation solution. To illustrate this statement, we emulated accelerometer and gyroscope signals (at 100 Hz) along a known trajectory followed by a car. Then, we corrupted them with some error driven by a selected (known) model $F(\boldsymbol{\theta})$ which is the sum of a Gaussian white noise and a first-order Gauss-Markov process:

$$\begin{aligned} x_{k+1} &= e^{-\beta \Delta t} x_k + w_k, \quad w_k \sim \mathcal{N}(0, q) \\ y_k &= x_k + v_k, \quad v_k \sim \mathcal{N}(0, r) \end{aligned} \tag{4.2.2}$$

with $q = \sigma_{GM}^2 \left(1 - e^{-2\beta \Delta t}\right)$, $r = \sigma_{WN}^2$, and Δt is the sampling interval. In this example, we have $\boldsymbol{\theta} = \{\beta, \sigma_{GM}^2, \sigma_{WN}^2\}$. We then used simulated GPS positions (1 Hz) as measurements in an EKF to estimate the inertial sensor biases and calibrate the sensors. We artificially introduced a GPS-free period of 50 seconds duration during which the position error growth is solely driven by the inertial sensor errors that accumulate with time. This effect is shown in Figure 4.2 for three scenarios (see also Table 4.1). In the first trajectory (solid line), we introduced the correct and complete error model in the EKF. In the second trajectory (dotted line), we used a wrong correlation time β, while in the third trajectory (dashed line), we didn't

Table 4.1: Parameter values used in the computation of the Kalman filter trajectory for the three scenarios.

Sensor	**Scenario**	β	σ_{GM}	σ_{WN}
Accelerometers		$1/s$	$\mu g/\sqrt{Hz}$	$\mu g/\sqrt{Hz}$
	Correct model	$1.0 \cdot 10^{-4}$	50.0	70.0
	Wrong β	$1.0 \cdot 10^{-2}$	50.0	70.0
	Without GM	–	–	70.0
Gyroscopes		$1/s$	deg/s/$\sqrt{Hz}$	deg/s/$\sqrt{Hz}$
	Correct model	$1.0 \cdot 10^{-4}$	10.0	30.0
	Wrong β	$1.0 \cdot 10^{-2}$	10.0	30.0
	Without GM	–	–	30.0

include the Gauss-Markov process at all. From Figure 4.2, it can be seen that the calculated trajectory is significantly better when using the correct model compared to the second and third scenarios which mimic the situation in which the model's parameters are unknown or badly estimated. In these cases, the estimated trajectory deviates by about 60 meters from the true one after 60 seconds, while this deviation is smaller than 10 meters when the correct model is employed.

4.3 Assumptions

We are interested in estimating $\boldsymbol{\theta}$, the parameter vector that defines the model $F(\cdot)$ of the underlying error components $\Delta\mathbf{f}^b$ and $\Delta\boldsymbol{\omega}^b_{ib}$ in Eq. (1.4.3) and (1.4.4) affecting accelerometers and gyroscopes. With this respect, we make the following assumptions:

- Considering Eq. (1.4.3) and (1.4.4), we restrict the problem to the modeling and the estimation of the stochastic sensor behavior included in $\mathbf{w}_\bullet$. For clarity reasons, we define $\{y_k : k = 1, \ldots, N\}$, as the time series of samples contained in one single line of $\mathbf{w}_\bullet$, i.e. the error signal observed by one sensor;
- We restrict the set of possible models $F(\cdot)$ to linear combinations of independent stochastic processes that are presented in Chapter 3. We refer to such models as *composite Gaussian processes* which have the following definition.

  Definition 4.3.1. *A composite process is a sum of independent processes.*

  As already mentioned, issues like computational load or observability must be taken into account when designing a navigation filter. We believe that this set of model structure is fairly general to be used with a large class of sensors. In particular, combinations of multiple first-order Gauss-Markov processes are of interest since they can approximate many random processes Brown and Hwang [1997], Bryson [2002], Xing [2010].

In other words, we search for the optimal design of the augmented states in Eq. (1.8.9) by choosing the type of processes $\delta\mathbf{f}^b_j$ and $\delta\boldsymbol{\omega}^b_{ib,j}$ to put in $\delta\mathbf{x}_a$ and $\delta\mathbf{x}_g$ of Eq. (2.2.5). The next sections describe standard engineering methods which are used to design such models.

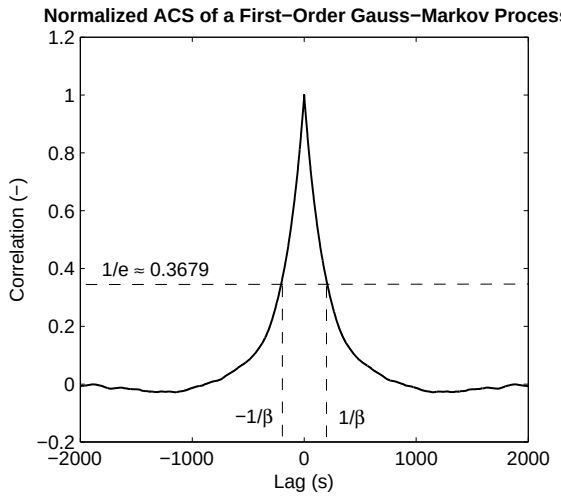

Figure 4.3: Autocorrelation sequence of the correlated part of a signal $\{y_k\}$ modeled as a first-order Gauss-Markov process. The inverse correlation time parameter β can be deduced at the e^{-1} point.

4.4 Signal Autocorrelation Method

Computing the ACS of the observed signal $\{y_k : k = 1, \ldots, N\}$ may reveal the presence or not of correlated noise. This modeling technique assumes the noise to be composed of an uncorrelated and a correlated part which are commonly handled by models of the type of Eq. (4.2.2). The procedure, described with a detailed example in Xing [2010], consists in low-pass filtering $\{y_k\}$ to remove the uncorrelated part, and then fitting a model to the ACS of the remaining correlated filter output. For example, Figure 4.3 illustrates the estimated ACS of the correlated part of an $\{y_k\}$ sequence simulated under model (4.2.2) using $\beta = 0.005$ and $\sigma_{GM} = 0.005$. The inverse correlation time β can be deduced from the normalized ACS using Eq. (3.6.11) (i.e. $\sigma = 1$) at $\hat{\rho}_{y,1/\beta} = e^{-1} \approx 0.3679$, and $\hat{\sigma}_{GM} = \sqrt{\text{var}\,[y_k]}$.

However, when several processes are superposed, the ACS does not always enable to clearly separate them. The estimation of the model parameters becomes challenging or even impossible in such cases.

4.5 Variance Methods

4.5.1 Allan Variance

The *Allan variance* was invented in 1966 by David Allan when he criticized the use of the sample variance estimator in the context of non time series. He proposed the Allan variance as an alternative theoretical measure of variability Allan [1966]. Although this method was originally intended to study the stability of oscillators, it has been successfully applied to problems dealing with a large number of different types of sensors, among which stands the modeling of inertial sensor errors Guerrier [2009], Hou [2004], El-Sheimy et al. [2008], Strus

et al. [2007], Xing and Gebre-Egziabher [2008]. In 1998, the IEEE standard put forward this technique as a noise identification method to determine the characteristics of the underlying random processes that perturb data The Institute of Electrical and Electronics Engineers Inc. [2005].

Let $\bar{Y}_k(\tau)$ be the sample average of τ consecutive observations, i.e.

$$\bar{Y}_k(\tau) = \frac{1}{\tau}\sum_{j=0}^{\tau-1} Y_{k-j}.$$

The Allan variance at scale τ, noted as $\sigma^2_{\bar{Y}}(\tau)$, aims to measure how much the sample average $\bar{Y}_k(\tau)$ changes from one period of time to another. This quantity is defined as half the expectation of squared differences between adjacent nonoverlapping $\bar{Y}(\tau)$:

$$\sigma^2_{\bar{Y}}(\tau) = \frac{1}{2}\mathbb{E}\left[\left(\bar{Y}_k(\tau) - \bar{Y}_{k-\tau}(\tau)\right)^2\right]. \tag{4.5.1}$$

Several estimators of the Allan variance, noted $\hat{\sigma}^2_{\bar{y}}(\tau)$, have been proposed. The asymptotically most efficient among them has been proposed in Greenhall [1991] and can be computed from a realization $\{y_k : k = 1, \ldots, N\}$ using

$$\hat{\sigma}^2_{\bar{y}}(\tau) = \frac{1}{2(N-2\tau+1)}\sum_{k=2\tau}^{N}\left(\bar{y}_k(\tau) - \bar{y}_{k-\tau}(\tau)\right)^2. \tag{4.5.2}$$

The Allan variance can be expressed in the frequency domain through the unique relationship between $\sigma^2_{\bar{Y}}(\tau)$ and the PSD $S_Y(\cdot)$ of the intrinsic processes El-Sheimy et al. [2008]:

$$\sigma^2_{\bar{Y}}(\tau) = 4\lim_{a\to\infty}\int_0^a S_Y(f)\frac{\sin^4(\pi f\tau)}{(\pi f\tau)^2}df \tag{4.5.3}$$

where f is the frequency. Eq. (4.5.3) links the parameter vector $\boldsymbol{\theta}$ to $\sigma^2_{\bar{Y}}(\tau)$. This relationship originates from the known form of the PSD function characterizing different noise processes which enables to express $\boldsymbol{\theta}$ as a function of $\sigma^2_{\bar{Y}}(\tau)$. A detailed discussion on how to express this link between $\boldsymbol{\theta}$ and $\sigma^2_{\bar{Y}}(\tau)$ can be found in Hou [2004]. Linear regions in the log-log "sigma-tau" plot, $\sigma_{\bar{Y}}(\tau)$ versus τ, are associated with regions of *power-law* behavior f^α in $S_Y(\cdot)$, i.e. with processes having a $S_Y(\cdot)$ of the form

$$S_Y(f) \propto |f|^\alpha \tag{4.5.4}$$

where α is a constant. In general, only five basic processes are considered with the Allan variance: quantization noise, white noise (also known as *angular random walk*), bias instability (also known as flicker noise), random walk (also known as *rate random walk*), and the random rate ramp. Under these considerations, $\boldsymbol{\theta}$ is usually estimated by performing linear regressions on (visually) identified linear regions in the Allan variance curve plotted using logarithmic scales. However, this method is only well defined for these few types of processes and it is not clear how inference on $\boldsymbol{\theta}$ can be made with this approach. Table 4.2 lists the theoretical values of $\sigma^2_{\bar{Y}}(\tau)$ together with the slopes of the associated Allan variance curve.

4.5.2 Hadamard Variance

The *Hadamard variance*, noted as σ_H^2, was proposed in Baugh [1971] as a generalization of the sample variance weighted with binomial coefficients:

$$\sigma_H^2 = \frac{1}{6}\mathbb{E}\left[\left(\bar{Y}_k(\tau) - 2\bar{Y}_{k-\tau}(\tau) + \bar{Y}_{k-2\tau}(\tau)\right)^2\right]. \tag{4.5.5}$$

Compared to the Allan variance, this approach has a higher spectral resolution and reduces the uncertainty of long-term estimates of the Allan variance without increasing the length of a data run. An estimator $\hat{\sigma}_H^2$ of the Hadamard variance can be found from a realization $\{y_k : k = 1, \ldots, N\}$ by replacing the theoretical by the empirical expectation:

$$\hat{\sigma}_H^2 = \frac{1}{6(N-3\tau+1)} \sum_{k=3\tau}^{N} \left(\bar{y}_k(\tau) - 2\bar{y}_{k-\tau}(\tau) + \bar{y}_{k-2\tau}(\tau)\right)^2. \tag{4.5.6}$$

A detailed description of the Hadamard variance can be found in Howe et al. [2005] which also introduced a modified version of this method.

4.5.3 Total Variance

The *total variance*, noted σ_T^2, was proposed in Howe [1995] as an estimator of the Allan variance that has lesser mean square error than the standard unbiased estimator of Eq. (4.5.2). An estimator of σ_T^2 is given by

$$\hat{\sigma}_T^2 = \frac{1}{N-1} \sum_{n=1}^{N-1} \left[\frac{1}{2(N-2\tau+1)} \sum_{k=2\tau}^{N} \left(\bar{y}_{k,n}(\tau) - \bar{y}_{k-\tau,n}(\tau)\right)^2 \right]$$

considering the realization $\{y_k : k = 1, \ldots, N\}$. Note that $\hat{\sigma}_T^2$ can be computationally very intensive. Consequently, several faster methods have been proposed in Greenhall et al. [1999], Percival [1997].

4.6 Power Spectral Density Method

The PSD analysis is a convenient method for analysing and characterizing signals, and for stochastic modeling Std 952-1997 [1998]. It is especially well suited for analysing periodic or aperiodic signals. Adapting Eq. (3.5.7) to a single-input, single-output linear system yields

$$S_Y(f) = |H(f)|^2 S_W(f). \tag{4.6.1}$$

Thus, for a sequence of white noise $\{W_k : k \in \mathbb{Z}\}$, the resulting PSD directly gives the system transfer function. Similarly to the Allan variance methods, noise coefficients of well defined processes, usually random walk, flicker noise, white noise and quantization noise, can be identified by linear regions in the f vs $S_Y(f)$ log-log curve. The right panel of Figure 4.4 depicts slopes of commonly used noise processes in a one-sided PSD curve. Details on the way how to estimate parameter values from the curve are given in Std 952-1997 [1998]. Note that the PSD method cannot distinguish between the random walk and the rate ramp which have both a slope of -2. Thus, the observed signal $\{y_k : k = 1, \ldots, N\}$ must be detrended prior to analysis (see Table 4.2).

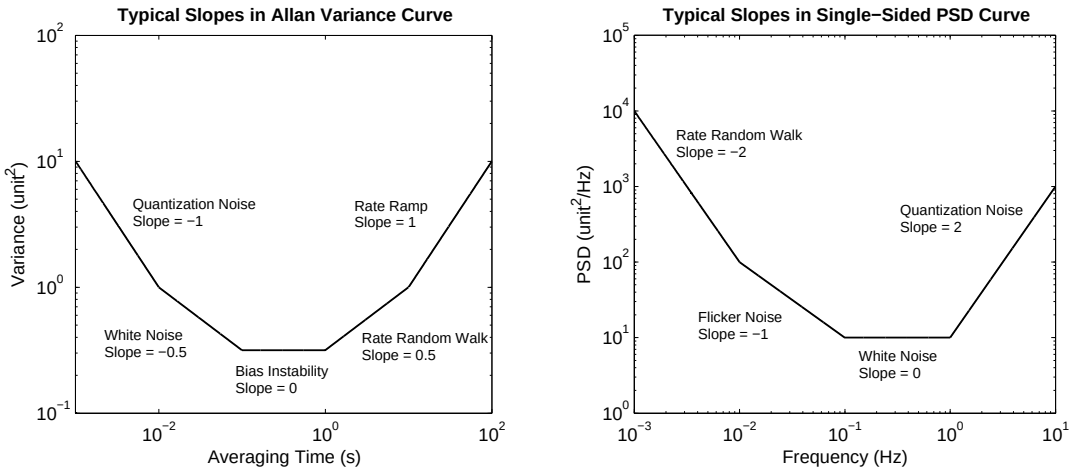

Figure 4.4: Typical slopes corresponding to commonly used noise models in Allan variance (left panel) and PSD (right panel) curves. The process parameters can be estimated by fitting lines over identified linear regions in the computed Allan variance/PSD sequence (using logarithmic scales).

Table 4.2: Slopes in PSD and Allan variance ("AV" in the Table) log-log curves together with Allan variance values of commonly used processes. *Note: Rate ramp removed by regression or by filtering* (adapted from Guerrier et al. [2013]).

Process	$\boldsymbol{\theta}$	**PSD Slope**	**AV Slope**	**AV Value**
WN	σ^2_{WN}	0	-0.5	$\sigma^2_{\hat{Y}}(\tau) = \frac{\sigma^2_{WN}}{\tau}$
RW	σ^2_{RW}	-2	0.5	$\sigma^2_{\hat{Y}}(\tau) = \frac{(2\tau^2+1)\sigma^2_{RW}}{6\tau} \approx \frac{\sigma^2_{RW}\tau}{3}$
GM	$\beta,\ \sigma^2_{GM}$	$[-2, 0]$	$[-0.5, 0.5]$	$\sigma^2_{\hat{Y}}(\tau) = \frac{\sigma^2_{GM}}{\beta^2\tau}\left[1 - \frac{1}{2\beta\tau}\left(3 - 4e^{-\tau\beta} + e^{-2\beta\tau}\right)\right]$
BI	$T_{BI},\ \sigma^2_{BI}$	-1	0	$\sigma^2_{\hat{Y}}(\tau) = \frac{2\sigma^2_{BI}\ln 2}{\pi}$
QN	Q	2	-1	$\sigma^2_{\hat{Y}}(\tau) = \frac{3Q^2}{\tau^2}$
RR	c_{RR}	See note	1	$\sigma^2_{\hat{Y}}(\tau) = \frac{c^2_{RR}\tau^2}{2}$

4.7 Time Series Analysis Method

An alternative to the state space representation for linear systems is given by

$$y_k = \sum_{j=1}^{p} a_j y_{k-j} + \sum_{j=1}^{q} b_j u_{k-j} + u_k \tag{4.7.1}$$

where u_k represents the normally distributed residual at time $t_k \in \mathbb{Z}$, and the sequences $\{a_j : j = 1, \ldots, p\}$ and $\{b_j : j = 1, \ldots, q\}$ are coefficients. If the generated $\{y_k\}$ is a stationary sequence, such a model is called an *Auto-Regressive Moving-Average* (ARMA) process of order p,q and will be noted ARMA(p,q) in this thesis. The way how to express an ARMA($\cdot$,$\cdot$) model in state space formulation can be found in many references such as Gelb [1974], Nassar et al. [2004]. Two special cases can be derived from model (4.7.1):

- The *Auto-Regressive* (AR) process of order p, noted AR(p), which assumes that

  $$y_k = \sum_{j=1}^{p} a_j y_{k-j} + u_k. \tag{4.7.2}$$

  The ACS of such a process can be shown to be Gelb [1974]

  $$\rho_{y,\tau} = \sum_{j=1}^{p} a_j \rho_{y,\tau-j} \quad \text{for} \quad \tau \in \mathbb{Z}. \tag{4.7.3}$$

  The way how to express an AR($\cdot$) model in state space formulation can be found in references like Gelb [1974], Nassar et al. [2004].

- The *Moving Average* (MA) process of order q, noted MA(q), which assumes that

  $$y_k = u_k - \sum_{j=1}^{q} b_j u_{k-j}. \tag{4.7.4}$$

  This model always produces a stationary process. Assuming that

  $$\mathbb{E}\left[u_k u_l\right] = \begin{cases} \sigma_u^2 & \text{if } \; l = k, \\ 0 & \text{if } \; l \neq k, \end{cases} \tag{4.7.5}$$

  then the corresponding ACS can be written as Gelb [1974]

  $$\rho_{y,\tau} = \begin{cases} \left(-b_k + \sum_{j=1}^{q-k} b_j b_{k+j}\right) \cdot \sigma_u^2 & \text{for } \; t_k \leq 0, \\ 0 & \text{for } \; t_k > 0 \end{cases} \tag{4.7.6}$$

for $\tau \in \mathbb{Z}$. Generally, the orders p and q are chosen using criterions such as the *Akaike Information Criterion* (AIC) or the *Bayesian Information Criterion* (BIC). More details about the AIC/BIC criterion will be given in Chapter 8. The coefficients $\{a_j\}$ and $\{b_j\}$ are estimated using the Box–Jenkins methodology Box et al. [2011], or the Yule-Walker, covariance, or Burg's methods in the case of pure AR($\cdot$) processes Nassar [2003].

Although authors like Nassar [2003] successfully applied AR($\cdot$) models to inertial sensors, we will not directly treat ARMA($\cdot, \cdot$) models and their associated estimation techniques in this thesis since, as we will discuss later in Chapter 7, they can be well approximated by sums of first-order Gauss-Markov processes.

4.8 Limitations

4.8.1 Autocorrelation Method

Estimating parameters of composite processes might become difficult when using classical calibration methods. Consider again the model (4.2.2) given as example in Section 4.2.2. Despite its appearent simplicity, the estimation task for such a model is non-trivial. The low-pass filter design required for isolating the correlated noise part when using the autocorrelation method is often problematic. This is especially true in cases of more complex composite models like sums of first-order Gauss-Markov processes.

4.8.2 PSD and Variance Methods

Although the Allan variance (and PSD) method is a well-established technique for identifying processes and estimating their parameters by performing linear regression of (visually) identified linear regions in such plots, it suffers from severe drawbacks:

- The Allan variance (and PSD) method works reasonably well only for processes which are clearly identifiable and separable in the spectral domain and not subject to spectral ambiguity Greenhall [1998]. However, it does not allow to directly read out the parameters of a Gauss-Markov process as large values of β make this process similar to Gaussian white noise, while small values of β approximate a random walk. Like the autocorrelation method, both the Allan variance and PSD do not allow the estimation of sums of Gauss-Markov processes.

- Inference (obtaining confidence intervals, tests, etc.) about the estimated parameters is in many cases impossible. Indeed, the system parameters are indirectly estimated through functions of coefficients estimated by linear regression (say $\hat{\boldsymbol{\beta}}$). The standard solution for deriving the (asymptotic) distribution of $\hat{\boldsymbol{\theta}}$ from the distribution $\hat{\boldsymbol{\beta}}$ is achieved through a first order approximation. In statistics, this approach is called the delta method (see Shao [2003] for details). In order to apply this method, it is required that the function $\boldsymbol{\beta} \mapsto \boldsymbol{\theta}$ is one-to-one and that $\hat{\boldsymbol{\beta}}$ is a consistent estimator. Unfortunately, this is generally not the case here. Consequently, deriving the (asymptotic) distribution of $\hat{\boldsymbol{\theta}}$ is not possible in general.

- The conventional Allan variance methodology is limited to models composed of processes characterized by linear regions in a "$\sigma_{\bar{y}}(\tau)$ v.s. τ" log-log plot and therefore this approach is far from being general.

In practice, models like those presented in Section 4.2.2 are estimated through *ad-hoc* tuning, by using available sensor specifications, or by experience Waegli and Skaloud [2007]. Recent investigations like Xing and Gebre-Egziabher [2008] have studied this type of models and proposed a methodology in which Gauss-Markov processes are used to overbound the sensor error. But the success of this methodology was quite limited in our experience. The next section describes another important drawback of the conventional Allan variance estimation method.

4.8.3 Consistency of the Allan Variance Method

The important concept of *consistency* is defined as follows.

Definition 4.8.1. *The consistency of an estimator can be defined as Shao [2003]*

$$\lim_{N\to\infty} P\left(||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||^2 \geq \epsilon\right) = 0, \;\; \forall \epsilon > 0. \tag{4.8.1}$$

where N is the size of the sample on which the estimation was performed. The practical signification of this limit is the following. Let $\hat{\boldsymbol{\theta}}_N$ be an estimate based on a sample of length N. Then, with an inconsistent estimator, there is no proof that

$$\mathbb{E}\left[||\hat{\boldsymbol{\theta}}_N - \boldsymbol{\theta}||^2\right] > \mathbb{E}\left[||\hat{\boldsymbol{\theta}}_{H\cdot N} - \boldsymbol{\theta}||^2\right] \quad for \quad H > 1 \tag{4.8.2}$$

and therefore the "quality" of the estimation may not increase with sample size N.

Consider a composite stochastic process $Y_k = \sum_m (Y_m)_k$ such that each $(Y_m)_k$ belongs to the set of models including the processes from the left panel of Figure 4.4 completed by any ARMA$(\cdot,\cdot)$ process. Assume further that $m \geq 2$ and that $\text{var}\left[(Y_m)_k\right] > 0, \forall m$. We are interested in estimating the parameters of the processes $(Y_m)_k$ belonging *exclusively* to the set of the left panel of Figure 4.4 using the Allan variance method. We show in Guerrier et al. that under these settings, the Allan variance estimation of the parameters of these processes is not consistent. As consistency is the most important property that an estimator should own, the application of the Allan variance is limited in such a case.

Assume for example that we observe a process $\{y_k : k = 1, \ldots, N\}$ which is driven from the following model

$$y_k = (y_{RW})_k + w_k. \tag{4.8.3}$$

Assume further that the process w_k is independent of $(y_{RW})_k$ and let

$$\boldsymbol{\theta} = \left\{\sigma^2_{WN}, \sigma^2_{RW}\right\} \tag{4.8.4}$$

be the set of parameters that specifies the model. The theoretical Allan variance of this system can be easily obtained from Table 4.2 and is given by

$$\sigma^2_{\bar{y}}(\tau) = \frac{6\sigma^2_{WN} + \left(2\tau^2 + 1\right)\sigma^2_{RW}}{6\tau} \tag{4.8.5}$$

whose logarithmic value $\log\left(\sigma^2_{\bar{y}}(\tau)\right)$ can be obtained in a typical Allan variance plot. However, the true $\log\left(\sigma^2_{\bar{y}}(\tau)\right)$ value is unknown in practice and only its estimated quantity $\log\left(\hat{\sigma}^2_{\bar{y}}(\tau)\right)$ is available. In this case, the Allan variance methodology would consist in applying a linear regression on the j first scales in order to estimate σ^2_{WN}, i.e

$$\log\left(\hat{\sigma}^2_{\bar{y}}(\tau)\right) = \log\left(\sigma^2_{\bar{y}}(\tau)\right) + \epsilon_\tau, \quad \tau = 1, ..., j. \tag{4.8.6}$$

This concept is illustrated in Figure 4.5 showing the Allan variance sequence of a simulated signal $\{y_k : k = 1, \ldots, N\}$ with $N = 10'000$ issued from Eq. (4.8.3). In this case, linear regression would be performed on the $j = 5$ first scales (dashed rectangle) for estimating σ^2_{WN}. Since we are interested in estimating σ^2_{WN}, $\log\left(\sigma^2_{\bar{y}}(\tau)\right)$ in Eq. (4.8.6) is replaced by

$$\log\left(\sigma^2_{\bar{y}}(\tau)\right) = \log\left(\frac{\sigma^2_{WN}}{\tau}\right) = \log\left(\sigma^2_{WN}\right) - \log\left(\tau\right). \tag{4.8.7}$$

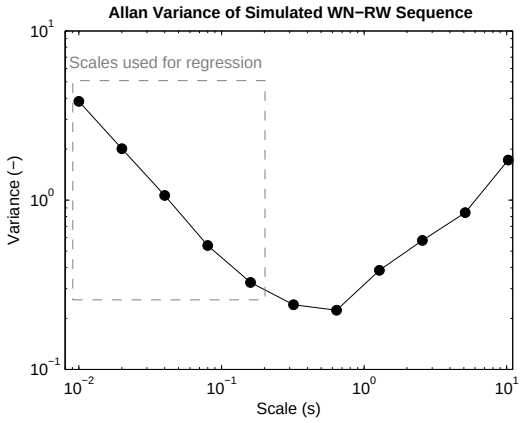

Figure 4.5: Estimated Allan variance sequence of a simulated signal issued from a sum of a white noise (with $\sigma^2_{WN} = 4$) and a random walk (with $\sigma^2_{RW} = 0.01$). Linear regression is performed on the $j = 5$ first scales (comprised withing the dashed rectangle) in order to estimate the white noise power σ^2_{WN}.

Considering Eq. (4.8.6) and (4.8.7) and assuming that ϵ_τ are errors assumed to be *iid* with expectation 0 and variance $\sigma^2_{\epsilon_\tau}$, we can construct an estimator for σ^2_{WN} as follows:

$$\hat{\sigma}^2_{WN} = \exp\left[\frac{1}{j}\sum_{i=1}^{j}\log\left(\hat{\sigma}^2_{\bar{y}}(i)\right) + \log(i)\right]. \tag{4.8.8}$$

Serroukh et al. [2000] have proven the consistency (in probability) of the so-called MODWT wavelet variance estimator (we will introduce it in Chapter 6) which implies the same property for the Allan variance estimator. Therefore, we can study the consistency of this estimator:

$$\begin{aligned}\operatorname*{plim}_{N\to\infty}\hat{\sigma}^2_{WN} &= \operatorname*{plim}_{N\to\infty}\exp\left[\frac{1}{j}\sum_{i=1}^{j}\log\left(\hat{\sigma}^2_{\bar{y}}(i)\right) + \log(i)\right] &(4.8.9)\\ &= \exp\left[\frac{1}{j}\sum_{i=1}^{j}\log\left(\sigma^2_{\bar{y}}(i)\right) + \log(i)\right] &(4.8.10)\end{aligned}$$

where the second equality is justified by Shao [2003] (see Theorem 1.10, p. 59) and "plim" is the probability limit operator denoting convergence in probability, i.e.

$$\operatorname*{plim}_{N\to\infty}\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}. \tag{4.8.11}$$

Substituting Eq. (4.8.5) into Eq. (4.8.10) yields

$$\hat{\sigma}^2_{WN} = \exp\left[\frac{1}{j}\sum_{i=1}^{j}\log\left(\frac{\sigma^2_{WN}}{i} + \frac{(2i+1)\sigma^2_{RW}}{6i}\right) + \log(i)\right]. \tag{4.8.12}$$

Now two cases can be investigated:

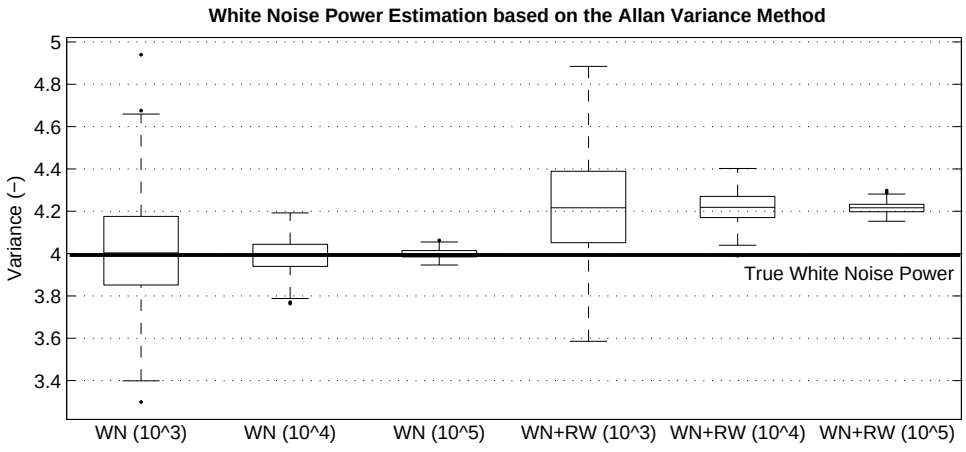

Figure 4.6: Inconsistency of the Allan variance method. Result of the Allan variance based estimation of σ^2_{WN} (true value depicted as a black horizontal line) when performed on 500 signals composed of a white noise only (WN, three first boxplots) and 500 signals composed of a white noise and a random walk (WN+RW, three next boxplots). Three signal lengths were used: 1000 (first and fourth boxplot), 10'000 (second and fifth boxplot) and 100'000 (third and sixth boxplot) samples. The bias in the $\hat{\sigma}^2_{WN}$ estimates is clearly visible on the right part.

1. If $\sigma^2_{RW} = 0$, we have from Eq. (4.8.12) that

$$\operatorname*{plim}_{N\to\infty} \hat{\sigma}^2_{WN} = \sigma^2_{WN} \tag{4.8.13}$$

   which demonstrates that $\hat{\sigma}^2_{WN}$ is a consistent estimator of σ^2_{WN} *if and only if*[2] $\sigma^2_{RW} = 0$.

2. If $\sigma^2_{RW} > 0$, then $\frac{(2i+1)\sigma^2_{RW}}{6i} > 0$ in Eq. (4.8.12) and we have

$$\operatorname*{plim}_{N\to\infty} \hat{\sigma}^2_{WN} = \sigma^2_{WN} + c, \quad \text{with} \quad c > 0. \tag{4.8.14}$$

   This last inequality clearly proves that the conventional Allan variance methodology does not provide a consistent estimator for σ^2_{WN} when $\sigma^2_{RW} > 0$.

This important result can be confirmed by further developing the example shown in Figure 4.5. We simulated 500 signals $\{y_k : k = 1, \ldots, N\}$ under two models: the first is a Gaussian white noise process

$$y_k = w_k \tag{4.8.15}$$

with $\sigma^2_{WN} = 4$, while the second is a sum of a Gaussian white noise and a random walk, i.e.

$$y_k = (y_{RW})_k + w_k \tag{4.8.16}$$

[2]The *if* statement comes from Eq. (4.8.13) while the *only if* statement is a consequence of the uniqueness of $\boldsymbol{\theta}$ in Eq. (4.8.4).

with $\sigma^2_{RW} = 0.01$. Three values for N were used, namely $N = 1'000$, $N = 10'000$ and $N = 100'000$. According to Figure 4.5, we computed the Allan variance sequence and performed the linear regression on the $j = 5$ first scales for each of the simulated signal $\{y_k\}$ in order to estimate $\hat{\sigma}^2_{WN}$. The results of the estimation are presented in the boxplots of Figure 4.6. The true value of the white noise variance is drawn as a thick black horizontal line. The first three boxplots show the 500 $\hat{\sigma}^2_{WN}$ values when estimated on the $j = 5$ first scales of the Allan variance sequence computed on signals issued from Eq. (4.8.15). This was done for the three values of N. In accordance with Eq. (4.8.13), the estimates are clearly unbiased in all three cases. However, the three next plots depict again the values of $\hat{\sigma}^2_{WN}$, but this time they were estimated on the $j = 5$ first scales of the Allan variance based on signals issued from Eq. (4.8.16). There is a clear bias for all three values of N. This confirms the result obtained in Eq. (4.8.14) claiming that the Allan variance is an inconsistent estimator when several processes are present.

Chapter 5

Likelihood-based Modeling Approach

5.1 Introduction

Estimating the matrices belonging to the discrete state space model parameters defined in Eq. (1.6.6) and repeated here:

$$\boldsymbol{\theta} = \{\boldsymbol{\Phi}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}, \mathbf{H}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_0, \mathbf{P}_0\},$$

is in general quite challenging. The books Harvey [1991] and Durbin and Koopman [2012] contain extensive accounts of state space models and their applications. For linear and/or Gaussian state space models, the *maximum likelihood estimator* (MLE) is a natural choice for the estimation of $\boldsymbol{\theta}$. Generally, the (log) likelihood function, noted

$$\ell(\boldsymbol{\theta}|\mathbf{y}_k, \mathbf{x}_k) = \log L(\boldsymbol{\theta}|\mathbf{y}_k, \mathbf{x}_k) \tag{5.1.1}$$

for $\{t_k : k = 1, \ldots, N\}$, $\mathbf{x}_k$ as defined in Section 1.6, and $\mathbf{y}_k$ the $(l \times 1)$ measurement vector[1], is a highly nonlinear and complicated function Shumway and Stoffer [2000]. Historically, the Newton-Raphson algorithm was employed to successively update $\boldsymbol{\theta}$ until the log likelihood was maximized Gupta and Mehra [1974]. However, a conceptually simpler estimation procedure was proposed in Shumway and Stoffer [1982] which relies on the *Expectation-Maximization* (EM) algorithm originally developed in Dempster et al. [1977]. The EM is a procedure that is guaranteed to converge to the MLE and is therefore often used when dealing with difficult likelihood maximization. It is based on the idea of replacing a complex likelihood maximization by a sequence of easier maximizations whose limit is the answer to the original problem Casella and Berger [2002].

The EM approach is particularly suited to problems with missing data which often render calculations cumbersome. With this approach, two different likelihood problems are considered. The first is the problem we are interested to solve, namely the *incomplete-data* problem, while the second is the problem that is actually solved by the EM, the so-called *complete-data* problem. Suppose for the moment that we could observe the state vector $\mathbf{x}_k$ in addition to the observation sequence $\mathbf{y}_k$ at digital times $\{t_k : k = 1, \ldots, N\}$. Then, the quantity $\ell(\boldsymbol{\theta}|\mathbf{y}_k, \mathbf{x}_k)$

[1] In this chapter, we use the notation $\mathbf{y}$ instead of $\mathbf{z}$ (see Section 1.6) to denote the measurement vector, since it is the commonly used denomination in the statistics literature.

could be easily maximized using results from the multivariate normal theory Shumway and Stoffer [2000]. This maximization can be considered as the complete-data problem. However, we do not have the complete data since the state vector $\mathbf{x}_k$ is unobserved. Thus, we aim to solve the incomplete-data problem. In this context, the EM algorithm offers an iterative method for finding the MLE $\hat{\boldsymbol{\theta}}$ by successively maximizing the conditional expectation of the complete data likelihood. A more formal treatment of this method is given in the next section.

In the context of integrated navigation, the EM algorithm has mainly been used for obtaining system process noise $\mathbf{Q}$ and measurement noise $\mathbf{R}$ in which all coefficients are estimated Huang et al. [2007], Einicke et al. [2009]. However, when using the EM for the more complex state space models describing stochastic error behaviour of inertial sensors, some coefficients in $\mathbf{Q}$ and $\mathbf{R}$ must remain fixed. An example can be provided by the stochastic error model which is composed of white noise, a first-order Gauss-Markov process, a random walk and a random rate ramp process where elements of the state space transition matrix $\mathbf{F}$ must remain fixed (e.g. the one coefficient of the random walk, and the null off-diagonal coefficients), and some others can be freely estimated (e.g. the β parameter of the Gauss-Markov process).

In Section 5.2 we show how to adapt the classical EM algorithm, referred to as the *unconstrained EM*, to constrain the state space model parameters. This adapted version will be referred to as the *constrained EM* algorithm. Constraints allow estimating more complex stochastic error models such as those typically used in navigation filters (see Section 3.6). In Section 5.3, we do a critical analysis of its practical use in inertial navigation. In Section 5.4, we simulate realistic examples for illustrating the algorithm performance. We then compare the results obtained by the EM algorithms with the benchmark methods, i.e. the Allan variance, Hadamard variance and total variance methods described in Chapter 4. Finally, an example with real data is included in Section 5.5 to demonstrate the positive impact of the derived stochastic model on the filtered trajectory. Identified limitations of the EM-based method are discussed in Section 5.6.

5.2 Expectation-Maximization Algorithm

5.2.1 The Likelihood Function

The log-likelihood of $\boldsymbol{\theta}$ given $\mathbf{y}_k$ and $\mathbf{x}_k$ for $k = 1, \ldots, N$ is

$$\begin{aligned}\ell(\boldsymbol{\theta}|\mathbf{y}_k, \mathbf{x}_k) = &-\frac{1}{2}\left(\mathbf{x}_1 - \boldsymbol{\mu}_0\right)^T \mathbf{P}_0^{-1}\left(\mathbf{x}_1 - \boldsymbol{\mu}_0\right) \\ &- \frac{1}{2}\sum_{k=2}^{N}\left(\mathbf{x}_k - \boldsymbol{\Phi}\mathbf{x}_{k-1} - \mathbf{u}_k\right)^T \mathbf{Q}^{-1}\left(\mathbf{x}_k - \boldsymbol{\Phi}\mathbf{x}_{k-1} - \mathbf{u}_k\right) \\ &- \frac{1}{2}\sum_{k=1}^{N}\left(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\right)^T \mathbf{R}^{-1}\left(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\right) - \frac{1}{2}\log|\mathbf{P}_0| \\ &- \frac{1}{2}N\log|\mathbf{Q}| - \frac{1}{2}N\log|\mathbf{R}| - \frac{N}{2}\log 2\pi.\end{aligned} \tag{5.2.1}$$

The objective of the EM algorithm is to find $\boldsymbol{\theta}$ that maximizes $\ell(\boldsymbol{\theta}|\mathbf{y}_k, \mathbf{x}_k)$. As $\mathbf{x}_k$ is unobservable in our case, it is replaced by the complete-data likelihood $\boldsymbol{\Psi} = \mathbb{E}\left[\ell(\boldsymbol{\theta}|\mathbf{y}_k, \mathbf{x}_k)\right]$ whose entire expression can be found in Holmes [2010], Shumway and Stoffer [1982].

5.2.2 The Algorithm

The general principle of the EM algorithm is shown in Figure 5.1. The EM algorithm switches iteratively between an *Expectation (E-)step* and a *Maximization (M-)step* McLachlan and Krishnan [1997]. On the $(j+1)$th iteration, the E- and M-steps are defined as follows:

E-Step Calculate $Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j)}\right)$, where

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j)}\right) = \mathbb{E}\left[\ell(\boldsymbol{\theta}^{(j)}|\mathbf{y}_k, \mathbf{x}_k)\right] = \boldsymbol{\Psi}^{(j)} \tag{5.2.2}$$

M-Step Choose $\boldsymbol{\theta}^{(j+1)}$ to be any value of $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ which belongs to:

$$\boldsymbol{\theta}^{(j+1)} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}}\, Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j)}\right) \tag{5.2.3}$$

The E- and M-steps are iteratively repeated until some convergence criterion is fulfilled. An example of convergence test could be

$$\left|L\left(\boldsymbol{\theta}^{(j+1)}|\mathbf{y}_k, \mathbf{x}_k\right) - L\left(\boldsymbol{\theta}^{(j)}|\mathbf{y}_k, \mathbf{x}_k\right)\right| < \epsilon \tag{5.2.4}$$

for some arbitrarily small ϵ McLachlan and Krishnan [1997].

In the E-step, the expected states, noted $\mathbf{x}_k^N$, (as well as the associated covariance matrices, noted $\mathbf{P}_k^N$) are computed such that they could be considered fixed in the M-step where $\boldsymbol{\Psi}^{(j)}$ will be maximized. Therefore, the E-step requires computing:

$$\mathbf{x}_k^N = \mathbb{E}\left[\mathbf{x}_k|\mathbf{y}_1, \ldots, \mathbf{y}_N, \boldsymbol{\theta}^{(j)}\right] \tag{5.2.5}$$

$$\mathbf{P}_k^N = \operatorname{cov}\left[\mathbf{x}_k|\mathbf{y}_1, \ldots, \mathbf{y}_N, \boldsymbol{\theta}^{(j)}\right] \tag{5.2.6}$$

$$\mathbf{P}_{k,k-1}^N = \operatorname{cov}\left[\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{y}_1, \ldots, \mathbf{y}_N, \boldsymbol{\theta}^{(j)}\right] \tag{5.2.7}$$

which can be calculated using a Kalman smoother.

In the M-step, the parameter vector is updated to $\boldsymbol{\theta}^{(j+1)}$ by finding the parameters that maximize $\boldsymbol{\Psi}^{(j)}$ considering the values $\mathbf{x}_k^N$, $\mathbf{P}_k^N$ and $\mathbf{P}_{k,k-1}^N$ obtained in the E-step as fixed. For doing that, the expression yielded by $\boldsymbol{\Psi}^{(j)}$ is minimized by computing the partial derivatives with respect to $\boldsymbol{\theta}^{(j)}$ and setting them to zero. The results of these derivatives for the classical unconstrained case can be found in many articles like Holmes [2010], Shumway and Stoffer [1982]. The work in Holmes [2010], Wu et al. [1996] provides the way of constraining coefficients in the matrices belonging to the set $\boldsymbol{\theta}$. Let $\mathbf{M}$ be any matrix contained in the set $\boldsymbol{\theta}$ with fixed and p free elements (*i.e.* to be estimated). This matrix can be decomposed in matrices containing the fixed and the free elements:

$$\mathbf{M} = \mathbf{M}_{fixed} + \mathbf{M}_{free} \tag{5.2.8}$$

Let the vec$(\cdot)$ operator create a column vector from $\mathbf{M}$ by stacking the column vectors of

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_{\bullet,1} & \mathbf{m}_{\bullet,2} & \mathbf{m}_{\bullet,3} \end{bmatrix} \tag{5.2.9}$$

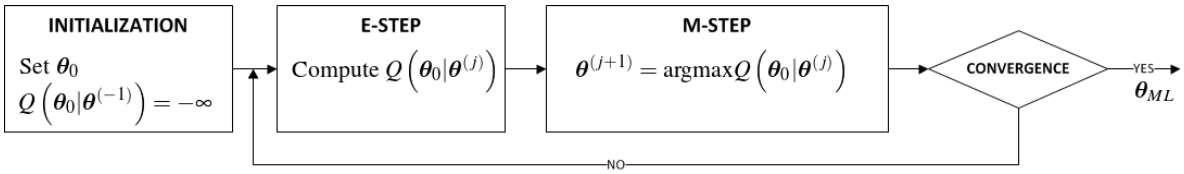

Figure 5.1: General principle of the EM algorithm.

below one another:

$$\text{vec}(\mathbf{M}) = \begin{bmatrix} \mathbf{m}_{\bullet,1} \\ \mathbf{m}_{\bullet,2} \\ \mathbf{m}_{\bullet,3} \end{bmatrix}. \tag{5.2.10}$$

Eq. (5.2.10) can be rewritten as the following linear combination:

$$\text{vec}(\mathbf{M}) = \mathbf{f} + \mathbf{D} \cdot \mathbf{m} \tag{5.2.11}$$

where $\mathbf{f} = \text{vec}(\mathbf{M}_{fixed})$ and $\mathbf{D} \cdot \mathbf{m} = \text{vec}(\mathbf{M}_{free})$ with $\mathbf{m}$ a $(p \times 1)$ vector containing the p free values, and $\mathbf{D}$ a design matrix transforming $\mathbf{m}$ into $\text{vec}(\mathbf{M}_{free})$. To derive the update equations, the likelihood has to be rewritten as a function of $\text{vec}(\mathbf{M})$ where $\mathbf{M}$ can be any parameter matrix for which the update equation is derived Holmes [2010]. Then, this result is rewritten as a function of $\mathbf{m}$ using Eq. (5.2.11). The $\mathbf{m}$ vector is deduced by setting $\partial\boldsymbol{\Psi}/\partial\mathbf{m} = 0$. Since detailed derivations can be found in Holmes [2010], only the final results are given here. If we set the following quantities:

$$\widetilde{\mathbf{P}}_k = \mathbf{P}_k^N + \mathbf{x}_k^N(\mathbf{x}_k^N)^T \tag{5.2.12}$$

$$\widetilde{\mathbf{P}}_{k,k-1} = \mathbf{P}_{k,k-1}^N + \mathbf{x}_k^N(\mathbf{x}_{k-1}^N)^T \tag{5.2.13}$$

we can write the update equations for estimating some individual parameters of $\boldsymbol{\theta}$ contained in Eq. (1.6.6) like:

- $\mathbf{u}$ update (unknown parameter $\boldsymbol{\Lambda}$ considered inside $\mathbf{u}$ in Eq. (1.6.4)):

$$\mathbf{u}^{(j+1)} = \mathbf{f}_\mathbf{u} + \mathbf{D}_\mathbf{u} \cdot \mathbf{m}_\mathbf{u} \tag{5.2.14}$$

with

$$\mathbf{m}_\mathbf{u} = \frac{1}{N-1}\left(\mathbf{D}_\mathbf{u}^T\mathbf{Q}^{-1}\mathbf{D}_\mathbf{u}\right)^{-1}\mathbf{D}_\mathbf{u}^T\mathbf{Q}^{-1}\sum_{k=2}^{N}\left(\mathbf{x}_k^N - \boldsymbol{\Phi}\mathbf{x}_{k-1}^N - \mathbf{f}_\mathbf{u}\right) \tag{5.2.15}$$

- $\boldsymbol{\mu}_0$ update (unknown parameter in Eq. (1.6.4)):

$$\boldsymbol{\mu}_0^{(j+1)} = \mathbf{f}_{\boldsymbol{\mu}_0} + \mathbf{D}_{\boldsymbol{\mu}_0} \cdot \mathbf{m}_{\boldsymbol{\mu}_0} \tag{5.2.16}$$

with

$$\mathbf{m}_{\boldsymbol{\mu}_0} = \left(\mathbf{D}_{\boldsymbol{\mu}_0}^T(\mathbf{P}_0)^{-1}\mathbf{D}_{\boldsymbol{\mu}_0}\right)^{-1}\mathbf{D}_{\boldsymbol{\mu}_0}^T(\mathbf{P}_0)^{-1}(\mathbf{x}_1^N - \mathbf{f}_{\boldsymbol{\mu}_0})$$

- $\mathbf{\Phi}$ update (unknown parameter in Eq. (1.6.4)):

$$\text{vec}\left(\mathbf{\Phi}^{(j+1)}\right) = \mathbf{f}_{\mathbf{\Phi}} + \mathbf{D}_{\mathbf{\Phi}} \cdot \mathbf{m}_{\mathbf{\Phi}} \tag{5.2.17}$$

with

$$\mathbf{m}_{\mathbf{\Phi}} = \left(\sum_{k=2}^{N} \mathbf{D}_{\mathbf{\Phi}}^T (\widetilde{\mathbf{P}}_{k-1} \otimes \mathbf{Q}^{-1}) \mathbf{D}_{\mathbf{\Phi}}\right)^{-1} \mathbf{D}_{\mathbf{\Phi}}^T \left(\sum_{k=2}^{N} \left[\text{vec}(\mathbf{Q}^{-1}\widetilde{\mathbf{P}}_{k,k-1}) - (\widetilde{\mathbf{P}}_{k-1} \otimes \mathbf{Q}^{-1})\mathbf{f}_{\mathbf{\Phi}} - \text{vec}(\mathbf{Q}^{-1}\mathbf{u}(\mathbf{x}_{k-1}^N)^T)\right]\right)$$

- $\mathbf{H}$ update (unknown parameter in Eq. (1.6.4)):

$$\text{vec}\left(\mathbf{H}^{(j+1)}\right) = \mathbf{f}_{\mathbf{H}} + \mathbf{D}_{\mathbf{H}} \cdot \mathbf{m}_{\mathbf{H}} \tag{5.2.18}$$

with

$$\mathbf{m}_{\mathbf{H}} = \left(\sum_{k=1}^{N} \mathbf{D}_{\mathbf{H}}^T (\widetilde{\mathbf{P}}_k \otimes \mathbf{R}^{-1}) \mathbf{D}_{\mathbf{H}}\right)^{-1} \mathbf{D}_{\mathbf{H}}^T \left(\sum_{k=1}^{N} \left[\text{vec}\left(\mathbf{R}^{-1}\mathbf{y}_k(\mathbf{x}_k^N)^T\right) - (\widetilde{\mathbf{P}}_k \otimes \mathbf{R}^{-1})\mathbf{f}_{\mathbf{H}}\right]\right) \tag{5.2.19}$$

- $\mathbf{Q}$ update (unknown parameter in Eq. (1.6.4)):

$$\text{vec}\left(\mathbf{Q}^{(j+1)}\right) = \mathbf{f}_{\mathbf{Q}} + \mathbf{D}_{\mathbf{Q}} \cdot \mathbf{m}_{\mathbf{Q}} \tag{5.2.20}$$

with

$$\mathbf{m}_{\mathbf{Q}} = \frac{1}{N-1} \left(\mathbf{D}_{\mathbf{Q}}^T \mathbf{D}_{\mathbf{Q}}\right)^{-1} \mathbf{D}_{\mathbf{Q}}^T \text{vec}(\mathbf{S}) \tag{5.2.21}$$

and

$$\mathbf{S} = \sum_{k=2}^{N} \Big(\widetilde{\mathbf{P}}_k - \widetilde{\mathbf{P}}_{k,k-1}\mathbf{\Phi}^T - \mathbf{\Phi}\widetilde{\mathbf{P}}_{k-1,k} - \mathbf{x}_k^N\mathbf{u}^T - \mathbf{u}(\mathbf{x}_k^N)^T + \mathbf{\Phi}\widetilde{\mathbf{P}}_{k-1}\mathbf{\Phi}^T + \mathbf{\Phi}\mathbf{x}_{k-1}^N\mathbf{u}^T\mathbf{\Phi}^T + \mathbf{u}(\mathbf{x}_{k-1}^N)^T\mathbf{\Phi}^T + \mathbf{u}\mathbf{u}^T\Big)$$

- $\mathbf{R}$ update (unknown parameter in Eq. (1.6.4)):

$$\text{vec}\left(\mathbf{R}^{(j+1)}\right) = \mathbf{f}_{\mathbf{R}} + \mathbf{R}_{\mathbf{R}} \cdot \mathbf{m}_{\mathbf{R}} \tag{5.2.22}$$

with

$$\mathbf{m}_{\mathbf{R}} = \frac{1}{N} \left(\mathbf{D}_{\mathbf{R}}^T \mathbf{D}_{\mathbf{R}}\right)^{-1} \mathbf{D}_{\mathbf{R}}^T \text{vec}\left(\sum_{k=1}^{N} (\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^N)(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^N)^T + \mathbf{H}\mathbf{P}_k^N\mathbf{H}^T\right)$$

The $\otimes$ symbol stands for the Kronecker product in this chapter. Note that as stated in Harvey [1991], Holmes [2010], simultaneous estimation of $\boldsymbol{\mu}_0$ and $\mathbf{P}_0$ makes the algorithm fail in practice. Thus, as proposed in Harvey [1991], we kept $\mathbf{P}_0$ fixed at a small value. It is also possible to remove $-\frac{1}{2}\left(\mathbf{x}_1 - \boldsymbol{\mu}_0\right)^T \mathbf{P}_0^{-1} \left(\mathbf{x}_1 - \boldsymbol{\mu}_0\right)$ from Eq. (5.2.1).

5.2.3 Asymptotic Distribution of the MLEs

In the context of state space models, the MLE $\hat{\boldsymbol{\theta}}$ is consistent and has an asymptotic normal distribution given by

$$\sqrt{N}\left(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}\right) \underset{N\to\infty}{\xrightarrow{\mathcal{D}}} \mathcal{N}\left(\mathbf{0}, \boldsymbol{J}\left(\boldsymbol{\theta}\right)^{-1}\right) \tag{5.2.23}$$

where $\boldsymbol{J}\left(\boldsymbol{\theta}\right)$ is the asymptotic information matrix defined as

$$\boldsymbol{J}\left(\boldsymbol{\theta}\right) = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left[-\frac{\partial^2 \ell(\boldsymbol{\theta}|\mathbf{y}_k)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T}\right]. \tag{5.2.24}$$

Moreover, the following theorem taken from Casella and Berger [2002] states that the EM estimate, denoted as $\hat{\boldsymbol{\theta}}_{EM}$, tends to the MLE, denoted as $\hat{\boldsymbol{\theta}}_{ML}$.

Theorem 1. *The sequence* $\{\hat{\boldsymbol{\theta}}^{(j)}\}$ *satisfies*

$$L\left(\hat{\boldsymbol{\theta}}^{(j+1)}|\mathbf{y}\right) \geq L\left(\hat{\boldsymbol{\theta}}^{(j)}|\mathbf{y}\right) \tag{5.2.25}$$

with equality holding if and only if successive iterations yield the same value of the maximized expected complete-data log likelihood, that is,

$$\mathbb{E}\left[\ell\left(\hat{\boldsymbol{\theta}}^{(j+1)}|\mathbf{y},\mathbf{x}\right)\middle|\hat{\boldsymbol{\theta}}^{(j)},\mathbf{y}\right] = \mathbb{E}\left[\ell\left(\hat{\boldsymbol{\theta}}^{(j)}|\mathbf{y},\mathbf{x}\right)\middle|\hat{\boldsymbol{\theta}}^{(j)},\mathbf{y}\right]. \tag{5.2.26}$$

There are mainly three conditions for the results stated above to hold and can be found in Caines [1987]. A full treatment of the necessary conditions for the consistency and the asymptotic normality of MLEs can also be found in Caines [1987]. Note that in practice, these conditions hold for the set of models considered in this thesis. In this research, we only provide some crucial elements to establish these results, similarly as it is done in Shumway and Stoffer [2000]. Indeed, it is necessary to assume that the absolute eigenvalues of the transition matrix $\boldsymbol{\Phi}$ are less than one. This assumption guarantees that the filter is stable. Moreover, the state space model must be observable and controllable to ensure that the results given in Eq. (5.2.23) and (5.2.24) hold.

5.3 Practical Issues

In general, the maximization of the likelihood of state space models is a difficult task due to the highly nonlinear form of this function. Therefore, the use of the EM algorithm implies three main issues:

1. The EM algorithm is a "hill-climbing" algorithm and can therefore converge to a local maximum. This makes the method sensitive to initial conditions. In practice, diffusion priors could be employed to perform some kind of pre-search of the parameter space. Alternatively, the Allan, Hadamard and total variance methods could be used to obtain an initial guess of some of the parameter values.

2. In numerical applications of the EM algorithm, setting a proper stopping criterion is crucial. The work of Dempster et al. [1977] showed that the EM algorithm is guaranteed to converge to at least a local maximum. Thus, the convergence criterion mentioned in Section 5.2.2 is in theory correct. In practice however, this is not feasible due to numerical imprecision and the large number of iterations that is often required to reach the maximum. A typical convergence criterion could be given by:

$$\frac{L\left(\boldsymbol{\theta}^{(j+1)}|\mathbf{y}_k, \mathbf{x}_k\right) - L\left(\boldsymbol{\theta}^{(j)}|\mathbf{y}_k, \mathbf{x}_k\right)}{0.5 \cdot |L\left(\boldsymbol{\theta}^{(j+1)}|\mathbf{y}_k, \mathbf{x}_k\right) + L\left(\boldsymbol{\theta}^{(j)}|\mathbf{y}_k, \mathbf{x}_k\right)| + c} < \epsilon$$

   where the averaging in the denominator increases the statibility of the criterion, and the c value is used to keep the criterion well behaved in the case where a fixed point is reached.

3. Joint estimation of $\mathbf{\Phi}$, $\mathbf{Q}$ and $\mathbf{u}$ can sometimes lead to instabilities. In such situations, the EM algorithm is very likely to diverge. For example, this situation could arise from ridges in the likelihood surface of $\mathbf{\Phi}$ vs. $\mathbf{u}$ and $\mathbf{\Phi}$ vs. $\mathbf{Q}$. In practice, it has been found that eliminating the estimation of the $\mathbf{u}$ parameter in the EM by using classical least-squares estimation largely improves the performance of the EM algorithm. However, this renders the inference on $\boldsymbol{\theta}$ improper since $\mathbf{u}$ is fixed (and thus viewed as true), which may bias the estimation of the remaining parameters.

5.4 Performance Study through Simulations

Several simulation scenarios are performed to analyse the potential of employing the classical and constrained EM algorithm for estimating parameters of composite stochastic processes.

Simulation 5.4.1. Let $\{y_k : k = 1, \ldots, N\}$ be an observed signal from a process driven from model $F(\boldsymbol{\theta})$ which is a sum of a random walk, white noise, and a random rate ramp (designated WN-RW-RR in the sequel):

$$Y_k = (Y_{RW})_k + (Y_{RR})_k + W_k, \qquad k \in \mathbb{Z}$$

where W_k, $(Y_{RW})_k$ and $(Y_{RR})_k$ are defined in Table 3.1. A corresponding discrete time-invariant state space model can be constructed as

$$\begin{aligned} x_k &= x_{k-1} + w_k + (c_{RR}\Delta t) \\ y_k &= x_k + v_k \end{aligned}$$

such that $w_k \sim \mathcal{N}(0, q)$ with $q = \sigma^2_{RW}$ and $v_k \sim \mathcal{N}(0, r)$ with $r = \sigma^2_{WN}$ The task of the EM is to estimate the following parameter set:

$$\boldsymbol{\theta} = \{\sigma^2_{WN}, \sigma^2_{RW}, c_{RR}\}.$$

This specific problem does not require any constraints on the state space parameters. Thus, the $\mathbf{M}_{fixed}$ matrices will be null, which leads to the classical unconstrained EM algorithm of Shumway and Stoffer [1982], and enables a complete comparison with the Allan variance

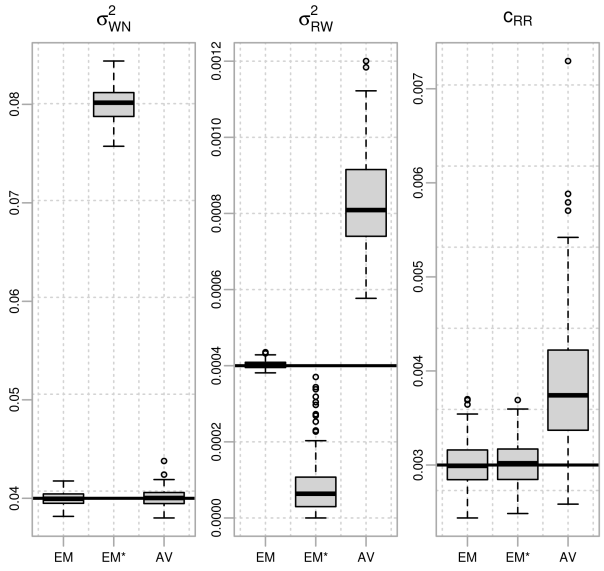

Figure 5.2: Performance comparison between the EM algorithm started at good (EM) and a bad ($EM^\star$) initial values, and the Allan variance technique (AV) for 200 simulated signals issued from a sum of a white noise, a random walk and a rate ramp process. The true parameters are marked by horizontal lines (adapted from Stebler et al. [2011a]).

since all processes are clearly identifiable (see the left panel of Figure 4.4). We simulate 200 synthetic signals $\{y_k\}$ of length $N = 6000$ issued from the following true parameter set:

$$\boldsymbol{\theta} = \{0.04, 4 \cdot 10^{-4}, 0.003\}$$

To highlight the importance of the initialization, the EM algorithm was started at two different initial values:

$$\begin{aligned} \boldsymbol{\theta}^{(0)} &= \{0.25, 10^{-8}, 0.0\} \\ \boldsymbol{\theta}^{\star(0)} &= \{25, 100, 10\} \end{aligned}$$

where the second set $\boldsymbol{\theta}^{\star(0)}$ contains values which are far away from $\boldsymbol{\theta}$. The EM estimates are compared to what the Allan variance technique would provide by fitting lines on the linear regions of the log-log plot which correspond to the white noise (slope is $-1/2$), random walk (slope is $1/2$) and random rate ramp (slope is 1) processes. Comparison of estimation is shown in Figure 5.2 where the EM and $EM^\star$ columns correspond to the EM results when started at $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\theta}^{\star(0)}$, respectively, and the horizontal lines represent the true parameter set $\boldsymbol{\theta}$.

In the first case (EM), the solution clearly converged to the global maximum of the likelihood function. The c_{RR} parameter was also correctly estimated since it is not dependent on the estimation of $\boldsymbol{\Phi}$ (which is fixed). In the second case ($EM^\star$), the estimation of σ^2_{WN} and σ^2_{RW} (defining q and r) is clearly affected by a convergence to a wrong local maximum. The performance of the Allan variance method for estimating σ^2_{WN} is relatively similar to the EM algorithm. However, the random walk and random rate ramp are often not well separable using

Table 5.1: RMSE and relative RMSE (R-RMSE) of the EM, EM* and Allan variance estimators for 200 simulated processes of size $N = 6000$ from model WN-RW-RR (adapted from Stebler et al. [2012]).

	EM		**EM***		**Allan Variance**	
	RMSE	**R-RMSE**	**RMSE**	**R-RMSE**	**RMSE**	**R-RMSE**
σ^2_{WN}	$7.67 \cdot 10^{-4}$	$1.92 \cdot 10^{-2}$	$4.02 \cdot 10^{-2}$	1.00	$9.09 \cdot 10^{-4}$	$2.27 \cdot 10^{-2}$
σ^2_{RW}	$1.17 \cdot 10^{-5}$	$2.93 \cdot 10^{-2}$	$3.24 \cdot 10^{-4}$	$8.09 \cdot 10^{-1}$	$4.45 \cdot 10^{-4}$	1.11
c_{RR}	$2.52 \cdot 10^{-4}$	$8.4 \cdot 10^{-2}$	$2.37 \cdot 10^{-4}$	$7.88 \cdot 10^{-2}$	$1.18 \cdot 10^{-3}$	$3.94 \cdot 10^{-1}$

this method, because the fitted line slopes do not perfectly correspond to the correct values for the respective processes.

The *Root-Mean-Square Error* (RMSE) as well as the *relative RMSE* (R-RMSE) of the three estimation techniques used in this simulation are listed in Table 5.1. The *EM* approach clearly yields the best results. Note that in Stebler et al. [2011a], we performed an additional comparison with the Hadamard and total variances. In general, the total variance had a smaller variance than the Allan and Hadamard variances. But the latter had a smaller bias than the two other variance-based methods.

Simulation 5.4.2. Let $\{y_k : k = 1, \ldots, N\}$ be an observed signal from a process driven from model $F(\boldsymbol{\theta})$ which is a sum of white noise, a first-order Gauss-Markov process, and a random rate ramp (designated as WN-GM-RR in the sequel):

$$Y_k = (Y_{GM})_k + (Y_{RR})_k + W_k, \qquad k \in \mathbb{Z}$$

where W_k, $(Y_{GM})_k$ and $(Y_{RR})_k$ are defined in Table 3.1. The corresponding discrete time-invariant state space model is (to the first order)

$$\begin{aligned} x_k &= (1 - \beta\Delta t)\, x_{k-1} + w_k + (c_{RR}\Delta t) \\ y_k &= x_k + v_k \end{aligned}$$

such that $w_k \sim \mathcal{N}(0, q)$ with $q = 2\beta\sigma^2_{GM}$ and $v_k \sim \mathcal{N}(0, r)$ with $r = \sigma^2_{WN}$. The task of the EM is to estimate the parameter set

$$\boldsymbol{\theta} = \{\beta, \sigma^2_{GM}, \sigma^2_{WN}, c_{RR}\}$$

defining $\boldsymbol{\Phi}, \mathbf{u}, \mathbf{Q}$ and $\mathbf{R}$ from the sequence $\{y_k\}$. Like in Simulation 5.4.1, this problem does not require any constraint on the state space model parameters, but this time, the $\boldsymbol{\Phi}$ matrix including the β value has to be estimated. This makes the global maximum search task in the likelihood "surface" more difficult.

The EM algorithm was applied on 200 realizations of $\{y_k\}$ with $N = 6000$ issued from the following parameters:

$$\boldsymbol{\theta} = \{0.008, 0.25, 0.64, 10^{-3}\}.$$

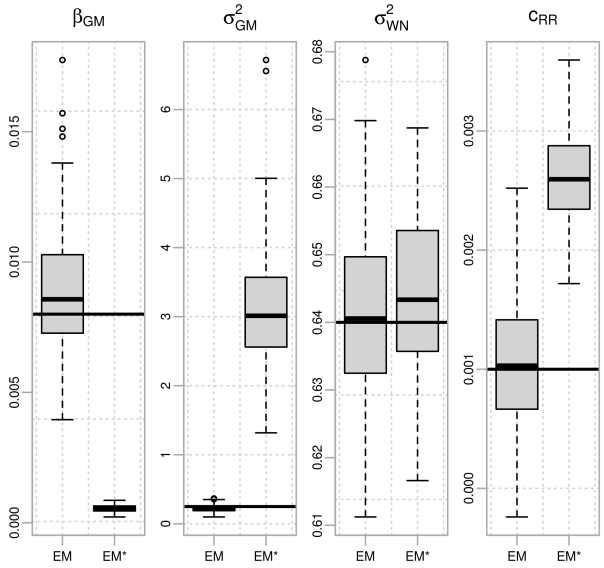

Figure 5.3: Performance comparison between the EM algorithm with prior estimation of c_{RR} by ordinary least-squares adjustment (EM) and without ($EM^\star$) for 200 simulated signals issued from a sum of a white noise, a Gauss-Markov and a rate ramp process. The true values of the parameters are marked by horizontal lines (adapted from Stebler et al. [2011a]).

The algorithm was started at the following initial parameter values:

$$\boldsymbol{\theta}^{(0)} = \{10^{-3}, 1.0, 1.0, 0.0\}.$$

To highlight the advantage of eliminating some parameters by other estimation techniques, the EM algorithm was runned with estimating $\mathbf{u}$ (denoted $EM^\star$), and with signal detrending through least-squares adjustment prior to EM estimation (denoted EM). The results are shown in Figure 5.3. They are much better if $\mathbf{u}$ is correctly eliminated from the EM estimation using data detrending, since the $\boldsymbol{\Phi}$ and $\mathbf{Q}$ updates depend on $\mathbf{u}$. The RMSE and R-RMSE values are listed in Table 5.2 from which it can be deduced that the EM approach performed the best.

We now study the restitution of Allan variance plots by the parameter set $\hat{\boldsymbol{\theta}}$ estimated via EM. We selected randomly three from the 200 estimated parameters $\hat{\boldsymbol{\theta}}$ for a case where $\mathbf{u}$ was

Table 5.2: RMSE and relative RMSE (R-RMSE) of the EM, and EM* estimators for 200 simulated processes of size $N = 6000$ from model WN-GM-RR (adapted from Stebler et al. [2012]).

	EM		**EM***	
	RMSE	**R-RMSE**	**RMSE**	**R-RMSE**
β_{GM}	$2.44 \cdot 10^{-3}$	$3.05 \cdot 10^{-1}$	$7.47 \cdot 10^{-3}$	$9.33 \cdot 10^{-1}$
σ^2_{GM}	$5.53 \cdot 10^{-2}$	$2.21 \cdot 10^{-1}$	3.02	$1.21 \cdot 10^{1}$
σ^2_{WN}	$1.26 \cdot 10^{-2}$	$1.97 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$2.03 \cdot 10^{-2}$
c_{RR}	$5.49 \cdot 10^{-4}$	$5.49 \cdot 10^{-1}$	$1.68 \cdot 10^{-3}$	1.68

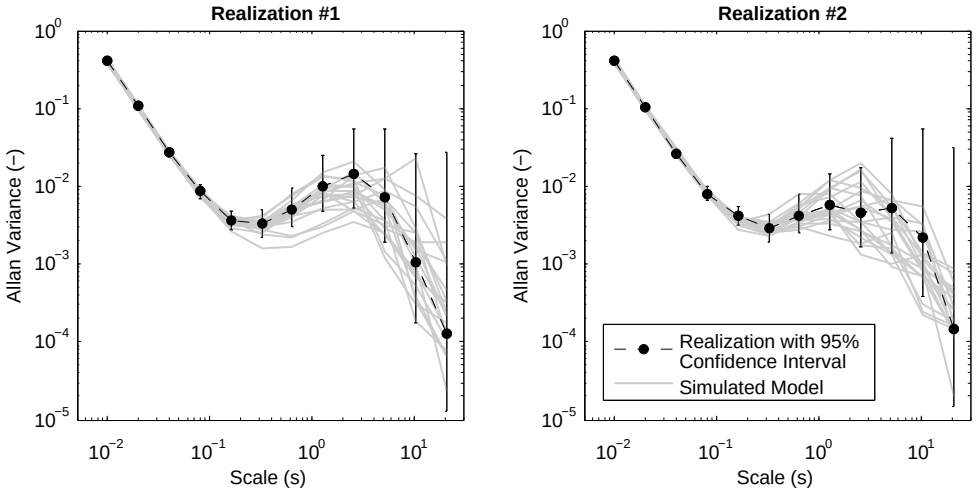

Figure 5.4: EM algorithm results for simulated signals containing a white noise, a Gauss-Markov and a random ramp process. Each panel shows the Allan variance of one realization issued from $F(\boldsymbol{\theta})$ (black curves) and 20 simulations driven from the corresponding $F(\hat{\boldsymbol{\theta}})$ (gray curves) model (adapted from Stebler et al. [2011a]).

removed by least-squares adjustment. We then computed the Allan variance for 20 realizations issued from these two solutions $\hat{\boldsymbol{\theta}}$ (thin curves in Figure 5.4) and compared them to the respective two true signals $\{y_k\}$ (thick curves in Figure 5.4). It can be seen that the resulting Allan variance sequences are fairly well contained in the 95% confidence interval associated with Allan variance estimation of the true signals.

Simulation 5.4.3. Let $\{y_k : k = 1, \ldots, N\}$ be an observed signal from a process driven from model $F(\boldsymbol{\theta})$ which is a sum of white noise, a first-order Gauss-Markov process, a random walk and a random rate ramp (designated as WN-GM-RW-RR in the sequel):

$$Y_k = (Y_{GM})_k + (Y_{RW})_k + (Y_{RR})_k + W_k, \qquad k \in \mathbb{Z}$$

where W_k, $(Y_{GM})_k$, $(Y_{RW})_k$ and $(Y_{RR})_k$ are defined in Table 3.1. This is modeled as a discrete time-invariant state space model of the form (to the first order):

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} 1 - \beta\Delta t & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_k + \begin{bmatrix} 0 \\ c_{RR}\Delta t \end{bmatrix} \\ y_k &= \begin{bmatrix} 1 & 1 \end{bmatrix} \mathbf{x}_k + v_k \end{aligned}$$

such that $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ with

$$\mathbf{Q} = \begin{bmatrix} 2\beta\sigma_{GM}^2 & 0 \\ 0 & \sigma_{RW}^2 \end{bmatrix}$$

and $v_k \sim \mathcal{N}(0, r)$ with $r = \sigma_{WN}^2$. The goal is to estimate the parameter set

$$\boldsymbol{\theta} = \{\beta, \sigma_{GM}^2, \sigma_{RW}^2, \sigma_{WN}^2, c_{RR}\}$$

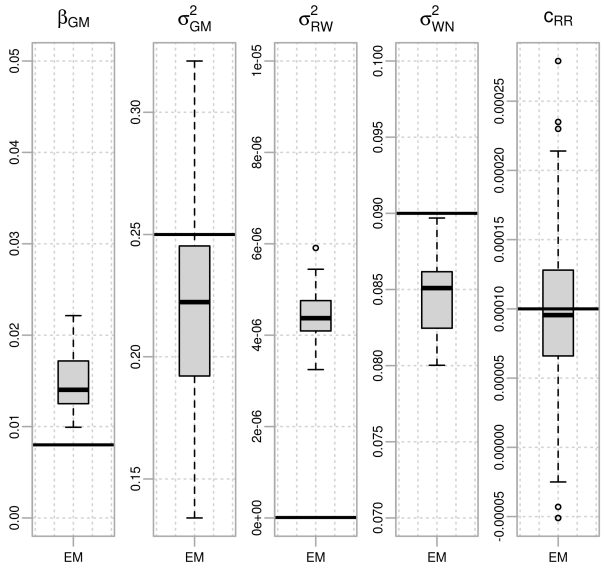

Figure 5.5: Performance the EM algorithm for 200 simulated signals issued from a sum of a white noise, a Gauss-Markov, a random walk and a random ramp process. The true values of the parameters are marked by horizontal lines (adapted from Stebler et al. [2011a]).

from the observed $\{y_k\}$. Such a problem typically requires that some coefficients in the involved matrices must remain fixed while others must be estimated. For example, all coefficients in $\mathbf{\Phi}$ excepting $1 - \beta\Delta t$ must remain fixed. In $\mathbf{u}$, the first coefficient must stay null, while only the diagonal of the $\mathbf{Q}$ matrix contains free elements. Since all the coefficients in $\mathbf{H}$ are fixed, this matrix has not to be updated. We illustrate the performance of the EM by the same procedure as for the previous simulation scenario. The signal $\{y_k\}$ is issued from the following parameters:

$$\boldsymbol{\theta} = \{0.008, 0.25, 10^{-8}, 0.09, 10^{-4}\}.$$

The initial parameters where set to

$$\boldsymbol{\theta}^{(0)} = \{10^{-4}, 10^{-6}, 10^{-10}, 2.5 \cdot 10^{-7}, 0\}.$$

The results of the 200 runs are shown in Figure 5.5. Note that $\mathbf{u}$ has been estimated by least-squares adjustement for improving the estimation of the remaining parameters in the EM. The estimation appears to be seriously biased, specially for the inverse correlation time of the Gauss-Markov process and for the random walk driving noise strength which are difficult to separate in the spectral space. As for the previous scenario, we computed the Allan variance for 20 realizations issued from two solutions $\hat{\boldsymbol{\theta}}$ (see Figure 5.6). The effect of the bias in some parameters is visible through the systematic overbounding in the middle part of the Allan variance sequences.

5.5 Application on Real Data Set

Three hours long data were collected from a static tactical-grade IMU (*IMAR-FSAS* iMAR GmbH [2009]) sampling at a frequency of 400 Hz in constant temperature conditions. In this

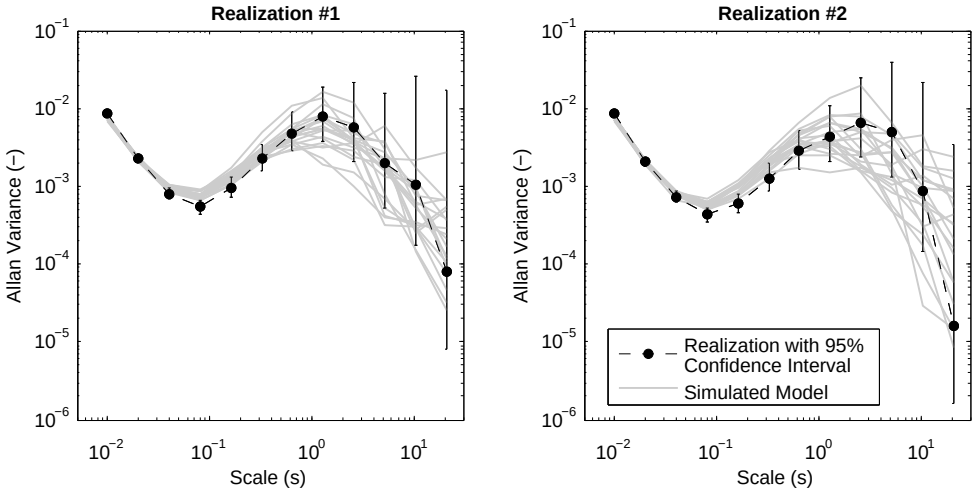

Figure 5.6: Constrained EM algorithm results for simulated signals containing a white noise, a Gauss-Markov, a random walk and a random ramp process. Each panel shows the Allan variance of one realization issued from $F(\boldsymbol{\theta})$ (black curves) and 20 simulations driven from the corresponding $F(\hat{\boldsymbol{\theta}})$ (gray curves) model (adapted from Stebler et al. [2011a]).

application case, we will loop through the filter design loop described in Figure 4.1 in which we use the EM algorithm at the model estimation stage.

5.5.1 Model Building

The Allan variance plots revealed that the gyroscope error signals are mainly composed of a white noise and thus present no need for more sophisticated modeling. However, the Allan variance plot of accelerometer errors (black curve in Figure 5.7 for the Y-axis accelerometer) shows a more complex structure[2]. Note that the analyses are similar for the X- and Z-axis sensors and are therefore not shown here. Since the slopes of the linear parts in the thick Allan variance curve do not correspond to any of the theoretical processes depicted in the left panel of Figure 4.4, we choose to model this error by superposing two first-order Gauss-Markov processes and a white noise. Such a model can be written as

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} 1-\beta_1\Delta t & 0 \\ 0 & 1-\beta_2\Delta t \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_k \\ y_k &= \begin{bmatrix} 1 & 1 \end{bmatrix} \mathbf{x}_k + v_k \end{aligned} \tag{5.5.1}$$

such that $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ with

$$\mathbf{Q} = \begin{bmatrix} 2\beta_1\sigma^2_{GM,1} & 0 \\ 0 & 2\beta_2\sigma^2_{GM,2} \end{bmatrix} \tag{5.5.2}$$

and $v_k \sim \mathcal{N}(0, r)$ with $r = \sigma^2_{WN}$.

[2]The estimation was computed on a signal downsampled to 100 Hz for limiting computational load.

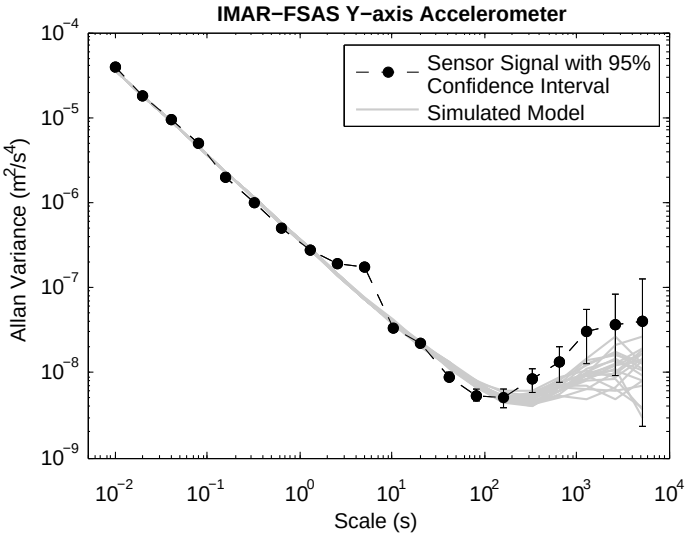

Figure 5.7: Results of the estimation of two Gauss-Markov processes and a white noise applied on the *IMAR-FSAS* Y-axis accelerometer error signal. The figure shows the Allan variance of 20 realizations issued from $\hat{\boldsymbol{\theta}}$ (thin curves) and the Allan variance of the sensor signal (thick curve) (adapted from Stebler et al. [2011a]).

5.5.2 Model Estimation

The goal is to estimate the parameter set

$$\boldsymbol{\theta} = \{\beta_1, \beta_2, \sigma^2_{GM,1}, \sigma^2_{GM,2}, \sigma^2_{WN}\}$$

from the signal $\{y_k\}$. For that, we applied the EM algorithm on the previously built model and obtained the following values:

$$\hat{\boldsymbol{\theta}} = \{0.0004, 0.10, 4 \cdot 10^{-8}, 10^{-8}, 3.6 \cdot 10^{-5}\}$$

where the units of the β and variances are $[1/s]$ and $[(m/s^2)^2]$, respectively.

5.5.3 Model Validation

Validation at Observation Level

The quality of the estimation is illustrated in Figure 5.7 in which Allan variance plots of 20 realizations issued from the estimated $\hat{\boldsymbol{\theta}}$ (thin curves) are compared with the one issued from the sensor signal (thick curve). The estimated white noise power level appears to fit the real signal (left part of the Allan variance curve). However, long-term errors modeled by the two Gauss-Markov processes match the signal's Allan variance sequence only approximately (right part of the Allan variance curve). This can be explained by several reasons which highlight the limitations of the constrained EM method on inertial sensors. First, the task of identifying the Gauss-Markov parameters within a process containing much higher power of white noise is difficult and induces very long convergence time. Second, accumulation of numerical imprecision

in many iterations may influence the results if the parameters are of small magnitude (which is the case here). Third, using longer time series would most likely improve the uncertainty of the parameter estimation. However, this was not feasible due to memory limitations of the computing hardware. Indeed, increasing the length of the analysed signal may improve the observation of the underlying long-term processes (right part in the Allan variance plot). In other words, since the 95% confidence intervals in this region will be decreased, the estimation of the Gauss-Markov process parameters by the EM algorithm will be improved.

Validation at State Level

In the sequel, we analyse the impact of the estimated model on the INS/GNSS integration via optimal forward Kalman filtering and backward smoothing. For that, the *IMAR-FSAS* IMU was mounted together with a high-grade dual-frequency GPS receiver (*JAVAD Delta*) on a car, and the motion was sampled at 100 Hz and 10 Hz, respectively. The carrier-phase GPS observation were double-differenced in post-processing to yield high-precision (centimeter-level) GPS positioning using a base GPS receiver (*Topcon Hiper Pro*) sampling at 10 Hz. These have been combined with the inertial observation in the EKF. To highlight the impact of proper stochastic modeling, we introduced artificially two outages in GPS solutions of different duration, at times where good and reliable GPS solutions were available as reference. During these outages, the navigation solution is solely dependent on inertial navigation, meaning that the residual systematic errors affecting these signals are integrated with time. We then recomputed the INS/GPS trajectory using the traditional *IMAR-FSAS* stochastic error model provided by the manufacturer (velocity random walk $< 50\mu g/\sqrt{\text{Hz}}$)[3], and compared it to the EKF/smoothed solution using the EM-estimated accelerometer model (the same gyroscope error model was used for both scenarios). In both cases we compare the positioning differences with respect to the reference. This allows to compute the positioning error along each direction in the l-frame (NED axes) by comparing both solutions with the reference trajectory (the one without gap).

The first 20 seconds long outage has been introduced in a time during which the car was turning in a roundabout. Figure 5.8 depicts the processed position differences along each axis when using the traditional *IMAR-FSAS* model (full curves) and the new model (dotted curves). Except for the East component, the new model significantly decreased the trajectory errors based on inertial coasting during this period.

The second outage was longer (about 130 seconds) and affected a period in which the car was moving on a straight road and its acceleration varied. As shown in Figure 5.8, the filtered trajectory errors were better bounded at the end of this outage (by a factor of two to four) when using the new model. Indeed, the maximum observed difference could be decreased from 23 meters to 10 meters along the North component, from -6 meters to -1.2 meters along the East component, and from 16 meters to 4.2 meters along the vertical component.

Finally, the following two additional remarks can be made. First, a close view on $\hat{\boldsymbol{\theta}}$ reveals that the Gaussian white noise (i.e. the velocity random walk) is dominant. Second, converting our estimated $\hat{\sigma}^2_{WN}$ to the same units as the ones used by the manufacturer yields $\hat{\sigma}^2_{WN} = 30\ \mu g/\sqrt{\text{Hz}}$ (by considering a sampling frequency of 400 Hz). This result is in accordance with the manufacturer's specification for velocity random walk.

[3]We refer to iMAR GmbH [2009] for the complete *IMAR-FSAS* IMU specifications.

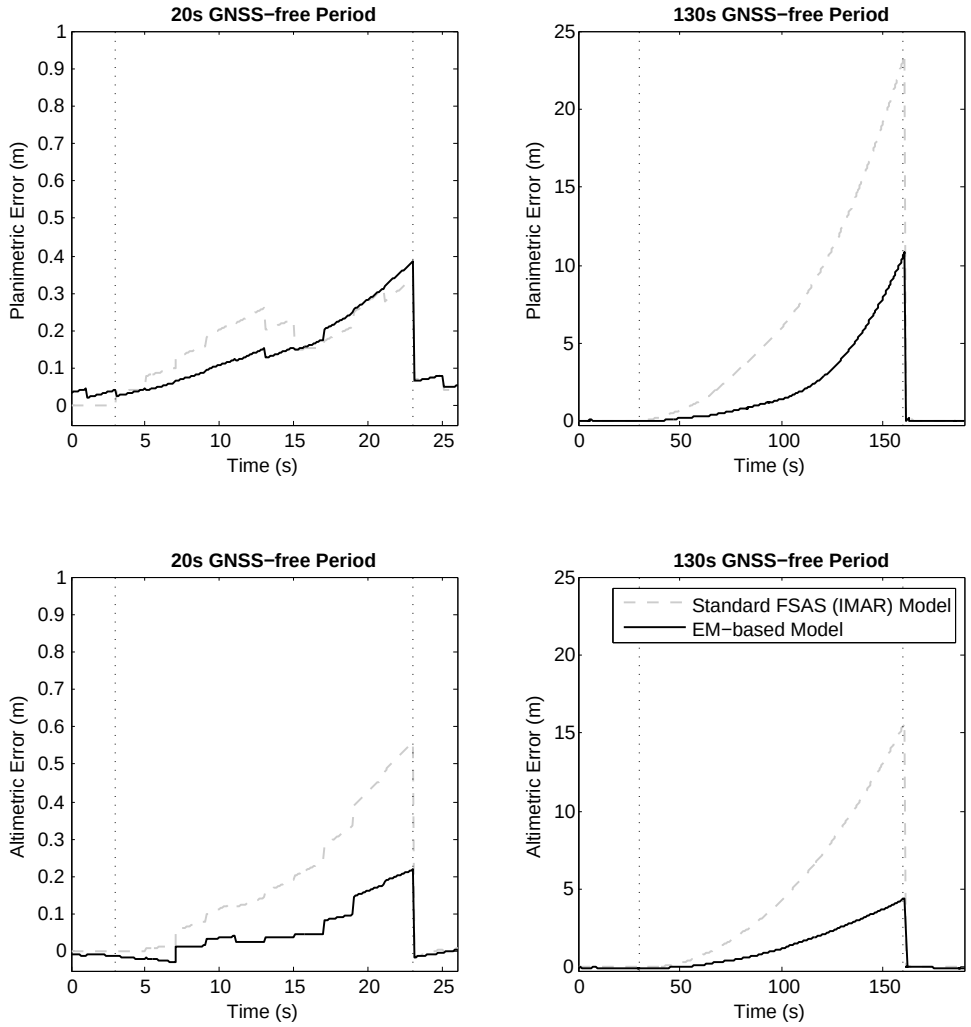

Figure 5.8: Performance of the *IMAR-FSAS* accelerometer error model estimated by employing the EM algorithm. The planimetric and altimetric errors are shown in the upper two and lower two panels, respectively. The left and right columns correspond to the 20 seconds-long and 130 seconds-long GPS outage period, respectively. The performance of the EKF when based on the manufacturer's error model is represented by dotted gray curves, and by full black curves when based on the model estimated by the EM algorithm (adapted from Stebler et al. [2011a]).

5.6 Limitations

Although we demonstrated that an EM algorithm can successfully estimate parameters of models for which the traditional Allan variance technique cannot be used, it only worked if preceded by some a priori data processing like data detrending. Moreover, we also showed that the EM method may converge to a local maximum if the initial values are “far" from the true parameter values. Therefore, using diffusion priors for the search of parameter initial values is recommended when applying this method in practice. The performed studies also revealed an important limit of the proposed method. The maximization step can be very complex and finding the MLE is not always a simple task. Moreover, the task becomes even more challenging when the observed process size is large and the model is far more sophisticated. Note also that, in contrast to the variance based methods, the EM based approach enables computing confidence intervals for the estimated parameters. Nevertheless, as shown in the lastly simulated example and the real application, the likelihood-based approach is still not capable of estimating noise with very complex error structures.

The next chapter introduces a new estimation framework which can be applied on models for which the Allan variance and MLE based estimators provide poor performance in practice.

Chapter 6

Generalized Method of Wavelet Moments

6.1 Introduction

In this chapter, we propose and evaluate a completely new framework, called the *Generalized Method of Wavelet Moments* (GMWM), for estimating parameters of L independent stochastic processes. Although the applicability of this work is general and therefore goes far beyond integrated navigation, we stay in the context of modeling inertial sensors from which it originated. The presented content is based on the work we published in Guerrier et al. [2013], Stebler et al. [2012] as a new estimation method for the parameter vector $\boldsymbol{\theta}$ that is based on matching the empirical and model-based wavelet variances. We demonstrated the consistency of our estimator in Guerrier et al. [2013] (see Theorem 1) together with the non-trivial conditions that must be satisfied for the estimator to be consistent. In particular, we showed in Theorem 2 that for a model made of the sum of L independent processes, the consistency is satisfied. This implies that for composite processes based on models considered in this thesis (i.e. Gaussian white noise, random ramp, quantization noise, random walk and $k < \infty$ autoregressive models of order 1), our estimator is consistent.

We begin by introducing the notion of wavelet variance in Section 6.2. Then we derive our estimator in Section 6.3 where we also prove its consistency. Similarly to the previous chapter, we validate the estimator first using simulation studies in Section 6.4. The way how to implement the algorithm will be discussed in Section 6.5 before concluding in Section 6.6. Note that no real case scenario is studied in this chapter, since Chapter 7 will be entirely dedicated to this task.

6.2 The Wavelet Variance

As pointed out by Percival and Guttorp [1994], the wavelet variance can be interpreted as the variance of a process after it has been subject to an approximate bandpass filter. The wavelet variance can be built using wavelet coefficients issued from a modified *Discrete Wavelet Transform* (DWT)Mallat [1999], Percival and Walden [2000] called the *Maximal Overlap DWT*

(MODWT) Greenhall [1991], Percival and Walden [2000]. The wavelet coefficients are built using wavelet filters $\{\tilde{h}_{j,l} : j = 1, \ldots, J\}$ which for $j = 1$ and for the MODWT satisfy

$$\sum_{l=0}^{L_1-1} \tilde{h}_{1,l} = 0, \sum_{l=0}^{L_1-1} \tilde{h}_{1,l}^2 = \frac{1}{2} \text{ and } \sum_{l=-\infty}^{\infty} \tilde{h}_{1,l}\tilde{h}_{1,l+2m} = 0 \tag{6.2.1}$$

where $\tilde{h}_{1,l} = 0$ for $l < 0$ and $l \geq L_1$, L_1 is the length of $\tilde{h}_{1,l}$, m is a nonzero integer. Considering the transfer function of $\tilde{h}_{1,l}$ as

$$\widetilde{H}_1(f) = \sum_{l=0}^{L_1-1} \tilde{h}_{1,l} e^{-i2\pi fl}, \tag{6.2.2}$$

the jth level wavelet filters $\{\tilde{h}_{j,l}\}$ of length $L_j = (2^j - 1)(L_1 - 1) + 1$ can be obtained by computing the inverse discrete Fourier Transform of

$$\widetilde{H}_j(f) = \widetilde{H}_1(2^{j-1}f) \prod_{l=0}^{j-2} e^{i2\pi 2^l f(L_1-1)} \widetilde{H}_1(\tfrac{1}{2} - 2^l f). \tag{6.2.3}$$

The MODWT filter is actually a rescaled version of the DWT filter $h_{j,l}$, i.e. $\tilde{h}_{j,l} = h_{j,l}/2^{j/2}$. Filtering an infinite sequence $\{y_k; k \in \mathbb{Z}\}$ using the wavelet filters $\{\tilde{h}_{j,k}\}$ yields the MODWT wavelet coefficients

$$\overline{W}_{j,k} = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} y_{k-l}, \; k \in \mathbb{Z}. \tag{6.2.4}$$

We define the wavelet variance at dyadic scales $\tau_j = 2^{j-1}$, as the variances of the $\{\overline{W}_{j,k}\}$ sequences, i.e.

$$\nu^2(\tau_j) = \text{var}\left[\overline{W}_{j,k}\right]. \tag{6.2.5}$$

Note that the wavelet variances are assumed not to depend on time. The condition for this property to hold is that the integration order d for the series $\{y_k\}$ to be stationary is such that $d \leq L_1/2$ and $\{\tilde{h}_{j,l}\}$ is based on a *Daubechies wavelet filter* (see Daubechies [1992] and Percival and Walden [2000], Chapter 8). This is due to the fact that Daubechies wavelet filters of width L_1 contain an embedded backward difference filter of order $L_1/2$. In such a case, the series of wavelet coefficients $\{\overline{W}_{j,k}\}$ is stationary with PSD $S_{W_j}(f) = |\widetilde{H}_j(f)|^2 S_{F_{\boldsymbol{\theta}}}(f)$, where $|\cdot|$ is denoting the modulus, and $F_{\boldsymbol{\theta}} = F(\boldsymbol{\theta})$. This means that the variance of wavelet coefficient series is equal to the integral of the coefficients' PSD Serroukh et al. [2000], i.e.

$$\nu^2(\tau_j) = \int_{-1/2}^{1/2} S_{W_j}(f)df = \int_{-1/2}^{1/2} |\widetilde{H}_j(f)|^2 S_{F_{\boldsymbol{\theta}}}(f)df. \tag{6.2.6}$$

Hence, there is an implicit link between the wavelet variances and the parameters of the data generating model $F(\boldsymbol{\theta})$. We exploit this connection when defining an estimator for $\boldsymbol{\theta}$, namely by matching a sample estimate of the wavelet variances, $\nu^2(\tau_j)$, together with the model-based expression of the wavelet variance given by the left handside of Eq. (6.2.6). For

wavelet variances based on Haar wavelet filters (see Eq. (6.2.16) below) and for the Gaussian white noise, random walk, rate ramp, quantization noise, AR(1), and ARMA(1,1) models, the integral in Eq. (6.2.6) is solved and given in Table 6.1, based on the results of Zhang [2008]. Wavelet variances for other models can be computed using the same methodology[1].

For a finite (observed) process $\{y_k : k = 1, \ldots, N\}$, the MODWT wavelet variance estimator given by

$$\hat{\nu}^2(\tau_j) = \frac{1}{M_j} \sum_{k=L_j}^{N} W_{j,k}^2 \tag{6.2.7}$$

with $W_{j,k} = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} y_{k-l}, k \in (L_j; N)$ and $M_j = N - L_j + 1$, is a consistent estimator for $\nu^2(\tau_j)$. With this respect, Serroukh et al. [2000] show that under suitable conditions, $\sqrt{M_j}\left(\hat{\nu}^2(\tau_j) - \nu^2(\tau_j)\right)$ is asymptotically normal with mean 0 and variance

$$S_{W_j}(0) = 2\int_{-1/2}^{1/2} S_{W_j}^2(f)df = 2\int_{-1/2}^{1/2} |\overline{H}_j(f)|^4 S_{F_{\boldsymbol{\theta}}}^2(f)df. \tag{6.2.8}$$

Eq. (6.2.8) can be estimated by means of

$$\widehat{S}_{W_j}(0) = \sum_{\tau=-M_j}^{M_j} \left[\frac{1}{M_j} \sum_{k=L_j}^{N} W_{j,k} W_{j,k+|\tau|}\right]^2 \tag{6.2.9}$$

and the asymptotic properties of Eq. (6.2.9) are given in Percival and Walden [2000], page 312. These results were extended to the multivariate case in Guerrier et al. [2013] were we demonstrated that under some regularity conditions, the asymptotic distribution of $\hat{\boldsymbol{\nu}}^2$ is given by

$$\sqrt{N}\left(\hat{\boldsymbol{\nu}}^2 - \mathbb{E}\{\boldsymbol{\nu}^2\}\right) \xrightarrow[N\to\infty]{\mathcal{D}} \mathcal{N}\left(0, \mathbf{V}_{\hat{\boldsymbol{\nu}}^2}\right) \tag{6.2.10}$$

where

$$\boldsymbol{\nu}^2 = \left[\nu^2(\tau_j)\right]_{j=1,\ldots,J} \quad \text{and} \quad \hat{\boldsymbol{\nu}}^2 = \left[\hat{\nu}^2(\tau_j)\right]_{j=1,\ldots,J}. \tag{6.2.11}$$

The matrix $\mathbf{V}_{\hat{\boldsymbol{\nu}}^2}$ has size $(J \times J)$ and is given by

$$\mathbf{V}_{\hat{\boldsymbol{\nu}}^2} = \begin{bmatrix} \sigma_{1,1}^2 & \cdots & \sigma_{1,J}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{J,1}^2 & \cdots & \sigma_{J,J}^2 \end{bmatrix}. \tag{6.2.12}$$

The elements of $\mathbf{V}_{\hat{\boldsymbol{\nu}}^2}$ can be obtained through

$$\sigma_{ml}^2 = 2\pi S_{W_m W_l}(0), \quad \text{for} \quad m, l = 1, \ldots, J \tag{6.2.13}$$

[1]However, as it will be shown only later, the possibility of expressing these error models analyticaly in the space of wavelet variances is not a necessary prerequisite for applying the new estimator.

Table 6.1: Analytical PSD and wavelet variance implied by white noise, random walk, random rate ramp, quantization noise, AR(1) and ARMA(1,1) processes. We use τ for τ_j. Note that the first-order Gauss-Markov process is a special case of AR(1) process (adapted from Guerrier et al. [2013]).

Process	**Abbr.**	**Parameter(s)**	$S_{F_{\boldsymbol{\theta}}}(f)$	$\nu^2(\tau)$ **(Haar)**	**Note**
White Noise	WN	σ^2_{WN}	$\frac{\sigma^2_{WN}}{2\pi}$	$\frac{\sigma^2_{WN}}{2\tau}$	$\sigma^2_{WN} > 0$
Random Walk	RW	σ^2_{RW}	$\frac{\sigma^2_{RW}}{8\pi \sin^2\left(\frac{f}{2}\right)}$	$\frac{\left(2\tau^2+1\right)\sigma^2_{RW}}{12\tau}$	$\sigma^2_{RW} > 0$
Random Ramp	RR	c_{RR}	$\frac{c_{RR}}{(2\pi f)^3}$	$\frac{\tau^2 c^2_{RR}}{4}$	–
Quantization Noise	QN	Q	$4Q^2 \sin^2(\pi f)$	$\frac{3Q^2}{2\tau^2}$	$Q > 0$
AR(1)	–	$a,\ \sigma^2_\nu$	$\frac{\sigma^2_\nu}{2\pi\left(1-2a\cos(f)+a^2\right)}$	$\frac{\left(\frac{\tau}{2}-3a-\frac{\tau a^2}{2}+4a^{\frac{\tau}{2}+1}-a^{\tau+1}\right)\sigma^2_\nu}{\frac{\tau^2}{8}(1-a)^2\left(1-a^2\right)}$	$\sigma^2_\nu > 0, \lvert a\rvert < 1$
ARMA(1,1)	–	$a,\ b,\ \sigma^2_\nu$	$\frac{\sigma^2_\nu\left(1-2b\cos(f)+b^2\right)}{2\pi\left(1-2a\cos(f)+a^2\right)}$	$\frac{\left(\frac{\tau}{2}-(1-a)^2\left(1-2ab+b^2\right)+(a-b)(1-ab)\left(\tau-3-\tau a+4a^{\frac{\tau}{2}}-a^{\tau}\right)\right)\sigma^2_\nu}{\frac{\tau^2}{8}(1-a)^2\left(1-a^2\right)}$	$\sigma^2_\nu > 0, \lvert a\rvert < 1$

where

$$S_{W_m W_l}(f) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} \rho_{W_m W_l, \tau} e^{-if\tau}. \tag{6.2.14}$$

The estimation of σ^2_{ml} is in general not straightforward. In Guerrier et al. [2013], we show that under the assumption of a Gaussian process for $\{y_k\}$, a suitable estimator is given by

$$\begin{aligned} \hat{\sigma}^2_{ml} = & \frac{1}{2} \sum_{\tau=-M(T_{ml})}^{M(T_{ml})} \left[\frac{1}{M(T_{ml})} \sum_{k \in T_{ml}} W_{m,k} W_{l,k+\tau} \right]^2 \\ & + \frac{1}{2} \sum_{\tau=-M(T_{ml})}^{M(T_{ml})} \left[\frac{1}{M(T_{ml})} \sum_{k \in T_{ml}} W_{m,k-\tau} W_{l,k} \right]^2 \end{aligned} \tag{6.2.15}$$

where T_{ml} is the smallest set of time indices containing both the indices in T_m and T_l (see Eq. (6.2.7)), and $M(T_{ml})$ their number. Alternatively, when the process is not Gaussian or when the sample size is very large as it is the case with the dataset analysed in Section 6.4 for which the computation of Eq. (6.2.15) is infeasible, one can use a *parametric bootstrap* to estimate $\text{cov}\left[\hat{\nu}^2(\tau_m), \hat{\nu}^2(\tau_l)\right]$. In such a scheme, Q samples of size N are simulated from $F(\hat{\boldsymbol{\theta}})$ on which Q wavelet variance sequences, $\hat{\nu}^2_q(\tau_m)$ and $\hat{\nu}^2_q(\tau_l)$ for $q = 1, \ldots, Q$, are computed and σ^2_{ml} is estimated by their empirical covariance.

A particular choice for the wavelet filter is given by the *Haar wavelet filter* whose first DWT filter ($j = 1$) is

$$\{h_{1,0} = 1/\sqrt{2},\ h_{1,1} = -1/\sqrt{2}\} \tag{6.2.16}$$

with length $L_1 = 2$. If the process is stationary with backward differences of order $d > 1$ one can use other wavelet filters such as Daubechies wavelet filters Daubechies [1992]. When the wavelet variance is evaluated with Haar wavelet filters, it is actually equal to half the Allan variance.

Chapter 4 highlighted the fact that the linear regression on identified linear regions of Allan variance plots provides reasonably estimated parameters only for a limited number of processes and is often biased. In the following Section, we propose instead a criterion based on a standardized distance between sample and model based wavelet variance that provides consistent estimators of the model's parameters for a wide range of models.

6.3 GMWM estimator

6.3.1 Principle

We saw in Eq. (6.2.6) that the variance of a wavelet coefficient sequence is equal to the integral of its PSD. Therefore, there exists a mapping

$$\boldsymbol{\theta} \mapsto \boldsymbol{\nu}^2(\boldsymbol{\theta}). \tag{6.3.1}$$

Such a mapping defines the theoretical wavelet variance implied by the parametric model $F(\boldsymbol{\theta})$. We exploit the connexion between the wavelet variance and $\boldsymbol{\theta}$ to define an estimator for $\boldsymbol{\theta}$ by trying in some sense to invert the relationship given in Eq. (6.3.1). This inverted map is used to compute the estimator $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}(\hat{\boldsymbol{\nu}}^2)$ where $\hat{\boldsymbol{\nu}}^2$ is the estimated wavelet variance. Finding explicitly an inverse mapping is in general impossible since this mapping is in most cases implicit. However, it is possible to invert the map in a specific point such as $\hat{\boldsymbol{\nu}}^2$ by calibrating the value of $\boldsymbol{\theta}$ in order to match $\boldsymbol{\nu}^2(\boldsymbol{\theta})$ with its empirical counterpart $\hat{\boldsymbol{\nu}}^2$.

Therefore, we propose to estimate the model's parameters using an estimator which combines on the one hand the wavelet variance and on the other hand the *Generalized Least-Squares* (GLS) principle, using the relationship given in Eq. (6.2.6). More precisely, we propose to find $\hat{\boldsymbol{\theta}}$ such that the wavelet variances implied by the model, say $\boldsymbol{\phi}(\boldsymbol{\theta})$, match the empirical wavelet variances, say $\hat{\boldsymbol{\phi}}$, and solve the following GLS optimization problem:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}} \left(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}(\boldsymbol{\theta})\right)^T \boldsymbol{\Omega} \left(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}(\boldsymbol{\theta})\right) \tag{6.3.2}$$

in which $\boldsymbol{\Omega}$, a positive definite weighting matrix[2], is chosen in a suitable manner (see below). Eq. (6.3.2) defines the GMWM estimator. The vector $\boldsymbol{\phi}(\cdot) = [\phi_j(\cdot)]_{j=1,\dots,J}$ is a binding function between $\boldsymbol{\theta}$ and $\boldsymbol{\nu}^2$ such that $\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{\nu}^2$, and $\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\nu}}^2$ and $\boldsymbol{\phi}(\hat{\boldsymbol{\theta}})$ are two estimators.

As mentioned in the introduction of this chapter, we proved the consistency of $\hat{\boldsymbol{\theta}}$ in Guerrier et al. [2013]. Hence, $\hat{\boldsymbol{\theta}}$ has the following distribution:

$$\sqrt{N} \cdot \left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right) \xrightarrow[N\to\infty]{\mathcal{D}} \mathcal{N}\left(0, \mathbf{V}_{\hat{\boldsymbol{\theta}}}\right) \tag{6.3.3}$$

where

$$\mathbf{V}_{\hat{\boldsymbol{\theta}}} = \mathbf{B}\mathbf{V}_{\hat{\boldsymbol{\phi}}}\mathbf{B}^T \tag{6.3.4}$$

and

$$\mathbf{B} = \left(\mathbf{D}^T\boldsymbol{\Omega}\mathbf{D}\right)^{-1}\mathbf{D}^T\boldsymbol{\Omega}. \tag{6.3.5}$$

The matrix $\mathbf{D}$ is given by

$$\mathbf{D} = \frac{\partial\boldsymbol{\phi}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^T} \tag{6.3.6}$$

and $\mathbf{V}_{\hat{\boldsymbol{\phi}}} = \mathbf{V}_{\hat{\boldsymbol{\nu}}^2}$. When $\boldsymbol{\Omega} = \mathbf{I}$, then

$$\mathbf{V}_{\hat{\boldsymbol{\theta}}} = \left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{D}^T\mathbf{V}_{\hat{\boldsymbol{\phi}}}\mathbf{D}\left(\mathbf{D}^T\mathbf{D}\right)^{-1}. \tag{6.3.7}$$

The most efficient estimator is obtained by choosing $\boldsymbol{\Omega} = \mathbf{V}_{\hat{\boldsymbol{\phi}}}^{-1}$, leading then to

$$\mathbf{V}_{\hat{\boldsymbol{\theta}}} = \left(\mathbf{D}^T\mathbf{V}_{\hat{\boldsymbol{\phi}}}^{-1}\mathbf{D}\right)^{-1}. \tag{6.3.8}$$

In practice, the matrix $\mathbf{D}$ is computed at $\hat{\boldsymbol{\theta}}$.

[2] $\boldsymbol{\Omega}$ has to be positive definite in order to ensure the convexity of Eq. (6.3.2).

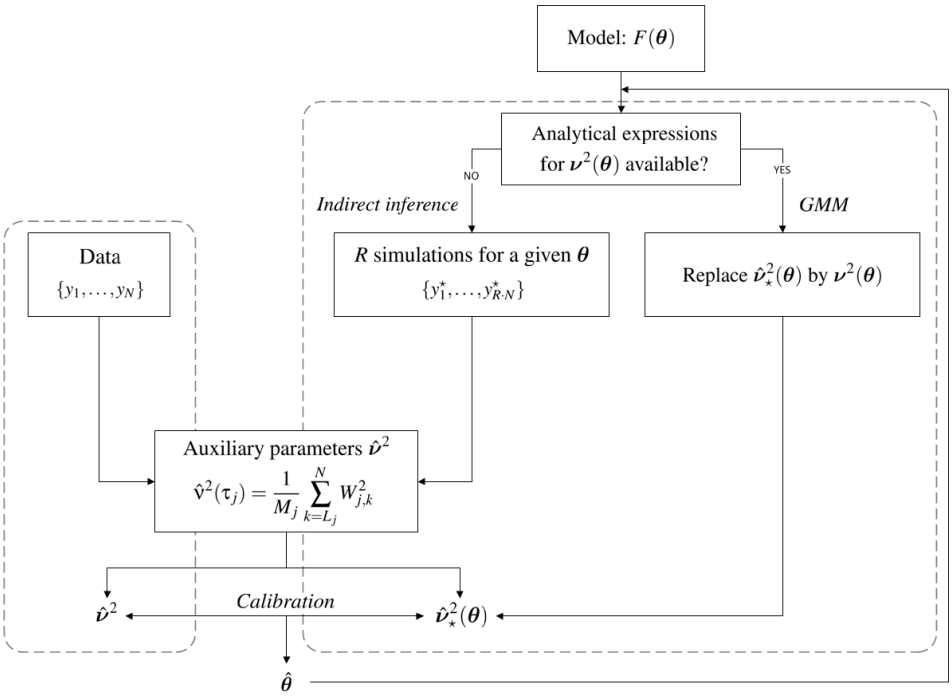

Figure 6.1: Principle of the GMWM estimator (adapted from Genton and Ronchetti [2003])

Obviously, the number of scales J should be $J \geq p$, with p the dimension of the parameter vector $\boldsymbol{\theta}$, but at the same time, as will be discussed in Section 6.4 (see Simulation 6.4.3), a too large J introduces variability in the estimator. In Guerrier et al., we propose a new method for selecting the scales J which minimize the determinant of $\mathbf{V}_{\hat{\boldsymbol{\theta}}}$ and propose a method to remove the finite sample bias of $\hat{\boldsymbol{\theta}}$ using a simulation based approach. Note that when $J > p$, i.e. the number of wavelet variances is greater than the dimension of the parameter vector $\boldsymbol{\theta}$, the goodness-of-fit of the model $F(\boldsymbol{\theta})$ to the data can be assessed by testing the hypotheses $H_0 : \mathbb{E}\left[\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}(\boldsymbol{\theta})\right] = 0$, $H_1 : \mathbb{E}\left[\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}(\boldsymbol{\theta})\right] \neq 0$ using the χ^2-test statistic

$$N \cdot \left(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}(\hat{\boldsymbol{\theta}})\right)^T \mathbf{V}_{\hat{\boldsymbol{\phi}}}^{-1} \left(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}(\hat{\boldsymbol{\theta}})\right) \xrightarrow[N\to\infty]{\mathcal{D}} \chi^2_{J-p} \tag{6.3.9}$$

under H_0 (see Hansen [1982]) and provided that $p < J < \infty$. The investigation of the finite sample properties of Eq. (6.3.9) are left for future research.

6.3.2 From the Generalized Method of Moments to Indirect Inference

The analytical expressions of the wavelet variances, $\boldsymbol{\nu}^2(\boldsymbol{\theta})$, used in Eq. (6.3.2) using the Haar wavelet filter defined in Eq. (6.2.16) can be computed for several well known models such as AR($\cdot$), sums of AR($\cdot$), ARMA($\cdot$,$\cdot$) and others using the general results of Zhang [2008] on the Allan variance. In addition, the analytical wavelet variance of sums of independent processes correspond to the sum of the wavelet variances of individual processes within the model. Indeed,

when the process is made up of the sum of independent processes, i.e. $Y_k = \sum_m (Y_m)_k$, Eq. (6.2.6) can be expanded to

$$\nu^2(\tau_j) = \int_{-1/2}^{1/2} |\widetilde{H}_j(f)|^2 \left(\sum_m S_{Y_m}(f) \right) df = \sum_m \nu_m^2(\tau_j) \tag{6.3.10}$$

with $S_{Y_m}(\cdot)$ the PSD and $\nu_m^2(\tau_j)$ the wavelet variance at scale τ_j of the sequence $\{(Y_m)_k\}$. Therefore, when an analytical expression for $\boldsymbol{\nu}^2(\boldsymbol{\theta})$ is available, the estimator defined in Eq. (6.3.2) can be seen as a *Generalized Method of Moments* (GMM) estimator (see Hansen [1982] for details).

As a possible extension of the GMWM when analytical expressions for $\boldsymbol{\phi}(\boldsymbol{\theta})$ in Eq. (6.3.2) are too complicate to compute, one can resort to simulations to compute $\boldsymbol{\phi}(\boldsymbol{\theta})$ and hence place the GMWM in the framework of *indirect inference* Gallant and Tauchen [1996], Gourieroux et al. [1993], Smith [1993]. Basically, given a sample of observations $\{y_k : k = 1, \ldots, N\}$ and an hypothetical model $F(\boldsymbol{\theta})$, we can define $\hat{\phi}_j$ as the wavelet variance $\hat{\nu}^2(\tau_j)$ estimated from the sample using Eq. (6.2.7), and $\hat{\phi}_j^\star(\boldsymbol{\theta})$ as the wavelet variance estimate $\hat{\nu}_\star^2(\tau_j)$ computed on a simulated series

$$\left\{ y_k^\star(\boldsymbol{\theta}) : k = 1, \ldots, R \cdot N \right\}, R \geq 1 \tag{6.3.11}$$

from $F(\boldsymbol{\theta})$. Alternatively, one can compute R wavelet variance estimates $\{\hat{\nu}_r^2(\tau_j) : r = 1, \ldots, R\}$ on simulated series

$$\left\{ y_k^{\star(r)}(\boldsymbol{\theta}) : k = 1, \ldots, N \right\} \tag{6.3.12}$$

and obtain

$$\hat{\nu}_\star^2(\tau_j) = \frac{1}{R} \sum_{r=1}^{R} \hat{\nu}_r^2(\tau_j). \tag{6.3.13}$$

Then $\hat{\boldsymbol{\phi}} = [\hat{\phi}_j]_{j=1,\ldots,J}$ and $\hat{\boldsymbol{\phi}}^\star(\boldsymbol{\theta}) = [\hat{\phi}_j^\star(\boldsymbol{\theta})]_{j=1,\ldots,J}$ are used in Eq. (6.3.2) to obtain an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$, which properties are described in e.g. Gourieroux et al. [1993]. In particular, for R sufficiently large, $\mathbf{V}_{\hat{\boldsymbol{\theta}}} \approx \mathbf{B}\mathbf{V}_{\hat{\boldsymbol{\phi}}}\mathbf{B}^T$. In that case, $\mathbf{B}$ can be computed numerically.

6.4 Simulations

This section is dedicated to the finite sample performance evaluation of the GMWM estimator compared to the MLE and the least-squares estimator. The simulations are an extention of the simulations presented in Chapter 5 and were taken from the work we published in Stebler et al. [2012]. For the GMWM estimator, we consider for $\boldsymbol{\Omega}$ in Eq. (6.3.2) the diagonal matrix with diagonal elements given by the inverse of the sample variance estimates of the MODWT using Eq. (6.2.9). Indeed, in our simulations, we found that choosing $\boldsymbol{\Omega} = \mathbf{V}_{\hat{\boldsymbol{\phi}}}^{-1}$ may lead in some cases to numerical instability, probably due to the estimation of the covariances using Eq. (6.2.15). For optimizing (6.3.2), we use a quasi-Newton optimization method. For the MLE, we use the EM algorithm together with the Kalman smoother as proposed in Chapter 5.

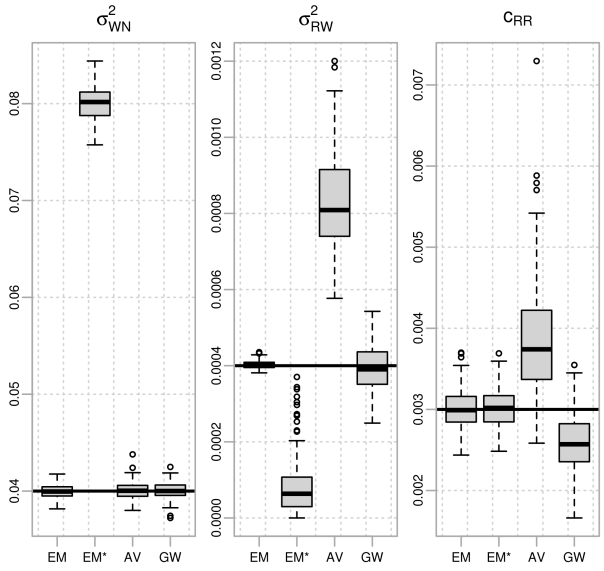

Figure 6.2: Performance comparison between the GMWM (GW) and the EM algorithm started at good (EM) and at bad ($EM^\star$) initial values, and the Allan variance technique (AV) for 200 simulated signals issued from a sum of a white noise, a random walk and a random ramp process. The true parameters are marked by horizontal lines (adapted from Stebler et al. [2012]).

Similarly to the procedure used in Chapter 5, several simulations were performed to validate our GMWM estimator. Composite stochastic processes which are assumed to have physically meaningful units (e.g. deg/s, μg/s), associated with different models $F(\boldsymbol{\theta})$ of increasing complexity were simulated with sampling interval Δt. The GMWM approach was applied to estimate $\boldsymbol{\theta}$ and these results were compared to alternative estimation methods (i.e. Allan variance and EM approaches).

Simulation 6.4.1. This simulation is the continuity of Simulation 5.4.1 (which was designated WN-RW-RR) in which the following model $F(\boldsymbol{\theta})$ was used:

$$Y_k = (Y_{RW})_k + (Y_{RR})_k + W_k, \qquad k \in \mathbb{Z}.$$

Remember that the parameters to estimate are

$$\boldsymbol{\theta} = \{\sigma^2_{WN}, \sigma^2_{RW}, c_{RR}\}.$$

The values of $\boldsymbol{\theta}^{(0)}$ for the GMWM estimator were set to 1.0 and several tests revealed an insensitivity of the estimator to the choice of $\boldsymbol{\theta}^{(0)}$ in terms of convergence. Figure 6.2 is the continuity of Figure 5.2 in which we added the results obtained by the GMWM estimator. Indeed, the GMWM algorithm converged correctly when initiated with $\boldsymbol{\theta}^{\star(0)}$. The RMSE as well as the R-RMSE of the GMWM method are listed in the left side of Table 6.2. These values have to be compared to the Allan variance, EM, and EM* performance listed in Table 5.1. The EM approach with $\boldsymbol{\theta}^{(0)}$ provides the best results but the GMWM is considerably better than $EM^\star$ or the Allan variance based approach.

Table 6.2: RMSE and relative RMSE (R-RMSE) of the GMWM estimator for 200 simulated processes of size $N = 6000$ from model WN-RW-RR (adapted from Stebler et al. [2012]).

	WN-RW-RR		**WN-GM-RR**	
	RMSE	**R-RMSE**	**RMSE**	**R-RMSE**
σ^2_{WN}	$8.59 \cdot 10^{-4}$	$2.15 \cdot 10^{-2}$	$1.38 \cdot 10^{-2}$	$2.15 \cdot 10^{-2}$
σ^2_{RW}	$5.99 \cdot 10^{-5}$	$1.50 \cdot 10^{-1}$	-	-
c_{RR}	$5.29 \cdot 10^{-4}$	$1.76 \cdot 10^{-1}$	$1.81 \cdot 10^{-4}$	$1.81 \cdot 10^{-1}$
β_{GM}	-	-	$3.31 \cdot 10^{-3}$	$4.14 \cdot 10^{-1}$
σ^2_{GM}	-	-	$9.67 \cdot 10^{-2}$	$3.87 \cdot 10^{-1}$

Simulation 6.4.2. This simulation is the continuity of Simulation 5.4.2 (which was designated as GM-WN-RR) in which the following model $F(\boldsymbol{\theta})$ was used:

$$Y_k = (Y_{GM})_k + (Y_{RR})_k + W_k, \qquad k \in \mathbb{Z}$$

with the following parameters to estimate:

$$\boldsymbol{\theta} = \{\beta, \sigma^2_{GM}, \sigma^2_{WN}, c_{RR}\}.$$

Figure 6.3 compares the performance of the GMWM estimator against the constrained EM ($EM^\star$) and the constrained EM with prior removal of the drift through least-squares adjustment (EM), when applied on 200 simulated signals. The GMWM method was able to estimate correctly $\boldsymbol{\theta}$ without any prior manipulation (i.e. data detrending), which is a clear advantage regarding inference on $\hat{\boldsymbol{\theta}}$. Moreover, the performance of the GMWM estimator is comparable to the EM approach (with prior drift removal) for β, σ^2_{GM} and σ^2_{WN} but is sensibly better regarding c_{RR}. RMSE and R-RMSE values are listed in the right part of Table 6.2 and have to be compared with the results listed in Table 5.2. Note that the initial values in $\boldsymbol{\theta}^{(0)}$ were all set to 1.0.

Simulation 6.4.3. Consider again the model $F(\boldsymbol{\theta})$ used in Simulation 6.4.2, i.e.

$$Y_k = (Y_{GM})_k + (Y_{RR})_k + W_k, \qquad k \in \mathbb{Z}$$

which has the following parameters:

$$\boldsymbol{\theta} = \{\sigma^2_{WN}, \sigma^2_{GM}, \beta, c_{RR}\}.$$

We are interested in estimating three (sub)models that we designate as

1. *Model 1*: we set $\sigma^2_{WN} = c_{RR} = 0$, and hence we have $Y_k = (Y_{GM})_k$;
2. *Model 2*: we set $c_{RR} = 0$, and hence we have $Y_k = (Y_{GM})_k + W_k$;
3. *Model 3*: the complete model, i.e. $Y_k = (Y_{GM})_k + (Y_{RR})_k + W_k$.

We simulated 100 sequences $\{y_k : k = 1, \ldots, N\}$ under *Model 3* by using $N = 6000$, $\Delta t = 1$ and

$$\boldsymbol{\theta} = \{4, 16, 0.05, 0.005\}.$$

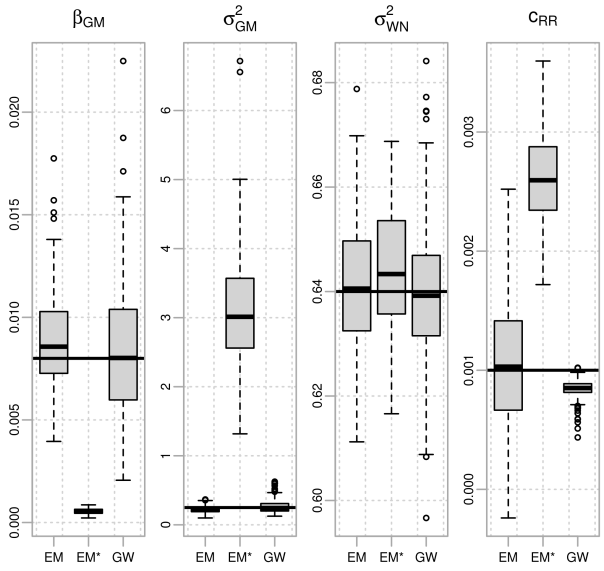

Figure 6.3: Performance comparison between the GMWM (GW), the EM algorithm with prior estimation of c_{RR} by ordinary least-squares (EM) and without ($EM^\star$) for 200 simulated signals issued from a sum of a white noise, a Gauss-Markov and a random ramp process. The true values of the parameters are marked by horizontal lines (adapted from Stebler et al. [2012]).

For the *Model 1* and *Model 2*, the parameters were constrained accordingly and not estimated. For both the GMWM (simulation based or not) and the EM, the initial values for the optimizations were set to

$$\boldsymbol{\theta}^{(0)} = \{1.0, 1.0, 1.0, 0.0\}$$

which is relatively far away from the true values $\boldsymbol{\theta}$. We found that the choice for the starting values is not a serious issue for the computation of the GMWM, except that starting far away can make the computational time longer. The results are listed in Table 6.3 in terms of RMSE and R-RMSE values obtained for each model by the GMWM and the EM estimator. Note that we also tried a simulation-based version of the GMWM (see Section 6.3.2) with a value of $R = 100$ and found that the RMSE are of the same order of magnitude as the RMSE values obtained with analytical wavelet variances (results not presented here). The sample wavelet variances were computed for $J = 12$ ($< \log(6000)/\log(2) \approx 12.55$) scales for all three models. The results show that with respect to the simpler *Model 1* and *Model 2*, the RMSE values are lower for the EM than for the GMWM estimator, while the RMSE of the EM explodes for *Model 3*. This last phenomena affects models with a drift component and was investigated in Chapter 5 dedicated to the EM algorithm. When the EM behaves well (*Model 1* and *Model 2*), it has a better performance in terms of RMSE than the GMWM estimator. However, one can further improve the efficiency of the latter by decreasing the number of scales J at which the wavelet variances are estimated. In this example $J = 12$ scales are used to estimate two or three parameters. Now if more scales are added (supposing a larger N), more variability is introduced in the GMWM estimator. The number of scales is obviously a function of the number of parameters p, but their choice (among the possible ones) depends

Table 6.3: RMSE and relative RMSE (R-RMSE) of the GMWM and EM estimators for 100 simulated processes of size $N = 6000$ from the models considered in Simulation 6.4.3 (adapted from Guerrier et al. [2013]).

		GMWM		**EM**	
		RMSE	**R-RMSE**	**RMSE**	**R-RMSE**
Model 1					
	σ^2_{GM}	2.00	0.13	1.26	0.08
	β	$8.14 \cdot 10^{-3}$	0.16	$4.09 \cdot 10^{-3}$	0.08
Model 2					
	σ^2_{GM}	1.92	0.12	1.55	0.10
	β	$1.05 \cdot 10^{-2}$	0.21	$4.94 \cdot 10^{-3}$	0.10
	σ^2_{WN}	0.13	0.03	0.14	0.04
Model 3					
	σ^2_{GM}	0.96	0.06	74.57	4.66
	β	$4.63 \cdot 10^{-3}$	0.09	0.04	0.85
	σ^2_{WN}	0.11	0.03	0.16	0.04
	c_{RR}	$2.79 \cdot 10^{-4}$	0.06	0.12	23.58

on the model. For example, consider the *Model 2*. A view on Figure 6.4 which depicts the wavelet variance sequences of the complete model (*Model 3*) and of the invidual processes (rate ramp, Gauss-Markov process), reveals that if the last three to four scales are ignored, then the wavelet variances are still able to capture information about the Gauss-Markov and white noise model components. Actually, removing the last four scales improves the efficiency of the GMWM estimator in this case (results not shown here).

Simulation 6.4.4. Three mixed Gauss-Markov processes are impossible to discriminate using the Allan variance method. Also, experiments revealed that the EM approach systematically diverges in such complex scenarios. Therefore, an attempt to retrieve the correct values of the individual model parameters is carried only with the GMWM estimation. The composite stochastic process $F(\boldsymbol{\theta})$ (designated as GM-GM-GM) we wish to estimate can be expressed as

$$Y_k = \sum_{m=1}^{3} (Y_{GM,m})_k, \qquad k \in \mathbb{Z}$$

where $(Y_{GM,m})_k$ is a first-order Gauss-Markov process with parameters

$$\boldsymbol{\theta} = \left\{\beta_m,\ \sigma^2_{GM,m}\right\}_{m=1,2,3}.$$

Thus, the goal of this simulation is to estimate the following parameter set:

$$\boldsymbol{\theta} = \{\beta_1, \sigma^2_{GM,1}, \beta_2, \sigma^2_{GM,2}, \beta_3, \sigma^2_{GM,3}\}.$$

To assess the performance of the GMWM in this context, 200 series $\{y_k : k = 1, \ldots, N\}$ with $N = 10^6$ were simulated under the following true parameter values:

$$\boldsymbol{\theta} = \{0.008, 2.5 \cdot 10^{-6}, 0.05, 4.5 \cdot 10^{-6}, 2.0, 29.50 \cdot 10^{-6}\}$$

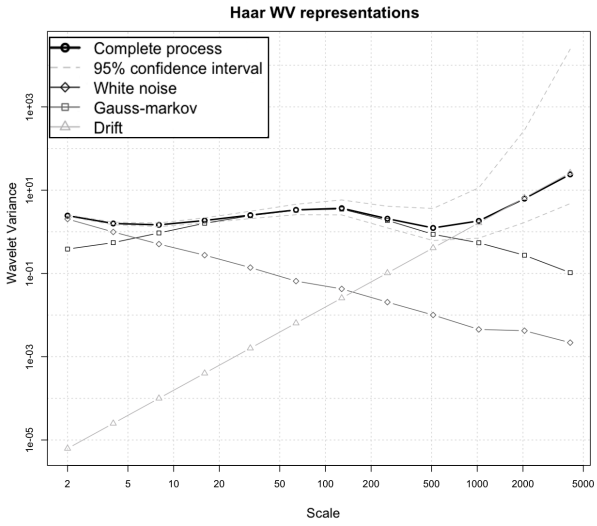

Figure 6.4: Haar wavelet variance sequence computed on the complete model composed of a white noise, a Gauss-Markov and a random ramp process (*Model 3*) used in Simulation 6.4.3 (thick black line), and on the individual processes composing the model (adapted from Guerrier et al. [2013]).

Figure 6.5 depicts the values $\hat{\boldsymbol{\theta}}$ and reveals that the GMWM technique was able to retrieve correctly the parameters of this complex model. The corresponding RMSE and R-RMSE values are listed in Table 6.4. Again, the initial values of $\boldsymbol{\theta}$ were all set to 1.0, and with exception of two runs out of 200, the GMWM estimator converged (i.e. the success of convergence without aiding was 99%). In these two cases, a grid search algorithm was employed to provide a "better" initial guess of the initial $\boldsymbol{\theta}$ to the GMWM and convergence occurred. Note also that the same simulation was repeated by setting all initial parameters to various values, and no significant difference was observed with respect to the results in Figure 6.5.

It should be mentioned that the composite process studied in Simulation 6.4.4, i.e. the sum of three Gauss-Markov processes, can be reparameterized as an ARMA(3,2) process (see e.g. Granger and Moris [1976], Terasvirta [1977]). Consequently, we could in principle estimate the latter instead of the former. However, when one of the Gauss-Markov processes lies near a unit root (as it is often the case with inertial sensors and in the simulation at hand), the estimation

Table 6.4: RMSE and relative RMSE (R-RMSE) of the GMWM estimator for 200 simulated processes of size $N = 10^6$ from model GM-GM-GM (adapted from Stebler et al. [2012]).

Parameter	**RMSE**	**R-RMSE**
β_1	$6.62 \cdot 10^{-4}$	$8.28 \cdot 10^{-2}$
$\sigma^2_{GM,1}$	$1.97 \cdot 10^{-7}$	$7.89 \cdot 10^{-2}$
β_2	$2.19 \cdot 10^{-3}$	$4.38 \cdot 10^{-2}$
$\sigma^2_{GM,2}$	$1.74 \cdot 10^{-7}$	$3.87 \cdot 10^{-2}$
β_3	$9.05 \cdot 10^{-3}$	$4.53 \cdot 10^{-3}$
$\sigma^2_{GM,3}$	$4.59 \cdot 10^{-8}$	$1.56 \cdot 10^{-3}$

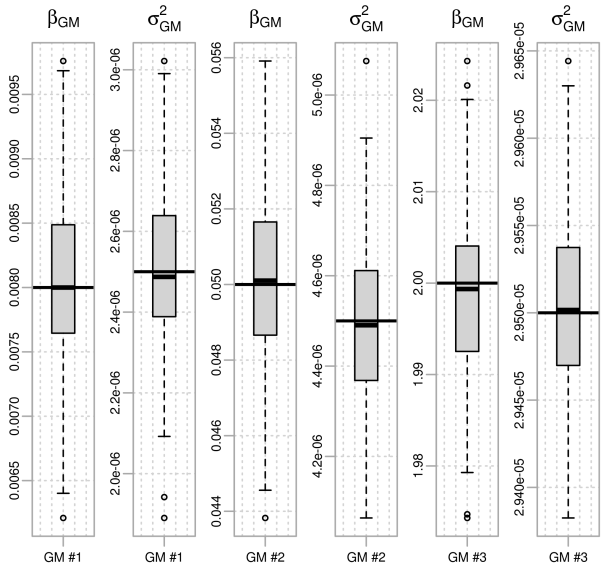

Figure 6.5: Performance of the GMWM algorithm for 200 simulated signals issued from a sum of three Gauss-Markov processes. The true parameters are marked by horizontal lines (taken from Stebler et al. [2012]).

of the associated ARMA$(\cdot, \cdot)$ model is rarely feasible. In addition, even if the estimation of the ARMA$(\cdot, \cdot)$ model is possible, the results shall be inverted to a Gauss-Markov-like representation since in many cases, and in particular in the simulation at hand, a sum of several Gauss-Markov models explains better the real underlying process (see Kittel [1958], Drexler [1992] for details). In order to recover the parameters of such a type of composite process from an estimated ARMA$(\cdot, \cdot)$ process together with their standard errors, several conditions need to be satisfied:

- The roots of the processes must lie outside the unit circle.
- The Jacobian matrix of the transformation between the two parametrizations must be invertible in order to apply the delta method Rao [1973], Benichou and Gail [1989].

With the signals generated in this Section, both conditions were not satisfied. In that case at least, estimating an ARMA$(\cdot, \cdot)$ process, converting the estimated model and performing an inference on the sum of Gauss-Markov processes is infeasible.

6.5 Implementation

The complete GMWM estimation framework has been implemented in a C++ based software optimized for handling very long signals (see Figure 6.6). The software takes as input files[3] containing the sensor signals to model. After choosing settings related to the MODWT transform

[3]Different proprietary formats such as *Applanix* or *XSens* are supported.

(upper left panel) such as the type of wavelet variance estimator, wavelet variance confidence interval, wavelet filter (Haar, Daubechies), or boundary conditions handling during filtering, as well as the computation of $\mathbf{V}_{\hat{\boldsymbol{\phi}}}$ (can be identity, diagonal or full), a model can be constructed (upper right panel). Several stochastic processes for which an analytical expression for $\boldsymbol{\phi}(\boldsymbol{\theta})$ is known can be selected. For each of them, user defined initial values as well as search boundaries can be defined. Several optimizers can be chosen: simplex, quasi-Newton, conjugate-gradients, constrained quasi-Newton, simulated annealing and Brent's method. Once the optimization is launched, the result of the wavelet variance curves matching can be inspected on a graph (lower left panel). The black curve represents $\hat{\boldsymbol{\phi}}$ computed on the sensor error signal, while the red curve is $\boldsymbol{\phi}(\hat{\boldsymbol{\theta}})$ estimated by the software. In addition, a processing report (lower right panel) containing the parameter values together with the 95% confidence intervals and processing settings can be generated and visualized.

Several issues related to the practical implementation of the GMWM framework will be described in the sequel.

6.5.1 Optimizer Initialization

The core of the GMWM estimation concept is the solving of an optimization problem. Sensitivity to initial values at which the optimizer starts is therefore a classical issue. We implemented the following strategy that will be described shortly.

The GMWM minimization problem can geometrically be interpreted as a minimization of the distance between the $\hat{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}(\hat{\boldsymbol{\theta}})$ curves, considering weights $\boldsymbol{\Omega}$. However in cases were $\hat{\boldsymbol{\phi}}$ has strong variations over all scales τ, the optimizer rapidly converges to a good match at the left part of the $\boldsymbol{\phi}$ curve. This is due to the $\boldsymbol{\Omega}$ weighting matrix which puts more importance on wavelet variance of small scales. A simple geometrical reformulation of Eq. (6.3.2) of the form

$$\hat{\boldsymbol{\theta}}^{(0)} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\ \boldsymbol{\kappa}^T\boldsymbol{\kappa} \tag{6.5.1}$$

where

$$\boldsymbol{\kappa} = \left[1 - \frac{\phi_j(\boldsymbol{\theta})}{\hat{\phi}_j}\right]_{j=1,\ldots,J} \tag{6.5.2}$$

enables to "flatten" the curves and to significantly decrease the convergence time. Only few iterations are then necessary to solve the original problem of Eq. (6.3.2).

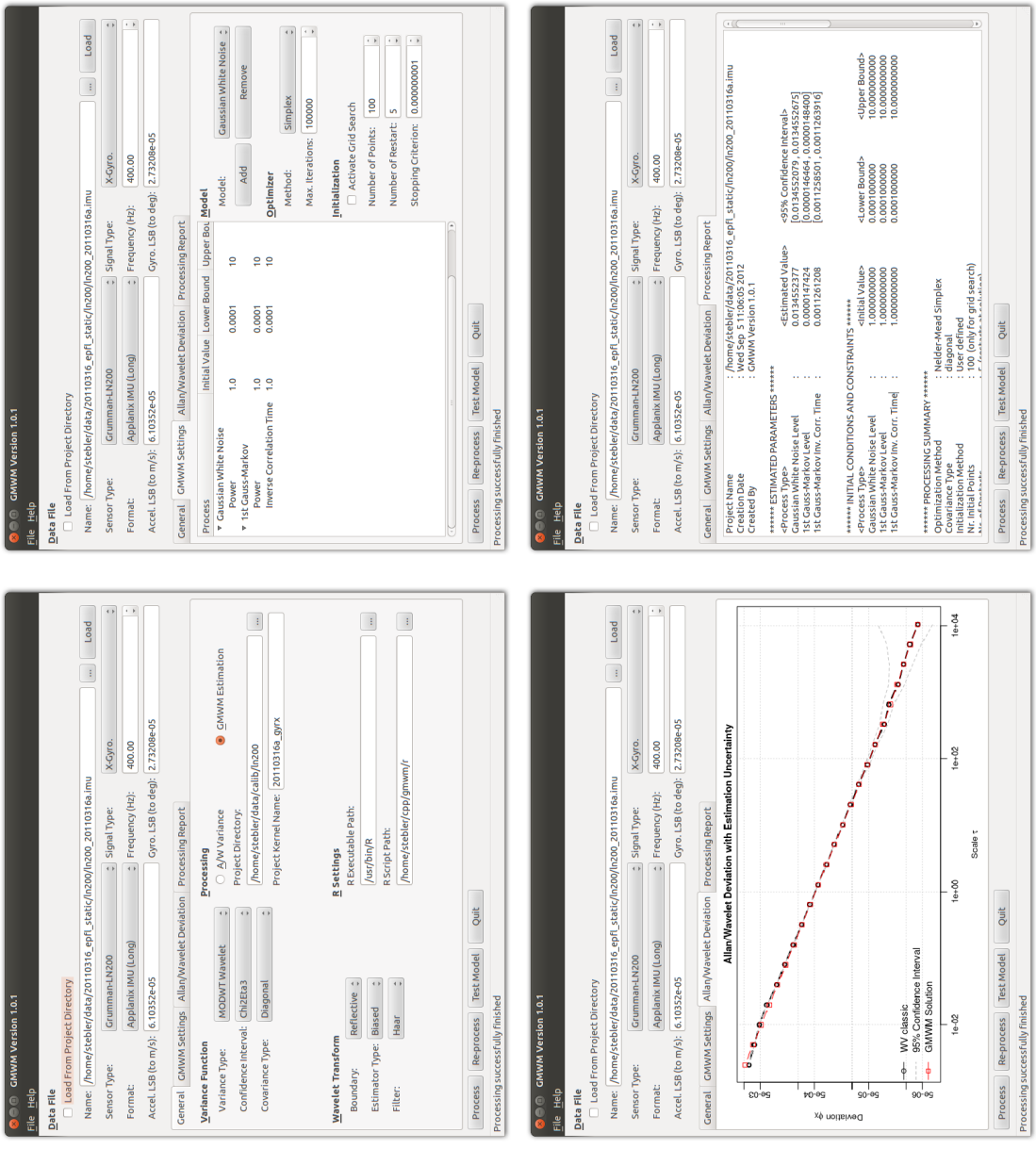

Figure 6.6: Graphical user interface of the C++ based software implementing the GMWM framework.

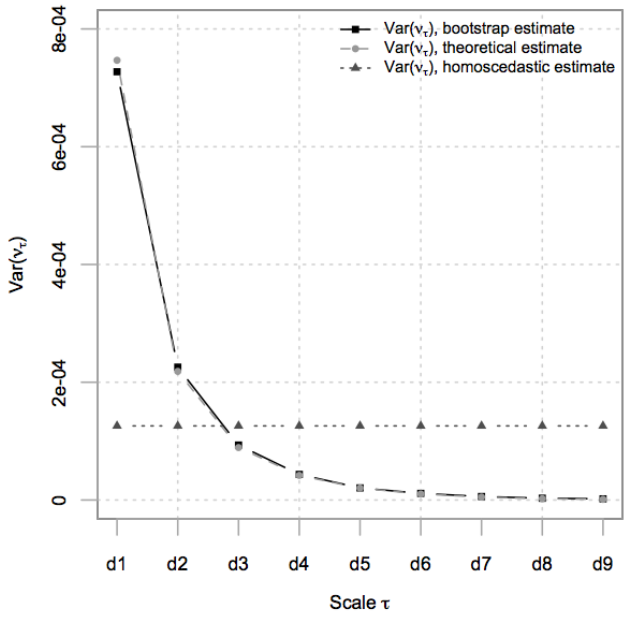

Figure 6.7: Estimated variances of $\hat{\boldsymbol{\nu}}^2$ with bootstrap (with $B = 1000$) and asymptotic (theoretical) approaches as well as the variances implied by an homoscedastic model (taken from Stebler et al. [2012]).

6.5.2 Note About the Wavelet Variance Covariance Estimation

This section presents a small simulation study that we published in Stebler et al. [2012] for the aim of illustrating how "far" the matrix $\mathbf{V}_{\hat{\boldsymbol{\nu}}^2}$ is from $\sigma_\epsilon^2\mathbf{I}$ (which is a common assumption in the standard Allan variance methodology). Indeed, B white noise processes, say $\{w_k^{(i)} : k = 1, ..., 1000,\ i = 1, ..., B\}$ with unit variance were generated. The wavelet variances of each process, denoted as $\hat{\boldsymbol{\nu}}_{(i)}^2$, were estimated. Then, the empirical covariances (and correlations) of $\hat{\boldsymbol{\nu}}^2$ were computed using

$$\hat{\boldsymbol{\Sigma}}_B = \frac{1}{B-1}\sum_{i=1}^{B}\left(\hat{\boldsymbol{\nu}}_{(i)}^2 - \bar{\boldsymbol{\nu}}^2\right)\left(\hat{\boldsymbol{\nu}}_{(i)}^2 - \bar{\boldsymbol{\nu}}^2\right)^T$$

where $\bar{\boldsymbol{\nu}}^2$ is the sample mean. For large B, the matrix $\hat{\boldsymbol{\Sigma}}_B$ is a fairly good approximation of $\mathbf{V}_{\hat{\boldsymbol{\nu}}^2}$ and corresponds to the (parametric) bootstrap estimator of cov $\left[\hat{\boldsymbol{\nu}}^2\right]$. Figure 6.7 shows the variances of $\hat{\boldsymbol{\nu}}^2$ (i.e. the diagonal of $\mathbf{V}_{\hat{\boldsymbol{\nu}}^2}$) estimated using the asymptotic formula version defined in Eq. (6.2.15) and the bootstrap estimator $\hat{\boldsymbol{\Sigma}}_B$. The variance (i.e. $\hat{\sigma}_\epsilon^2$) implied by the model $\sigma_\epsilon^2\mathbf{I}$ is also depicted. As expected, the bootstrap and asymptotic estimators are very close and therefore this demonstrates, at least in this example, the validity of Eq. (6.2.15). In addition, these two estimates are very far from $\hat{\sigma}_\epsilon^2$ which illustrates the fact that standard regression approach used in the Allan variance methodology is unsuitable in terms of efficiency.

Figure 6.8 presents the correlations estimated with the bootstrap and the asymptotic estimators which are also very close. Note also that the wavelet variances are highly correlated between neighboring scales. This confirms the inadequacy of the implicit model $\sigma_\epsilon^2\mathbf{I}$.

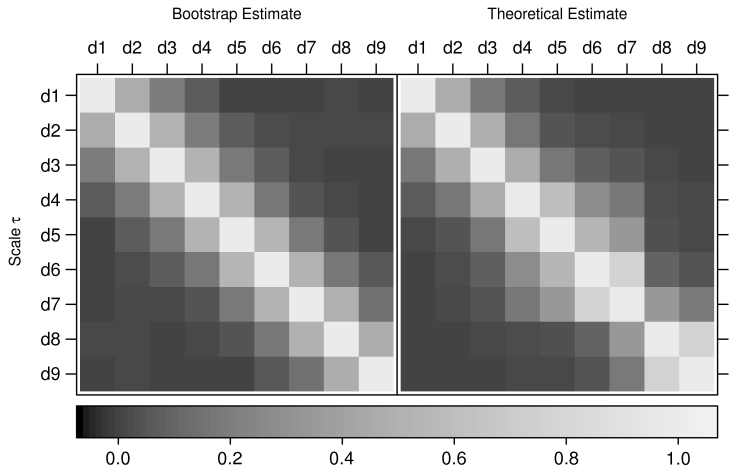

Figure 6.8: Comparison between the wavelet variance correlations estimated using a bootstrap approach with $B = 1000$, on the left part, and theoretical correlations computed using Eq. (6.2.15), on the right part (taken from Stebler et al. [2012]).

6.6 Conclusion

In this chapter, we presented a new estimator for the parameters of composite stochastic processes which is consistent for the class made of the sum of M independent white noise, random ramp, quantization noise, random walk and AR(1) (i.e. first-order Gauss-Markov) processes with $M < \infty$. The enlargement of this class involves verifying the conditions provided in Theorem 2 and more generally in Theorem 1 of the study we published in Guerrier et al. [2013] and is left for future research. As demonstrated in Stebler et al. [2012], the GMWM has many advantages compared to existing alternative methods for applications in sensor calibration.

Note that one could wonder why using wavelet variances in $\boldsymbol{\phi}$ instead of the PSD for which a similar estimation procedure could be used. We believe that the wavelet-based approach is more suitable for the following reasons that we mentioned also in Guerrier et al. [2013]:

- Inference on the PSD would make the optimization of a least-squares type measure (between the empirical and model based PSD) more difficult to solve when the PSD has large variability over very narrow frequency bands. As shown in Percival and Walden [2000], the wavelet coefficient at scale τ_j is associated with frequencies in the interval $[1/2^{j+1}, 1/2^j]$ and Eq. (6.2.6) can be approximated by

$$\nu^2(\tau_j) \approx 2 \int_{1/2^{j+1}}^{1/2^j} S_Y(f) df. \tag{6.6.1}$$

  This means that the wavelet variance summarizes the information in the PSD using just one value per octave frequency band. This property is particularly useful when the PSD is relatively featureless within each octave band. In the case of the widely used pure power

law processes ($S_Y(f) \propto |f|^\alpha$) for example, from Eq. (6.6.1) one gets $\nu^2(\tau_j) \propto \tau_j^{-\alpha-1}$, meaning that no information is lost when using the "PSD summary" given by the wavelet variance.

- The computation of empirical wavelet variances is more straightforward than nonparametric PSD. For example, the periodogram is an inconsistent estimator of $S_Y(f)$ and can be badly biased even for large samples sizes (because of the frequency leakage effects). Therefore, more sophisticated PSD estimators and/or smoothing techniques such as prewhitening or tapering shall be employed with a hope to approach the consistency provided by the GMWM estimator.
- The PSD of two important processes in sensor error models, namely the random ramp (drift) and the random walk, cannot be distinguished (both have slope of -2 in a log-log representation of the PSD).
- The MODWT on which the wavelet variance computation is based requires a number of multiplications of order $N \log_2 N$, which is the same order as the widely used fast Fourier transform algorithm. For this reason, the employment of wavelet variance does not increase the computational burden.

In summary, we demonstrated that the GMWM estimator is able to handle complex error models for which the Allan variance-based techniques failed and the EM algorithm did not converge. We will further investigate its applicability in the context of inertial navigation.

Chapter 7

Study of Stochastic Errors in Inertial Sensors

7.1 Introduction

The aim of this chapter is to perform sensor calibration using the proposed GMWM framework of Chapter 6. Its operation within a filter design loop is demonstrated on several IMUs belonging to the tactical and low-grade classes. The first part of the chapter is dedicated to the modeling of sensor error signals acquired in static laboratory conditions (Section 7.2). The performance of the constructed models when operated within the Kalman filter is then evaluated in Section 7.3. The second part (Section 7.4) investigates the influence of the dynamics on the sensor error behavior before drawing conclusions in Section 7.5.

7.2 Evaluation in Static Conditions

7.2.1 Error Signal Construction

Data were collected from a static *Litton LN-200* tactical-grade IMU (sampling at 400Hz during 6 hours; see Figure 7.1), an *XSens MTi-G* MEMS-based IMU (sampling at 100Hz during 4 hours; see left panel of Figure 7.4), and an *XSens MTx* MEMS-based IMU (sampling at 100Hz during 4 hours; see right panel of Figure 7.4). All devices operated under constant temperature and stationary conditions. After mean removal, the observed signals $\{y_k : k = 1, \ldots, N\}$ at hand contain measurement errors driven by stochastic processes issued from an unknown model $F(\boldsymbol{\theta})$. First, the GMWM estimator will be applied on accelerometer and gyroscope error signals of each device. Second, the models are validated by simulating twenty realizations under $\hat{\boldsymbol{\theta}}$, computing their Haar wavelet variance, and comparing them to the Haar wavelet variance of the signal under study. Note that exactly the same conclusion could be drawn by comparing the Allan variance curves because of the linear relationship between the Haar wavelet variance and the Allan variance. For each error signal under study, an analysis of what the Allan variance calibration procedure would provide when applied on the signals is presented.

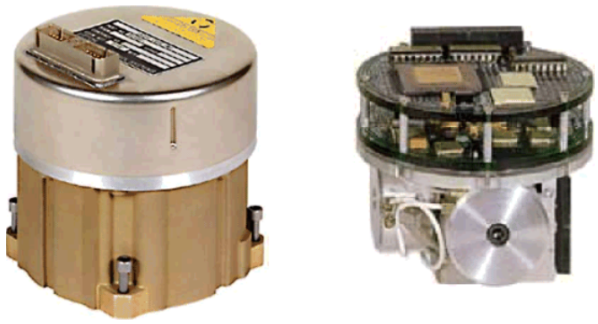

Figure 7.1: *Litton LN-200* tactical-grade IMU (taken from Litton Guidance and Control Systems Division [1996]).

7.2.2 Tactical-Grade IMU (Litton LN-200)

Let the observed *LN-200* gyroscope and accelerometer error signal both be designated as the sequence $\{y_k : k = 1, \ldots, N\}$ with $N = 9'220'599$ samples.

Model Building

Computation of the Haar wavelet variance (or Allan variance) curve on the (X-axis) gyroscope error signal reveals the presence of a weakly correlated noise structure since the slope of the curve lies near $-1/2$ (see black points in the left panel of Figure 7.2). However, the curve signifcantly deviates from the theoretical Gaussian white noise shape at small scales (left side of the plot). At this stage, the designer has the choice either to favor a simple Gaussian white noise model of the type

$$Y_k = W_k, \quad \text{for} \quad k \in \mathbb{Z} \tag{7.2.1}$$

or to account for this phenomena by choosing a more complex model. We choose to estimate a first-order Gauss-Markov model with an expected very short correlation time (or large β parameter value). The process will therefore be described by

$$Y_k = (Y_{GM})_k, \quad \text{for} \quad k \in \mathbb{Z}. \tag{7.2.2}$$

The task of the GMWM estimator is to estimate the parameters

$$\boldsymbol{\theta} = \{\beta, \sigma^2_{GM}\} \tag{7.2.3}$$

from the observed gyroscope error signal $\{y_k\}$.

By adopting the same methodology for the (X-axis) accelerometer, the Haar wavelet variance curve computed on the signal (black points in the right panel of Figure 7.2) clearly reveals the presence of quantization noise (left part of the curve), Gaussian white noise (middle part) and a random walk (right part). Therefore, the model we choose is written as:

$$Y_k = (Y_{QN})_k + (Y_{RW})_k + W_k, \quad \text{for} \quad k \in \mathbb{Z}. \tag{7.2.4}$$

The following parameter set has to be estimated:

$$\boldsymbol{\theta} = \{Q, \sigma^2_{WN}, \sigma^2_{RW}\} \tag{7.2.5}$$

from the observed signal $\{y_k : k = 1, \ldots, N\}$.

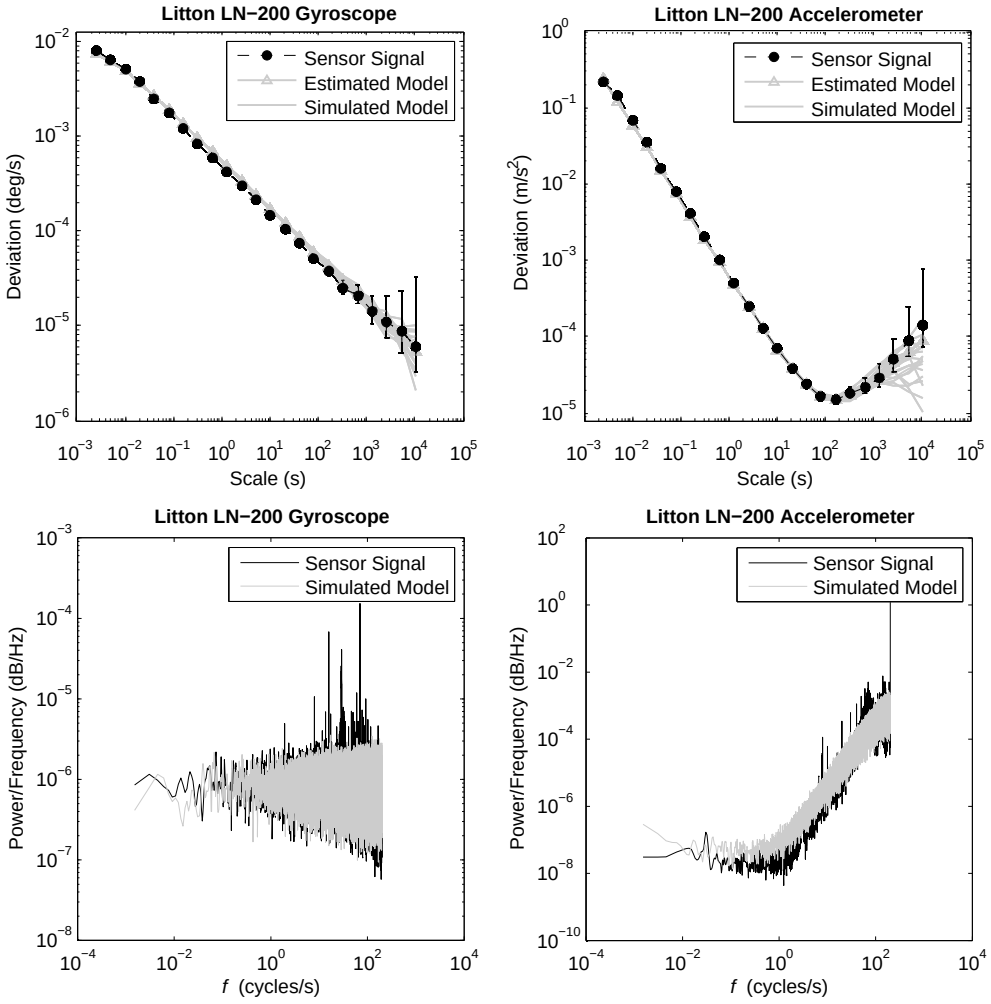

Figure 7.2: Results of the *Litton LN-200* accelerometer and gyroscope error modeling. The upper panels depict wavelet variance sequences of the X-axis gyroscope (left panel) and accelerometer (right panel) error signal (black circles) with associated 95% confidence intervals. The wavelet variance sequences resulting from the estimated model and the sequences of 20 synthetic signals issued from the estimated model, are drawn as gray triangles and gray lines, respectively. The black lines in the lower panels show the PSD of the gyroscope (left panel) and accelerometer (right panel) error signals, and the gray lines correspond to the PSD of the estimated model.

Model Estimation

The GMWM estimator, was operated considering $\mathbf{\Omega}$ in Eq. (6.3.2) as a diagonal matrix with diagonal elements given by the inverse of the sample variance estimates of the MODWT using Eq. (6.2.9). We used a quasi-Newton optimization method for solving the optimization problem defined in Eq. (6.3.2).

Obviously, the white noise model estimation result obtained for the gyroscope signal employing the GMWM estimator will be almost identical to what would be obtained when using Allan variance-based analysis (not shown here). For the Gauss-Markov model, the values of the estimated parameters are:

$$\begin{aligned} \hat{\beta} &= 560.82 \ \text{ Hz} \\ \hat{\sigma}_{GM} &= 0.012 \ \text{deg/s}. \end{aligned} \tag{7.2.6}$$

Since the *LN-200* was sampling at 400 Hz, this process can be interpreted as a white noise with level $\hat{\sigma}_{WN} = 0.0005 \ \text{deg/s}/\sqrt{\text{Hz}}$. Comparing this value to the manufacturer's model Litton Guidance and Control Systems Division [1996] specifying an angular random walk PSD level of $0.0006 - 0.0025 \ \text{deg/s}/\sqrt{\text{Hz}}$ confirms the order of magnitude of our estimate. The result of the wavelet variance matching defined in Eq. (6.3.2) can be visualized with the wavelet variance of the estimated model (gray line with triangle markers in the left panel of Figure 7.2). It can be seen that the model accounts for the curve deflection in the left side of the plot.

The parameter models obtained for the accelerometer are the following:

$$\begin{aligned} \hat{Q} &= 0.19 \\ \hat{\sigma}_{WN} &= 15 \ \mu g/\sqrt{\text{Hz}} \\ \hat{\sigma}_{RW} &= 5.4 \cdot 10^{-4} \ \mu g/\sqrt{\text{Hz}} \end{aligned} \tag{7.2.7}$$

Again, the estimation quality can visually be evaluated by checking the wavelet variance fit between the estimated model (gray line with triangle markers) and the signal in the right panel of Figure 7.2. Note that compared to the manufacturer's model specifying a velocity random walk PSD level of 50 $\mu g/\sqrt{\text{Hz}}$, the value of our estimated model, i.e. $\hat{\sigma}_{WN}$, is lower.

Model Validation

We validate the model at observation level by generating 20 synthetic signals $\{(y_j^\star)_k : k = 1, \ldots, N; j = 1, \ldots, 20\}$ issued from the estimated model $F(\hat{\boldsymbol{\theta}})$. Then we compute the Haar wavelet variances (or Allan variances) $\boldsymbol{\nu}_j^\star$ for each $\{(y_j^\star)_k\}$ and plot them on the wavelet variance plots of Figure 7.2 as gray lines. Although the confidence intervals associated to the $\boldsymbol{\nu}_j^\star$ sequences are not drawn for clarity reasons, excepting a few realizations of the accelerometer model, they all intersect with the signal wavelet variance confidence interval (black lines above and below the black circles). A similar validation can be done by computing the PSD of the error signals (gray lines in lower panels of Figure 7.2) and comparing them to the PSD of one realization issued from the estimated model (black lines). Only one realization is drawn for clarity reasons. For both, the gyroscope and the accelerometer, the PSD of the estimated model matches fairly well the PSD associated to $\{y_k\}$. Therefore, it can be assumed that correct stochastic assumptions are done when designing a navigation filter with these models.

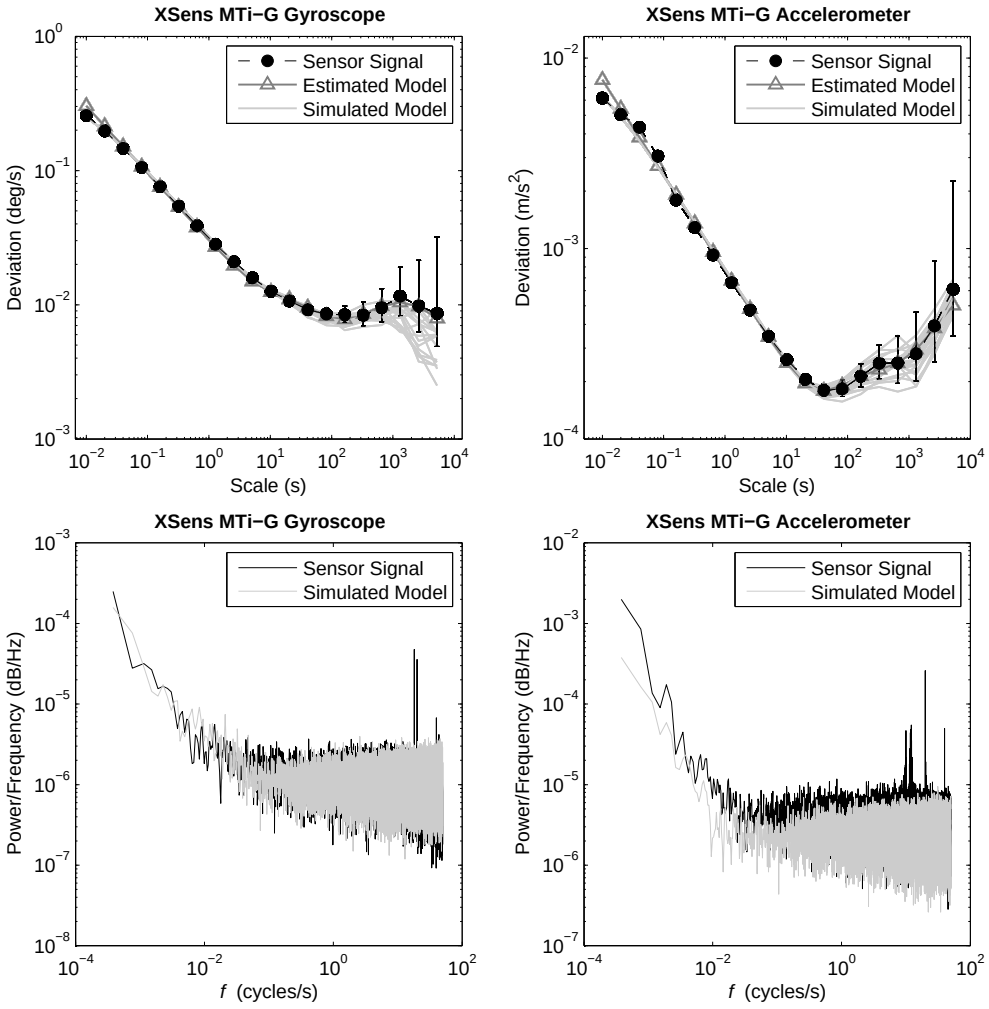

Figure 7.3: Result of the *XSens MTi-G* accelerometer and gyroscope error modeling. The upper panels depict the wavelet variance of the X-axis gyroscope (left panel) and accelerometer (right panel) error signal (black circles) with associated 95% confidence interval, the wavelet variance resulting from the estimated model (gray triangles), and the wavelet variance of 20 synthetic signals issued from the estimated model (gray lines). The black lines in the lower panels show the PSD of the gyroscope (left panel) and accelerometer (right panel) error signals, and the gray lines correspond to the PSD of the estimated models.

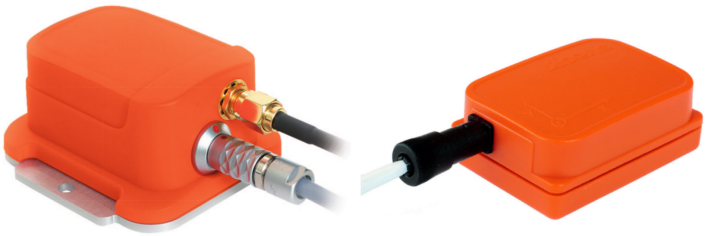

Figure 7.4: *XSens MTi-G* (left panel) and *MTx* (right panel) MEMS-based IMUs (adapted from XSens Technologies B.V [2009b,a]).

7.2.3 MEMS-Based IMU (XSens MTx/MTi-G)

Here, we observe signals from MEMS-based accelerometers and gyroscopes mounted in *XSens MTx* and *MTi-G* IMUs that will be noted as $\{y_k : k = 1, \ldots, N\}$ with $N = 2'057'424$ samples.

Model Building

Using the same methodology as above, a look on the wavelet variance curves for one (X-axis) *MTi-G* gyroscope (upper left panel in Figure 7.3) and for one (X-axis) *MTi-G* accelerometer (upper right panel in Figure 7.3) reveals the presence of correlated and uncorrelated noise. For both signals, we used a model $F(\boldsymbol{\theta})$ composed of the following processes:

$$Y_k = \sum_{m=1}^{M} (Y_{GM,m})_k + W_k, \quad \text{for} \quad k \in \mathbb{Z}. \tag{7.2.8}$$

where M was determined by going several times through the filter design loop (see Figure 4.1). In each loop, the value of M was changed and the final model was accepted as a compromise between the model complexity and the matching quality (goodness-of-fit). Therefore, a value of $M = 2$ was chosen for the gyroscope, and $M = 3$ for the accelerometer model. Therefore, we have the following parameters to estimate:

$$\boldsymbol{\theta} = \left\{\sigma^2_{WN}, \beta_m, \sigma^2_{GM,m}\right\}_{m=1,\ldots,M} \tag{7.2.9}$$

from the observed gyroscope and accelerometer error signal $\{y_k\}$.

Model Estimation

The estimated model for the gyroscope has the following values:

$$\begin{aligned}
\hat{\sigma}_{WN} &= 0.032 \ \ \text{deg/s}/\sqrt{\text{Hz}} \\
\hat{\beta}_1 &= 0.024 \ \ \text{Hz} \\
\hat{\sigma}_{GM,1} &= 7.63 \cdot 10^{-4} \ \ \text{deg/s}/\sqrt{\text{Hz}} \\
\hat{\beta}_2 &= 9.03 \cdot 10^{-4} \ \ \text{Hz} \\
\hat{\sigma}_{GM,2} &= 8.64 \cdot 10^{-4} \ \ \text{deg/s}/\sqrt{\text{Hz}}
\end{aligned} \tag{7.2.10}$$

The quality of the fit can be judged with the gray line with triangle markers in the upper left panel of Figure 7.3. Despite the slight deviation from the first wavelet variance point of the signal with respect to the model, the estimated model still overbounds the uncorrelated noise level. Note that despite the difficulty to compare models, the angular random walk provided by the manufacturer is 0.05 deg/s/$\sqrt{\text{Hz}}$ XSens Technologies B.V [2009a].

For the accelerometer, the estimated model has the following parameter values:

$$\begin{aligned} \hat{\sigma}_{WN} &= 85.5\ \mu g/\sqrt{\text{Hz}} \\ \hat{\beta}_1 &= 25.8\ \text{ Hz} \\ \hat{\sigma}_{GM,1} &= 143\ \mu g \\ \hat{\beta}_2 &= 0.04\ \text{ Hz} \\ \hat{\sigma}_{GM,2} &= 103\ \mu g \\ \hat{\beta}_3 &= 2.32 \cdot 10^{-4}\ \text{ Hz} \\ \hat{\sigma}_{GM,3} &= 900\ \mu g \end{aligned} \tag{7.2.11}$$

As an indication, the velocity random walk PSD level indicated by the manufacturer is 0.002 m/s^2/$\sqrt{\text{Hz}}$ XSens Technologies B.V [2009a] which is slightly higher than our estimated $\hat{\sigma}_{WN}$ value of $8.55 \cdot 10^{-4}$ m/s^2/$\sqrt{\text{Hz}}$. Similarly to the gyroscope, the short-term noise structure is quite complex and none of the considered models in this thesis could easily approximate this.

Model Validation

In Figure 7.3, the PSD of one realization issue from the estimated model is depicted as a gray line superposed to the signal PSD (black line) for both, the *MTi-G* gyroscope (lower left panel) and the *MTi-G* accelerometer (lower right panel). The quality of the matching shows that the spectral noise structure generated by both models correctly reproduces the sensors' noise shapes.

Until now, we assumed a unique model for all three sensors composing an IMU. To verify this assumption, we individually modeled each of the three gyroscopes and accelerometers composing an *MTi-G* and *MTx* IMU. In the upper panels of Figure 7.5, the three Haar wavelet variance curves corresponding to the error signals of the gyroscopes (left panel) and the accelerometers (right panel) from the *MTi-G* IMU are depicted as black lines. The gray lines represent the Haar wavelet variance issued from the GMWM estimation. Since the order of magnitude is clearly similar among each sensor type, the use of one single model per sensor type in the filter design is relevant in this case. The same operation was repeated on the *MTx* IMU and the results are shown in the lower panels of Figure 7.5. The same conclusions as for the *MTi-G* can be drawn from these panels.

7.2.4 PSD Estimation Capability of the Wavelet Variance

In the conclusion of Chapter 6, Eq. (6.6.1) highlighted the fact that the wavelet variance provides an octave-band estimate of the PSD. This was one reason to use the wavelet variance signal instead of the PSD in cases where the PSD is relatively featureless. According to Percival

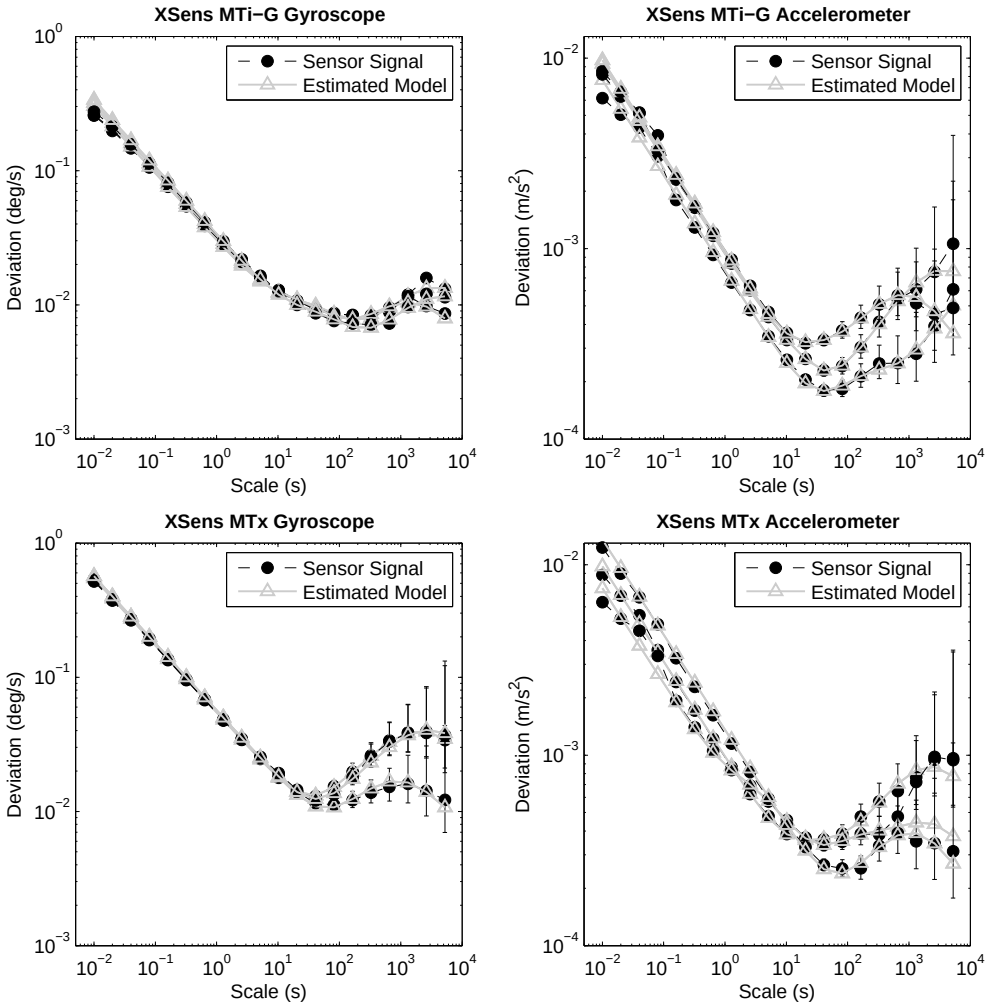

Figure 7.5: Result of the GMWM modeling for the *XSens MTi-G* (upper panels) and the *MTx* (lower panels) accelerometer and gyroscope error signals. For each IMU axis, the wavelet variance sequences estimated on the original error signals are drawn as black circles (together with the 95% confidence intervals). The wavelet variance sequences resulting from the estimated models are drawn as gray lines with triangles. Note that the confidence intervals associated to the model-based wavelet variances are not drawn for readibility reasons.

and Walden [2000], as the width L of the wavelet filter $\{\tilde{h}_l\}$ used to form $\{\tilde{h}_{j,l}\}$ increases, the approximation expressed in Eq. (6.6.1) improves because $\{\tilde{h}_{j,l}\}$ then becomes a better approximation to an ideal band-pass filter. Therefore, by making the assumption that L is chosen such that Eq. (6.6.1) is a reasonable approximation, it is possible to estimate $S_Y(\cdot)$ using a function $\bar{S}_Y(\cdot)$ that is piecewise constant over octave bands $\left[\frac{1}{2^{j+1}\Delta t}, \frac{1}{2^j \Delta t}\right]$ for $j = 1, \ldots, J$. Then, when $\frac{1}{2^{j+1}\Delta t} < f \leq \frac{1}{2^j \Delta t}$, we assume

$$\bar{S}_Y(f) = C_j \tag{7.2.12}$$

where C_j is a constant defined such that Percival and Walden [2000]

$$\int_{1/(2^{j+1}\Delta t)}^{1/(2^j \Delta t)} S_Y(f) df = \int_{1/(2^{j+1}\Delta t)}^{1/(2^j \Delta t)} \bar{S}_Y(f) df = \frac{C_j}{2^{j+1}\Delta t}. \tag{7.2.13}$$

From Eq. (6.6.1), we have

$$\nu^2(\tau_j) \approx \frac{C_j}{2^j \Delta t} \tag{7.2.14}$$

and hence we can use $\hat{C}_j = 2^j \hat{\nu}^2(\tau_j)\Delta t$ to estimate PSD levels. The upper panels of Figure 7.6 shows estimated PSD levels $\hat{C}_j$ plotted as a constant line over each octave band for the *MTi-G* gyroscope and accelerometer, respectively. The full line corresponds to the Haar-based estimate, while the dashed line represents the estimate using a higher order D(6) wavelet filter. In this case, there is good agreement between the two PSD estimates over all frequencies, meaning that the use of the Haar filter is sufficient. The well-known frequency leakage phenomenon was also mentioned in Chapter 6. Therefore, we also plotted PSD estimates based upon the leakage-prone periodograph (circles) and a multitaper estimator (asterisks) which should be relatively free of leakage (for more details on periodogram and multitaper PSD estimators, see Percival and Walden [2000]). In order to obtain the periodogram and multitaper PSD estimates, we averaged the PSD estimates obtained at each Fourier frequency $f_k = \frac{k}{N\Delta t}$, $k = 1, 2, \ldots$ (gray line corresponding to the multitaper PSD estimate) over octave frequency bands, meaning that each of the octave band averaged estimates is plotted versus the average of the Fourier frequencies associated with the estimates. In this case, the Haar-based PSD estimate does not suffer from frequency leakage which should be visible at the low power portion of the estimated PSD (at high frequencies).

7.3 Influence on EKF Solution

7.3.1 Testing Models

As already mentioned in Section 4.2.1, validating and comparing different models on the state level basis is extremely tricky. Issues like observability, non-modeled parasite signals generated from the dynamics, or environmental condition variations may largely affect the output of the navigation filter. Therefore, a statement telling that one model is *in general* more accurate than another must be considered with caution. In this thesis, we chose to construct models on signals sampled in static conditions and to validate them using tools such as wavelet variance

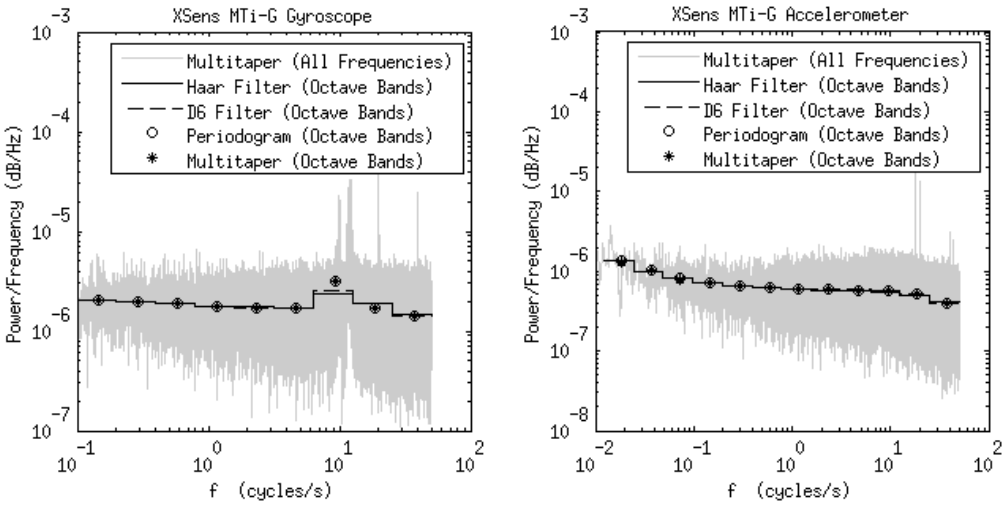

Figure 7.6: Comparison of octave band PSD estimates for the *XSens MTi-G* gyroscope (left panel) and accelerometer (right panel) error signal based on the periodogram, a multitaper PSD estimate and Haar and D(6) wavelet variance estimates.

(or Allan variance) and PSD on the observation level. This means that the built model is valid for the conditions at hand during calibration.

A navigation filter designer may be interested in analysing the effect of a constructed model on a trajectory, independently from any unmodeled effects which were not experienced during the calibration phase. In this way, the model can be tested under the aimed dynamics and typical operation duration. Although the GMWM estimator opens the door to the estimation of complex composite models such as sums of Gauss-Markov processes, they are not necessarily observable under all dynamical conditions. Moreover, the duration of the trajectory may be too short to estimate some processes with large correlation time. These two reasons may justify choosing a simplified version of the model, i.e. removing some of the Gauss-Markov processes, or lumping them together. Simulation provides a means for performing such an analysis. In this respect, we use the following procedure:

1. An accurate navigation solution for a given trajectory is computed using signals from high-grade sensors (typically tactical/navigation-grade IMUs, L1/L2 carrier-phase differential GNSS positioning).

2. The real error signals acquired by the sensor under static conditions are added to synthetic inertial signals emulated along the reference trajectory (note that the synthetic signal matches the chosen reference perfectly; hence it contains no errors at all).

3. Artifical outages in GNSS position/velocity observations are added to the dataset which is subsequently processed by INS/GNSS integration implementing a closed-loop EKF.

4. The qualily of the model is judged by analysing the actual navigation error as well as the EKF-predicted accuracy during inertial coasting mode (i.e. periods with no external aiding).

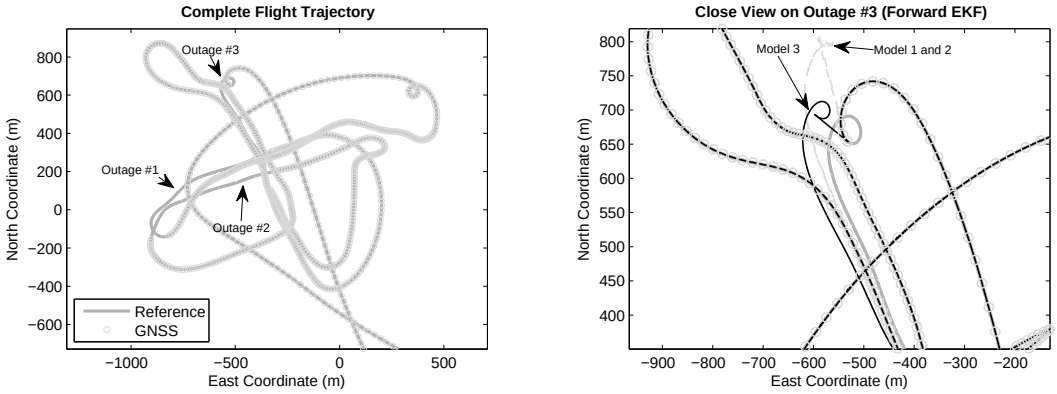

Figure 7.7: Emulated trajectory issued from an ALS flight. The left panel shows the reference trajectory with the three articial GNSS-free periods. The right panel provides a close view on the third GNSS-free period, together with the estimated trajectory using *Model 3*.

7.3.2 Example: XSens MTi-G IMU operating on a Small Flying Platform

In this example, we assume the *XSens MTi-G* IMU, for which a model was constructed in Section 7.2.3, to operate on a light flying platform such as an *Unmanned Air Vehicle* (UAV). The target application is mapping using remote sensing sensors which must be accurately georeferenced using the integrated inertial and geodetic-grade GNSS observations. Typical mission durations are between 5 to 30 minutes. We are interested in studying the behavior of the EKF operating in the aimed context with the given set of navigation sensors used for georeferencing.

The left panel of Figure 7.7 shows an extract of a trajectory issued from a helicopter flight performing *Airborne Laser Scanning* (ALS) Schaer et al. [2009]. There, the laser data were georeferenced using a trajectory obtained by integrating observations from a tactical-grade *Litton LN-200* IMU and a *Javad Legacy* L1/L2 GNSS receiver (rover). The centimeter-level accurate GNSS solution was obtained by carrier-phase differential post-processing of the rover observation with a base GNSS receiver (*Topcon Hiper Pro*). Finally, the smoothed integrated navigation solution provided the trajectory which will serve as a reference as well as a base for emulation. We then emulated specific force and angular rate observations along this trajectory, and corrupted them with the real error signal observed in (static) calibration (see Section 7.2.3).

We are now interested in comparing versions of the constructed model expressed by Eq. (7.2.8), i.e.

- *Model 1*: $Y_k = W_k,\ k \in \mathbb{Z}$. This model is computationaly the most efficient but does not account for any correlated errors.
- *Model 2*: $Y_k = \sum_{m=1}^{M} (Y_{GM,m})_k + W_k,\ k \in \mathbb{Z},\ m = 1, \ldots, M$ with $M = 1$ for the gyroscopes, and $M = 2$ for the accelerometers. Referring to Eq. (7.2.10), the correlated gyroscope errors are modeled by $Y_{GM,1}$ with parameters $\hat{\sigma}^2_{GM,1}$ and $\hat{\beta}_1$. For the accelerometer, the correlated errors are handled by the sum of $Y_{GM,1}$ and $Y_{GM,2}$ whose associated parameters ($\hat{\sigma}^2_{GM,1}$, $\hat{\beta}_1$ and $\hat{\sigma}^2_{GM,2}$, $\hat{\beta}_2$) are given in Eq. (7.2.11).

- *Model 3*: complete model, i.e. with $M = 2$ for the gyroscopes and $M = 3$ for the accelerometers. This model accounts for errors with short and long correlation times given in Eq. (7.2.10) and (7.2.11), but is computationaly more demanding.

We introduce three artifical GNSS-free periods, each of 60 seconds duration. The first outage occurs approximately 7 minutes after mission start, the second after 11 minutes and the third after 15 minutes. The error growth during each outage can be visualized in Figure 7.8. Although not clearly visible in the figure, a general view reveals that *Model 2* performs only slightly better than *Model 1*. All models perform similarly during the first outage, meaning that some of the model parameters could not yet be correctly estimated by that time. However, after being longer in mission, *Model 3* clearly outperforms the two others. Regarding planimetric positioning error (upper left panel), the improvement of *Model 3* compared to *Model 1* (and *2*) is more than 30% for the second outage, and raises to around 60% at the third outage. The right panel of Figure 7.7 shows a close view on the trajectory during this third GNSS-free period. With respect to altimetric error, the upper right panel of Figure 7.8 indicates that *Model 3* always performed significantly better than *Model 1* and *2*. The lower left and right panels show the North-axis velocity (East and vertical velocity have similar behavior) and the roll angle (pitch and heading angles have similar behavior) errors, respectively. The same conclusion as for the planimetry can also be drawn for the accuracy of velocity and attitude.
Beside navigation, the EKF-estimated covariance matrix $\mathbf{P}_k$, $k \in \mathbb{Z}$ needs to be evaluated for each model. Figure 7.9 depicts the true latitude error and North-axis velocity error, considering that similar conclusions can be made from the other state variables. The estimated state variables during the third outage period are depicted as thick black (*Model 3*) and lightgray (*Model 1* and *2*) lines, surrounded by their estimated 3-σ error bounds (thiner lines with same color) extracted from the diagonal of $\mathbf{P}_k$. The true state is drawn as the thickest gray line. From both panels it can be seen that the estimated precision encompasses the true error only for the complete model, i.e. *Model 3*, while it is clearly underestimated with the two other models.

At this stage, the filter can be optimized with the following recommendations:

- The first Gauss-Markov process, i.e. $(Y_{GM,1})_k$, can safely be neglected, since it brings no significant improvement over *Model 1*.

- If *Model 1* or *Model 2* is chosen, the suppression of the model(s) accounting for long-term correlated errors, i.e. $(Y_{GM,1})_k$ and/or $(Y_{GM,2})_k$, may be compensated by augmenting the white noise level in W_k. This will improve the correspondance between the estimated filter precision with the real error.

Note that this example highlights again the difficulty of designing a filter using conclusions issued from a single realization. Although the recommendations listed above are certainly valid for the given scenario, they may change when the filter is submitted to a different dynamics (e.g. ground-based or pedestrian navigation). This indicates that a navigation filter should be designed according to a given operation context.

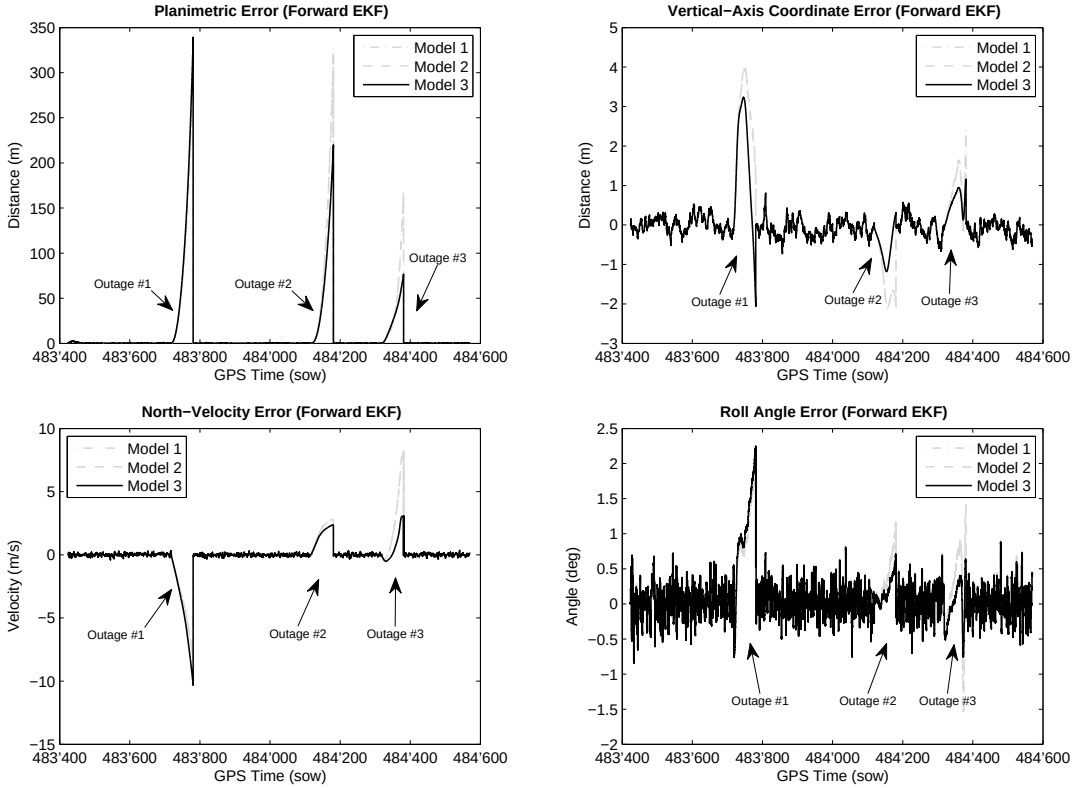

Figure 7.8: Navigation performance achieved by an *XSens MTi-G* device operated on an emulated trajectory with three GNSS-free periods. The upper panels show planimetric (left) and altimetric (right) positioning errors, while the lower panels depict North-axis velocity error (left) and roll angle error (right) for three tested models.

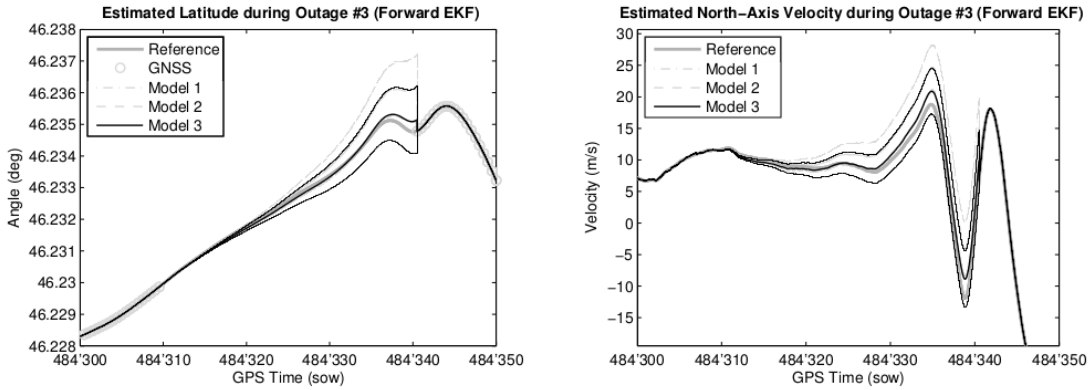

Figure 7.9: Estimated Latitude (left panel) and North-axis velocity (right panel) using three versions of the constructed model, together with the estimated 3-σ confidence levels.

7.4 Influence of Dynamics on Sensor Errors

In the preceding sections, we assumed no influence of the dynamics in the construction of the models. The sensor error signals were acquired in static laboratory conditions and the resulting models were hence validated for such conditions. In other words, these models accounted for the internal sensor errors, but not for the possible modifications caused by the motion (e.g. vibrations). Effects like vibrations or scale-dependent errors may further affect the sensor error behavior. Furthermore, (random) constant biases were not considered up to this moment. This section is dedicated to the analysis of sensor error signals acquired in moving conditions. We show how to construct these error signals and verify their potential dependency on vehicle dynamics.

7.4.1 Error Signal Construction

The sensor error signal is computed with respect to a navigation-grade IMU operated on the same platform as the sensor under test. Here, the tested IMU is the *XSens MTx* IMU which was previously studied. The IMU was mounted together with the *Ixsea AIRINS* navigation-grade IMU, considered as a true reference. Hence, the *XSens MTx* error signal generation requires the less straightforward definition and calibration of the following relationships: the spatial offset between both triads forming the IMUs, and the time alignment of both devices to a common reference. These two aspects will be treated in the sequel.

Space Alignment

Consider the reference IMU, noted R-IMU, and the IMU under study, noted S-IMU, rigidly mounted on the same platform. Assume that b-frame is equivalent to the R-IMU instrumental frame, and that the S-IMU provides observations in his instrumental s-frame. The b-frame and s-frame origins are separated by a vector $\mathbf{r}^b_{b\to s}$, called *inter-IMU leverarm*, and their relative orientation is expressed by the $\mathbf{C}^s_b$ DCM, called *inter-IMU boresight*. The relationship between $\mathbf{C}^s_b$ and the estimated boresight, denoted as $\hat{\mathbf{C}}^s_b$, may be expressed in terms of misalignment errors as

$$\mathbf{C}^s_b = (\mathbf{I} + \mathbf{\Psi})\,\hat{\mathbf{C}}^s_b \tag{7.4.1}$$

where $\mathbf{\Psi} = \left[\boldsymbol{\psi}^b_{b\to s}\times\right]$ contains the misalignment error angles $\boldsymbol{\psi}^b_{b\to s} = [\psi_1, \psi_2, \psi_3]^T$ between the b-frame and the s-frame.

The S-IMU observations $\boldsymbol{\omega}^s_{is}$ and $\mathbf{f}^s$ must be corrected by boresight $\mathbf{C}^s_b$ and lever-arm $\mathbf{r}^b_{b\to s}$ effects. Both quantities can either be known *a priori* or estimated. For the latter case, the estimation should ideally be done in laboratory conditions using controlled machines (e.g. a rotating table). However, for cases were no such infrastructure is available, $\mathbf{C}^s_b$ and $\mathbf{r}^b_{b\to s}$ may be estimated by using directly the vehicle on which the sensors are mounted. With this respect, two EKF-based estimation techniques will be presented here.

Boresight Estimation through Attitude Update This method, also used in Waegli [2009], estimates $\mathbf{C}_b^s$ in the S-IMU/GNSS filter by employing attitude updates provided by the R-IMU/GNSS processed solution. Indeed, the S-IMU/GNSS EKF can be fed with $\mathbf{z}_{\boldsymbol{\varphi}}$ measurements issued from the R-IMU/GNSS filter, i.e.

$$\mathbf{z}_{\boldsymbol{\varphi}} = \begin{bmatrix} z_r & z_p & z_y \end{bmatrix}^T \tag{7.4.2}$$

where z_r, z_p, z_y are roll, pitch and yaw observations, respectively. If again the same assumption of small ψ_1, ψ_2 and ψ_3 angles is done, the measurement model can be expressed as:

$$\mathbf{z}_{\boldsymbol{\varphi}} = h(\mathbf{x}_c) + \boldsymbol{v}_{\boldsymbol{\varphi}} \approx (\mathbf{I} - \boldsymbol{\Psi})\,\boldsymbol{\varphi} + \boldsymbol{v}_{\boldsymbol{\varphi}} \tag{7.4.3}$$

where $\mathbf{x}_c = \boldsymbol{\varphi}$ are the augmented states such that $\boldsymbol{\varphi} = [r, p, y]^T$ is the vector containing the attitude angles estimated in the S-IMU/GNSS EKF[1]. Linearized the model expressed in Eq. (7.4.3) yields the following observation design matrix:

$$\mathbf{H}_{\boldsymbol{\varphi}} = \begin{bmatrix} \mathbf{0}_{3\times6} \mid (\mathbf{I}_{3\times3} - \boldsymbol{\Psi}) \mid \mathbf{0}_{3\times6} \mid [\boldsymbol{\varepsilon}^l\times] \end{bmatrix}. \tag{7.4.4}$$

If the S-IMU is of poor quality, the solution provided by the S-IMU/GNSS filter may be considerably affected by the imperfections of initialization stage. Therefore, working with the R-IMU/GNSS filter as described further may be advantageous.

Boresight and Lever-arm Estimation through Inertial Measurement Aiding This method enables joint $\mathbf{C}_b^s$ and $\mathbf{r}_{b\to s}^b$ estimation by feeding the R-IMU/GNSS filter with the less precise $\boldsymbol{\omega}_{is}^s$ and $\mathbf{f}^s$ measurements. The relation between $\boldsymbol{\omega}_{is}^s$ and $\boldsymbol{\omega}_{ib}^b$ is given by

$$\boldsymbol{\omega}_{is}^s = \mathbf{C}_b^s \boldsymbol{\omega}_{ib}^b \tag{7.4.5}$$

which is true under the conditions that $\dot{\mathbf{C}}_b^s = 0$ and $\dot{\mathbf{r}}_{b\to s}^b = 0$. If $\boldsymbol{\Omega}_{is}^s = [\boldsymbol{\omega}_{is}^s\times]$ and $\boldsymbol{\Omega}_{ib}^b = \left[\boldsymbol{\omega}_{ib}^b\times\right]$, Eq. (1.5.2) enables to deduce the relation between $\mathbf{f}^b$ and $\mathbf{f}^s$ by considering the s-frame as the a-frame:

$$\mathbf{f}^s = \mathbf{C}_b^s \left(\mathbf{f}^b + \dot{\boldsymbol{\Omega}}_{ib}^b \mathbf{r}_{b\to s}^b + \boldsymbol{\Omega}_{ib}^b \boldsymbol{\Omega}_{ib}^b \mathbf{r}_{b\to s}^b\right) \tag{7.4.6}$$

in which $\dot{\boldsymbol{\Omega}}_{ib}^b \mathbf{r}_{b\to s}^b$ and $\boldsymbol{\Omega}_{ib}^b \boldsymbol{\Omega}_{ib}^b \mathbf{r}_{b\to s}^b$ represent centrifugal and Coriolis forces, respectively.

Under the condition that ψ_1, ψ_2 and ψ_3 are small, we have $\mathbf{C}_b^s \approx \mathbf{I} - \boldsymbol{\Psi}$. By denoting $\boldsymbol{\omega}_{is}^s$ and $\mathbf{f}^s$ respectively as $\mathbf{z}_{\boldsymbol{\omega}}$ and $\mathbf{z}_{\mathbf{f}}$, the linearized measurement models given in Eq. (7.4.5) and (7.4.6) are

$$\mathbf{z}_{\boldsymbol{\omega}} = h(\mathbf{x}_c) + \boldsymbol{v}_{\boldsymbol{\omega}} \approx \boldsymbol{\Omega}_{ib}^b \boldsymbol{\psi}_{b\to s}^b + \boldsymbol{v}_{\boldsymbol{\omega}} \tag{7.4.7}$$

and

$$\begin{aligned} \mathbf{z}_{\mathbf{f}} &= h(\mathbf{x}_c) + \boldsymbol{v}_{\mathbf{f}} \\ &\approx -\left([\mathbf{f}^b\times] + \boldsymbol{\Omega}_{ib}^b \boldsymbol{\Omega}_{ib}^b [\mathbf{r}_{b\to s}^b\times] + \dot{\boldsymbol{\Omega}}_{ib}^b [\mathbf{r}_{b\to s}^b\times]\right) \boldsymbol{\psi}_{b\to s}^b + \mathbf{C}_b^s \left(\boldsymbol{\Omega}_{ib}^b \boldsymbol{\Omega}_{ib}^b + \dot{\boldsymbol{\Omega}}_{ib}^b\right) \delta\mathbf{r}_{b\to s}^b + \boldsymbol{v}_{\mathbf{f}} \end{aligned} \tag{7.4.8}$$

[1] These values are obtained from $\mathbf{q}_b^l$ estimated in the state vector.

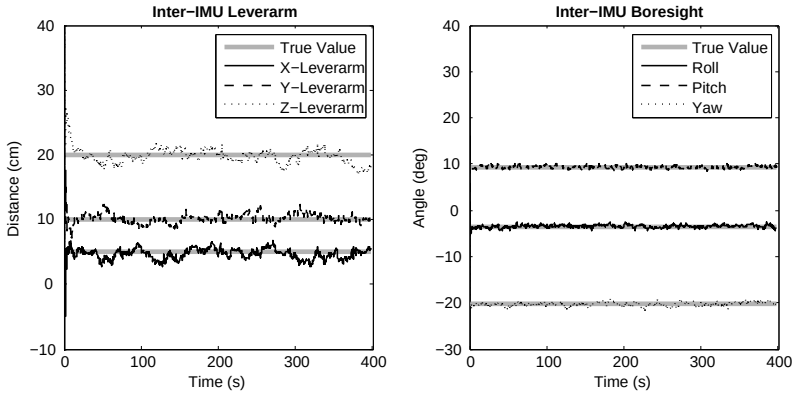

Figure 7.10: Result of a joint inter-IMU leverarm (left panel) and boresight (right panel) estimation using emulated IMUs. The true values are shown as gray lines.

where

$$\mathbf{x}_c = \left[\begin{array}{cc} \left(\boldsymbol{\psi}^b_{b\to s}\right)^T & \left(\mathbf{r}^b_{b\to s}\right)^T \end{array} \right]^T \tag{7.4.9}$$

are the augmented calibration states. The design matrix $\mathbf{H_f}$ for the whole state vector can be deduced from Eq. (7.4.7) and (7.4.8), yielding

$$\mathbf{H_f} = \left[\begin{array}{c|c|c} \mathbf{0}_{3\times 15} & \boldsymbol{\Omega}^b_{ib} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 15} & -[\mathbf{f}^b\times] - \boldsymbol{\Omega}^b_{ib}\boldsymbol{\Omega}^b_{ib}[\mathbf{r}^b_{b\to s}\times] - \dot{\boldsymbol{\Omega}}^b_{ib}[\mathbf{r}^b_{b\to s}\times] & \mathbf{C}^s_b\left(\boldsymbol{\Omega}^b_{ib}\boldsymbol{\Omega}^b_{ib} + \dot{\boldsymbol{\Omega}}^b_{ib}\right) \end{array} \right]. \tag{7.4.10}$$

An illustrative example is shown in Figure 7.10 in which the boresight angles between an emulated low-grade and a tactical-grade IMU has been estimated using inertial measurements of the former to feed the EKF of the latter (forward filtering solution). The true boresight parameters $\mathbf{r}^b_{b\to s}$ and $\mathbf{C}^p_b$ are depicted as thick gray lines. The augmented states $\boldsymbol{\psi}^b_{b\to s}$ and $\mathbf{r}^b_{b\to s}$ are driven by the following Gauss-Markov processes:

$$\dot{\boldsymbol{\psi}}^b_{b\to s} = \boldsymbol{\beta}_{\boldsymbol{\psi}}\boldsymbol{\psi}^b_{b\to s} + \mathbf{w}_{\boldsymbol{\psi}} \tag{7.4.11}$$

$$\dot{\mathbf{r}}^b_{b\to s} = \boldsymbol{\beta}_{\boldsymbol{r}}\mathbf{r}^b_{b\to s} + \mathbf{w}_{\boldsymbol{r}} \tag{7.4.12}$$

where $\boldsymbol{\beta}_{\boldsymbol{\psi}}$ and $\boldsymbol{\beta}_{\boldsymbol{r}}$ are (3×1) vectors containing the inverse correlation times set to large values (i.e. are practically random constants), and $\mathbf{w}_{\boldsymbol{\psi}}$ and $\mathbf{w}_{\boldsymbol{r}}$ are the PSD levels of the processes associated to the boresight and leverarm states, respectively. The filter was initialized with null values for both, boresight and leverarm states. The convergence time to the true values last a few seconds only. Figure 7.11 depicts the estimated smoothed inter-IMU angles $\boldsymbol{\psi}^b_{b\to s_i}$ with $i = 1, 2$ for two *XSens MTx* IMUs (measuring in their respective s_i instrumental frame) with respect to the reference signals provided by a navigation-grade IMU (*Ixsea Airins*). The data were collected on a vehicle during a 15 minutes long trajectory. The final boresight angles are estimated as a weighted mean (considering the smoothed variances).

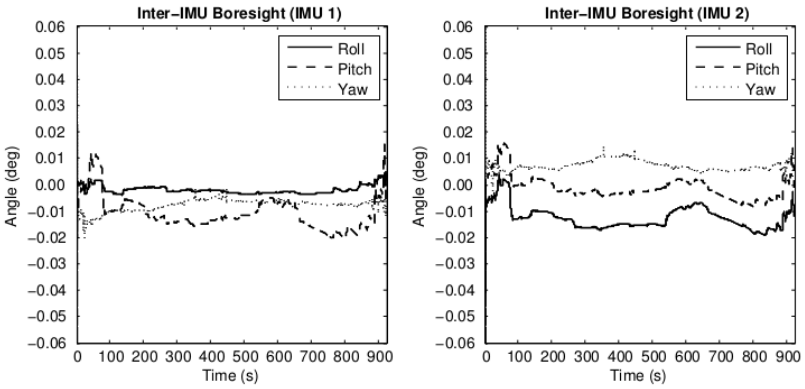

Figure 7.11: Result of the inter-IMU boresight angle estimation for two *XSens MTx* IMUs using inertial measurements from an *Ixsea Airins* navigation-grade IMU.

Time Alignment

The samples of the R-IMU and S-IMU shall be tied to a common absolute time frame. For that, GPS time can be used. We will see later in Chapter 8 (Section 8.6) how inter-IMU time synchronization can be realized.

Once the data are aligned in space and time, the dynamic sensor error is considered as the difference between the transformed S-IMU signals and the corresponding R-IMU signals.

7.4.2 Dynamic Error Signal Analysis for MEMS-Based IMUs

The *XSens MTx* gyroscope and accelerometer error signals obtained by the described methodology are shown in the upper two panels of Figure 7.12. Note that only the Z-axis is shown, since similar results were obtained for the remaining two axes. A first view on the gyroscope error signal (left panel) reveals no clear dependency on dynamics. This gives confidence in the estimated boresight angles as well as the time synchronization between both IMUs.

However, in Figure 7.13 we plotted the angular rate errors against the angular rate (upper left panel), the angular acceleration $\dot{\boldsymbol{\omega}}_{ib}^{b}$ (upper right panel), and the angular jerk $\ddot{\boldsymbol{\omega}}_{ib}^{b}$ (lower left panel). Although there is no dependency of the gyroscope errors with angular rate, there is some correlation with angular acceleration and jerk since a trend can be identified from the graphs.

We now compare the structure of the gyroscope error signal acquired in dynamic conditions with its counterpart signal acquired in static conditions whose wavelet variance was drawn in the lower left panel of Figure 7.5 (black circles). As mentioned in Section 7.2.3, a unique model can be assumed for all three axes of the *XSens MTx* IMU. The resulting model is of the type given in Eq. (7.2.8) with $M = 2$ and its parameter values together with the associated 95% confidence intervals are listed in Table 7.1 in the *static model* column. We build a model on the dynamic error signal using the GMWM estimator with the intention of comparing the

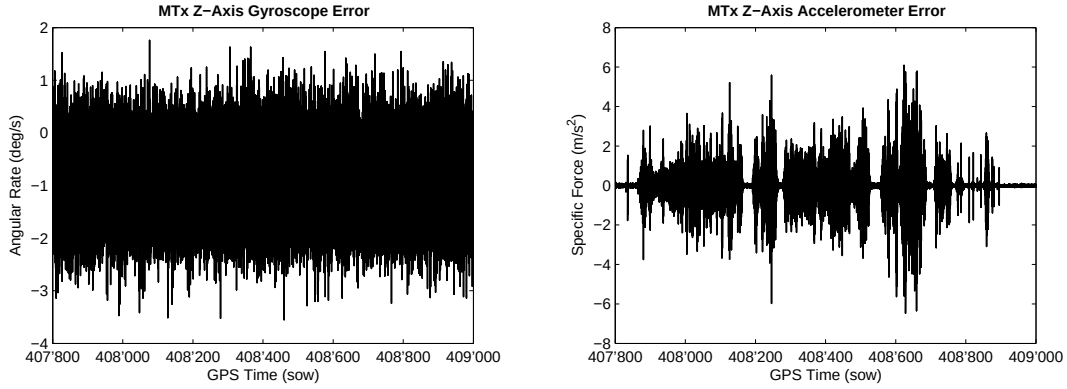

Figure 7.12: Typical *XSens MTx* gyroscope (left panel) and accelerometer (right panel) error signals computed from the reference IMU under dynamics.

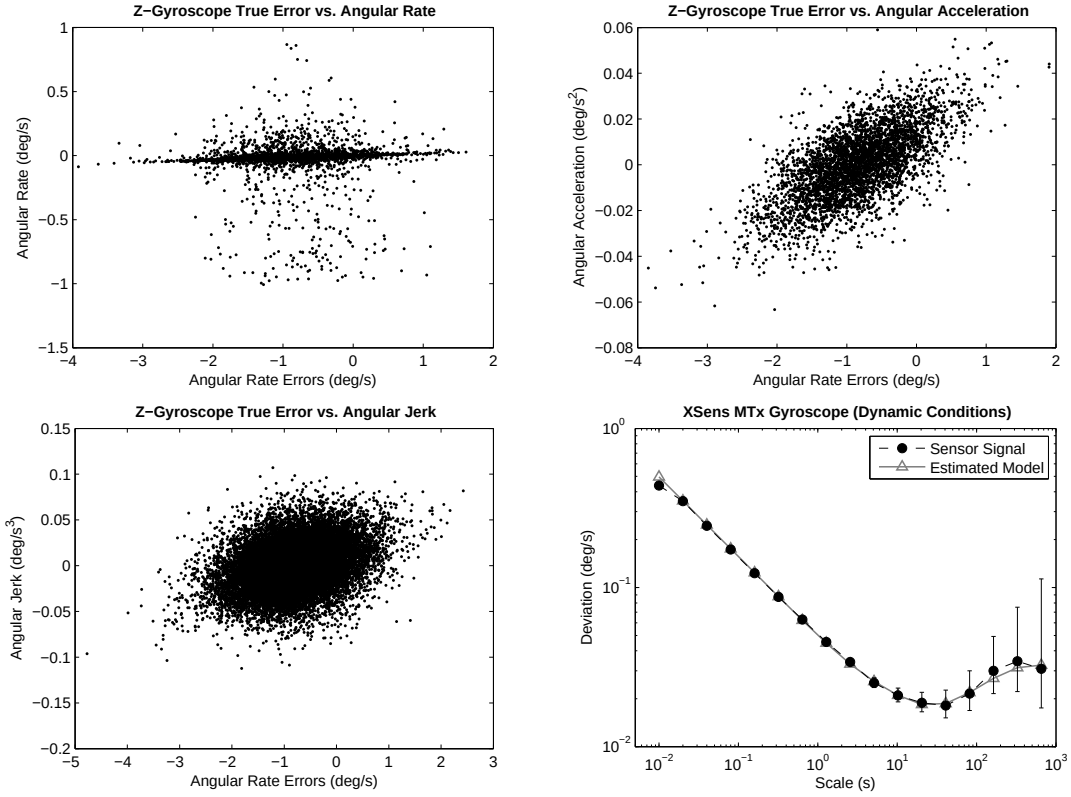

Figure 7.13: Correspondance between the "true" error signal constructed for a *XSens MTx* gyroscope under dynamics, and angular rate (upper left panel), angular acceleration (upper right panel), and angular jerk (lower left panel). The lower right panel depicts the wavelet variance sequence computed on the gyroscope error signal acquired in moving conditions (black circles), together with the wavelet variance issued from an estimated model (see Table 7.1).

Table 7.1: Comparison between the *XSens MTx* gyroscope model constructed on a signal acquired in non-moving conditions (static model) and in moving conditions (dynamic model).

Process	**Parameter**	**Unit**	**Static Model**	**Dynamic Model**
White Noise	σ^2_{WN}	$(\text{deg/s})^2$	0.50552 ± 0.00035	0.65382 ± 0.00001
Gauss-Markov	β_1	$1/\text{s}$	0.00492 ± 0.01892	0.00155 ± 0.00082
	$\sigma^2_{GM,1}$	$(\text{deg/s})^2$	0.00142 ± 0.00001	0.00981 ± 0.00005
Gauss-Markov	β_2	$1/\text{s}$	113.51523 ± 0.00404	1.68012 ± 0.04980
	$\sigma^2_{GM,2}$	$(\text{deg/s})^2$	0.05528 ± 0.00051	0.00168 ± 0.00005

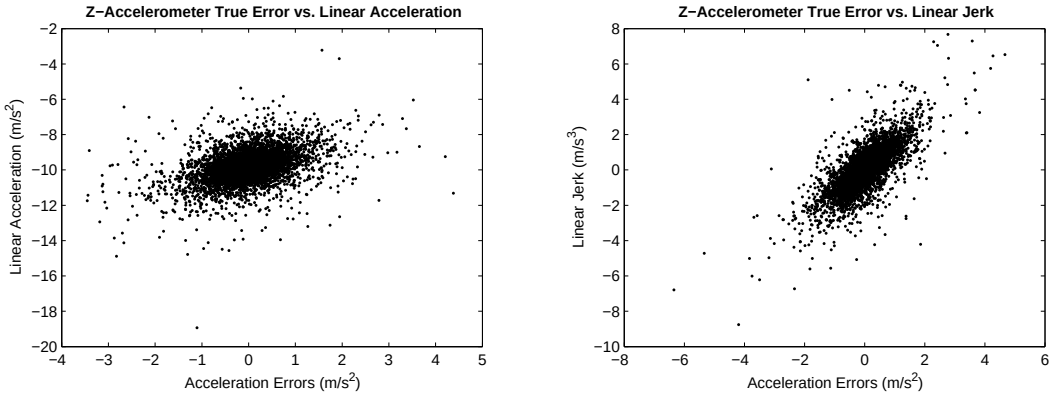

Figure 7.14: Typical correspondance between *XSens MTx* specific force error (with respect to reference IMU), and linear acceleration (left panel) and jerk (right panel).

structure of the noise and its level with the static model. The result of the modeling is shown in the lower right panel of Figure 7.13 in which we draw the wavelet variance curve of the gyroscope dynamic error signal (dashed black line with black circles) and the wavelet variance of the estimated model (gray curve). The model which best suites the signal is of the same type as the model defined in Eq. (7.2.8). After trying several values of M, the best results (in terms of goodness-of-fit) were again obtained with $M = 2$. In Table 7.1, we show the values of the estimated model parameters for the model constructed in dynamic conditions in the *dynamic model* column. Comparing both columns indicates that the dynamics has not changed significantly the gyroscope noise structure, i.e. that the filter can run with the same error model, independently from the vehicle dynamics. However, the magnitude of the Gaussian white noise level as well as the Gauss-Markov driving noise level change significantly. This is also the case for the inverse correlation times of both Gauss-Markov processes which differ significantly in magnitude. However, this difference could be well caused by the relatively short dynamic error signal used to compute the wavelet variance. Regarding the accelerometer error, the right panel of Figure 7.12 clearly indicates the effect of the platform vibrations through important variations of the error levels. Obviously, computing the wavelet variance on such a signal makes not much sense. In Figure 7.14, we plotted the accelerometer errors against the linear acceleration (left panel) and linear jerk (right panel). Similarly to the gyroscope, a clear correlation between the error signal and the linear acceleration and jerk can be read from the trend in both panels.

The results obtained in this section confirm the work of Wis and Colomina [2010] in which a similar preliminary analysis was performed using a *IMAR INAV-FJI-IDEG-00* reference IMU for studying a *Litton LN-200* IMU and a *Systron Donner Motionpak II* MEMS-based IMU. They proposed then introducing jerk components in the sensor error models, i.e in $\Delta\boldsymbol{\omega}_{ib}^{b}$ and $\Delta\mathbf{f}^{b}$ in our case. In this research, we will not deal with such kinds of models. Instead, we will use the approach of trying to estimate the true (dynamical) errors by introducing redundancy in the navigation system through the use of multiple inertial sensors operating at the same time on the same platform. From these redundant sensors, residuals can be observed, estimated and included in an adaptive filtering scheme, for example. This will be the subject treated in the third part of this thesis.

7.5 Conclusion

In this second part of the thesis, we mainly focused on the navigation filter design task, i.e. the modeling of sensor errors. Its importance in terms of final navigation accuracy has been proved using simulated and real data sets. The severe limitations of the existing methods for modeling stochastic errors have been highlighted and two new methodes were proposed in this respect. The first is an adaptation of the likelihood-based EM algorithm to the context of inertial sensors. We showed how to modify the classical (unconstrained) form of the EM algorithm for estimating models such as first-order Gauss-Markov processes. This method enabled to estimate models for which the classical Allan variance technique could not be applied. However, computational drawbacks as well as convergence to wrong solutions inherent to likelihood-based methods are the main drawbacks of this approach. Then, we developed a new theory and implemented a completely new estimation framework, called GMWM, for which we proved the asymptotic consistency. This new method enabled to estimate very complex stochastic models such as composite stochastic processes in a few seconds only. Moreover, confidence intervals could be computed for each estimated parameter using this approach. Finally, we attempted to analyse the influence of platform dynamics on the sensor error behavior. For that, we built a complex experimental setup for generating error signals acquired while the vehicle was in motion. The analysis revealed the dependency of the error to some dynamics quantities such as linear and angular jerk. Although we demonstrated that a single error model accounting for static and dynamic conditions could be afforded for the gyroscopes, the accelerometer error level varied significantly with dynamics mainly due to parasite signals such as vibrations. This motivated the attempt to estimate this error signal in dynamic conditions by using multiple inertial sensors. This will be the subject of the next part of the thesis.

Part III

Redundancy in Inertial Observation

Chapter 8

Estimation Concepts with Redundant Inertial Sensors

8.1 Introduction

The investigations made in Section 7.4 indicated the influence of the encountered dynamics on the sensor error behavior. Parasite signals affecting the platform motion (e.g. vibrations) make the sensor noise level vary in time (mainly for accelerometers). These nondesired signals cannot be investigated in laboratory conditions as they vary according to the type of vehicle and the environmental conditions. However, the use of multiple inertial sensors experiencing the same conditions introduces the necessary measurement redundancy into the system for estimating these types of effects.

Redundancy in inertial sensors improves the navigation performance on several levels:

1. The level of sensor noise can be estimated directly from the data and provide hence a better view on the reality.
2. The noise level of the overall system can be reduced.
3. Defective sensors can be detected and isolated via *Fault Detection and Isolation* (FDI) procedures.
4. A more accurate navigation solution (mainly the attitude) can be achieved.

With respect to the first and second point, the recent investigations of Stebler et al. [2011b], Waegli et al. [2010] developed modeling concepts accounting for varying noise levels. The authors were mainly motivated by the fact that stochastic errors may vary as a function of environmental conditions applied to the sensors, i.e. temperature, electrical power, magnetic fields Guerrier [2009], or the dynamics. To which extent this statement is realistic can be deduced from Figure 8.1. This figure depicts (in gray) the true *XSens MTx* accelerometer (left panel) and gyroscope (right panel) error signal constructed in Section 7.4. The varying noise level can be deduced from the plotted residuals (i.e. deviations from the estimated mean signal using multiple *MTx* devices by a process discussed later in the chapter) shown as a black line

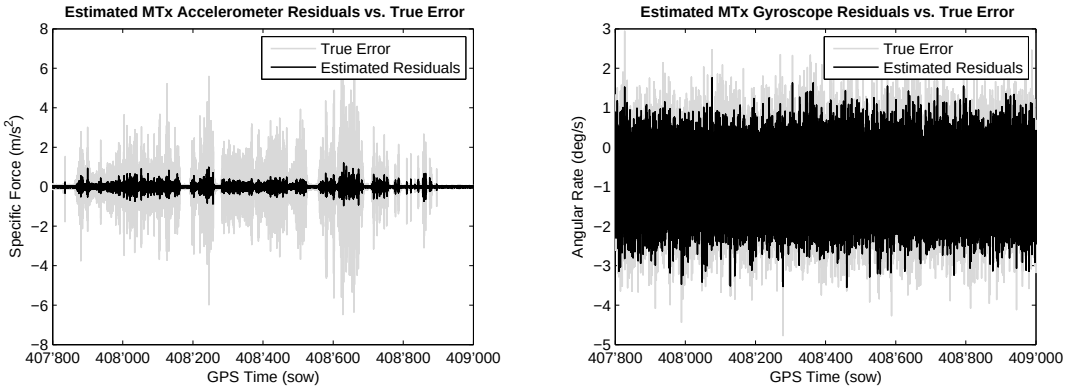

Figure 8.1: Comparison between the residuals estimated from three *XSens MTx* IMUs, and the true acceleration error (left panel) and angular rate error (right panel) with respect to a reference IMU.

in the respective panels. Two preliminary statements can be brought at this point. First, the magnitude of the estimated noise level is realistic in the case of the gyroscope but not so for the accelerometers whose output is prone to vibrations. Second, the general structure of the error signal is clearly captured by both estimated residuals signals. Although less visible, the confirmation of the similarity between the gyroscope true error and the estimated residuals is brought in the left panel of Figure 8.2. In this panel, the estimated residuals are plotted against the true error signal. A clear correlation can be deduced from the trend of the cloud. Regarding these observations, Guerrier [2009] proposed two methods. The first assumed that all sensors have the same variance which remains constant over a certain time interval. The second used ARMA$(\cdot,\cdot)$ models, introduced in Chapter 4 (Section 4.7), for modeling the correlated part of the noise, and *Generalized Auto-Regressive Conditional Heteroskedasticity* (GARCH) models for continuously estimating the residual variance for each individual sensor. Both strategies enabled to compute a so-called *synthetic IMU* (sometimes also termed *virtual IMU*) by introducing measurement weights for reducing the noise level in the resulting system.

With respect to the previously mentioned third advantage, Guerrier et al. [2012] also investigated several FDI procedures applied on MEMS-based redundant IMUs. Guerrier [2009], Guerrier et al. [2012] demonstrated the conditions for optimal configuration of multiple IMU triads. Note that already in 1974, Pejsa [1974], followed by Sturza [1988a,b] in the late eighties, investigated FDI procedures by studying optimal geometries for the sensor spatial distribution.

The last point (4.) was extensively studied by Bancroft [2009], Waegli et al. [2008], Waegli [2009] who tested different INS/GNSS integration schemes based on redundant low-grade (MEMS-based) IMUs. The tested schemes were initially proposed in 2003-2004 by Colomina et al. [2003, 2004a,b] who developed several levels on which redundancy can be generated in inertial navigation, mainly for the fields of remote sensing and photogrammetry in which more precise inertial sensors are operated. Note also that a review of multisensor fusion methodologies (mainly for aircraft navigation systems) dated from 2005 can be found in Allerton and Jia [2005].

The content of this chapter stems mainly from the work of Colomina et al. [2004a,b] where different architectures for realizing and processing redundant inertial data are proposed. We

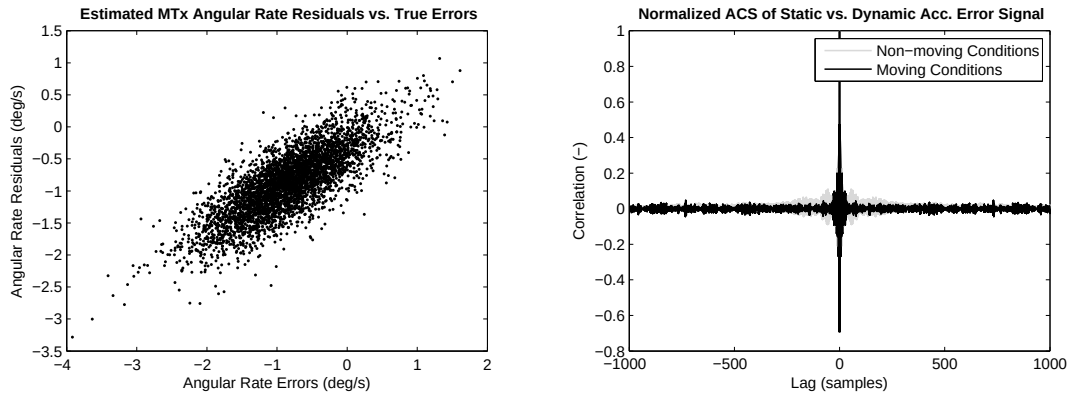

Figure 8.2: Estimated angular rate residuals versus angular rate error (left panel), and autocorrelation sequence of the accelerometer error computed on a static (gray curve) and dynamic (black curve) portion of the signal.

start from these redundancy schemes and go further by developing each of them while paying special attention to implementation and operational aspects. In Section 8.2 we expose the general frame of inertial redundancy according to Colomina et al. [2004a,b] before moving to Section 8.3 in which we describe what assumptions we made in this work. Both Sections 8.4 and 8.5 investigate algorithms working at two different levels of redundancy. Following the software aspects, we describe the hardware realization in Section 8.6 which offers a smooth transition to Chapter 9 dedicated to the operation and tests of these concepts. Note that we do not treat FDI aspects in this thesis and refer to Guerrier et al. [2012] for more details about this topic.

8.2 Redundancy in Inertial Navigation

A conventional IMU contains six sensors to measure six unknowns: three collocated accelerometers and three gyroscopes, arranged in an orthogonal triad. With this respect, redundancy can be realized at two levels Colomina et al. [2004a,b]:

- More than three gyroscopes and three accelerometers are packaged in a single unit. Such a type of unit will be referred to as *redundant IMU* (RIMU). The relative orientation between the individual sensors and the instrumental frame is assumed to be known.
- Several conventional IMUs are operated at the same time on the same platform. We will designate this type of redundancy by *distributed IMUs* (DIMU). However, issues such as relative spatial alignment, i.e. leverarm and orientation, and time alignment between the individual IMUs have to be considered.

RIMU or DIMU provide data which have to be processed with algorithms accounting for redundancy. Basically, redundancy can be treated at two stages Colomina et al. [2004a,b]:

- At *observation space*: the observations are blended in order to construct an imaginary, non-redundant IMU aligned to an instrumental frame. The resulting data, named *synthetic*

IMU, can be treated using conventional mechanization algorithms described in Chapters 1 and 2.

- At *state space*: the navigation software has to be modified to account for the redundant observations. Regarding this, there are two possibilities. Either the single-IMU mechanization equations developed in Chapters 1 and 2 are modified to account for the redundant observations, or the filter state is augmented to model the relative geometry between the individual IMUs. The first possibility will be called the *extended mechanization* and assumes knowledge of the relative geometry. Hence it can be applied on data issued from RIMU or DIMU. The second possibility will be designated as the *geometrically-constrained navigation* which is of particular interest for DIMU with poorely known relative orientation.

Now once the big picture on the inertial redundancy concept has been underlined, the general assumptions made throughout the third part of the thesis can be described.

8.3 Assumptions

Consider R IMUs rigidly mounted on a common platform and time-synchronized with respect to a common time-base. The associated instrumental frame of the i-th IMU is designated as the s_i-frame, for $i = 1, \ldots, R$. The specific force and angular rate measurements provided by the i-th IMU are noted as $\mathbf{f}^{s_i}$ and $\boldsymbol{\omega}_{is_i}^{s_i}$, respectively.

We make the following assumptions:

1. Between each pair of sensors (x, y), we assume

$$\begin{aligned} \dot{\mathbf{r}}_{x\to y}^{x} &= 0 \\ \dot{\mathbf{C}}_{x}^{y} &= 0 \end{aligned} \tag{8.3.1}$$

   with $x, y \in \{s_i : i = 1, \ldots, R\}$ such that $x \neq y$. The quantity $\mathbf{r}_{x\to y}^{x}$ is the leverarm vector relating the origin of frame x to frame y expressed in frame x, and $\mathbf{C}_{x}^{y}$ expresses the relative orientation of frame x with respect to frame y. In other words, Eq. (8.3.1) states that all the conventional IMUs composing the DIMU are mounted on a common rigid platform, meaning that the relative alignment between the individual IMUs is not varying in time. By definition, the conditions in Eq. (8.3.1) are always satisfied for RIMUs.

2. All the conventional IMUs forming the DIMU are of the same type. This means that they contain accelerometers and gyroscopes of the same type and hence are *theoretically* expected to provide identical performance. Similarly, this condition should always be fullfilled in RIMUs.

3. All the conventional IMUs forming the DIMU have been subject to calibration procedures accounting for systematic effects due to internal axis-misalignments or temperature dependencies. In other words, $\mathbf{N}_\bullet$, and the deterministic parts in $\mathbf{b}_\bullet$ and $\mathbf{S}_\bullet$ in Eq. (1.4.3) and (1.4.4) have been estimated (see Section 1.4). Hence, data coming from each conventional IMU are supposed to be free of these effects.

8.4 Redundancy in Observation Space

Inertial data provided by the conventional IMUs are brought into a synthetic IMU computer. This computer is in charge of processing an IMU (with three orthogonal axes) whose virtual instrumental frame is aligned with b-frame. Let $\boldsymbol{a}_k$, $\boldsymbol{g}_k$ be $(3R \times 1)$ vectors containing specific force and angular rate signals provided at digital times $\{t_k : k \in \mathbb{Z}\}$, respectively, by R orthogonal triads of accelerometers and gyroscopes, i.e.

$$\boldsymbol{a}_k = \begin{bmatrix} \mathbf{f}_k^{s_1} \\ \vdots \\ \mathbf{f}_k^{s_R} \end{bmatrix} \quad \text{and} \quad \boldsymbol{g}_k = \begin{bmatrix} (\boldsymbol{\omega}_{is_1}^{s_1})_k \\ \vdots \\ (\boldsymbol{\omega}_{is_R}^{s_R})_k \end{bmatrix} \tag{8.4.1}$$

where $\mathbf{f}_k^{s_i}$ and $(\boldsymbol{\omega}_{is_i}^{s_i})_k$ for $i = 1, \ldots, R$ are the specific force and angular rate of the i-th IMU. The synthetic accelerometer and gyroscope data vectors at time t_k, respectively noted $\mathbf{f}_k^b$ and $(\boldsymbol{\omega}_{ib}^b)_k$, have size (3×1) and can be constructed using

$$\begin{aligned} \mathbf{f}_k^b &= \boldsymbol{\Pi}_a \boldsymbol{a}_k \\ (\boldsymbol{\omega}_{ib}^b)_k &= \boldsymbol{\Pi}_g \boldsymbol{g}_k \end{aligned} \tag{8.4.2}$$

where $\boldsymbol{\Pi}_a$ and $\boldsymbol{\Pi}_g$ in Eq. (8.4.2) are orthogonal projectors given by

$$\boldsymbol{\Pi}_\bullet = \left(\mathbf{H}_\bullet^T \mathbf{W}_\bullet \mathbf{H}_\bullet\right)^{-1} \mathbf{H}_\bullet^T \mathbf{W}_\bullet. \tag{8.4.3}$$

in which symbol $\bullet$ can either be a (accelerometers) or g (gyroscopes). The matrix $\mathbf{H}_\bullet$ rotates the data from the individual triads to the b-frame:

$$\mathbf{H}_\bullet = \begin{bmatrix} \mathbf{C}_{b(1)}^{s_1} \\ \vdots \\ \mathbf{C}_{b(R)}^{s_R} \end{bmatrix}, \tag{8.4.4}$$

assuming that the gyroscopes and accelerometers are mounted on the same triads, i.e. $\mathbf{H}_a = \mathbf{H}_g$. The matrix $\mathbf{W}_\bullet$ is a matrix which weights the individual devices in the computation of the synthetic IMU. The resulting synthetic gyroscope and specific force observations, denoted by $\boldsymbol{\omega}_{ib}^b$ and $\mathbf{f}^b$, respectively, can then be introduced in the conventional (i.e. single-IMU based) navigation filter.

In the sequel, we provide a generalized description of the notion of synthetic IMU as suggested in Colomina et al. [2004a] and go further by proposing two methods of computing weights $\mathbf{W}_\bullet$.

8.4.1 Synthetic IMU

Let $\{(Y_{i,j})_k \ : k \in \mathbb{Z}\,; i = 1,\ldots,R\,; j = 1,2,3\}$ be a random variable corresponding to the signal recorded at digital times $\{t_k : k \in \mathbb{Z}\}$ by sensors i on the axis j and let $\{(y_{i,j})_k : k = 1,\ldots,N\,; i = 1\ldots,R\,; j = 1,2,3\}$ denote its realization. Therefore, $(y_{i,j})_k$ is any element of either vector $\boldsymbol{a}_k^b$ or $\boldsymbol{g}_k^b$ which are the vectors containing the *rotated*[1] individual IMU observations, noted $\mathbf{f}_k^{b(i)}$ and $(\boldsymbol{\omega}_{ib}^{b(i)})_k$, i.e.

$$\boldsymbol{a}_k^b = \begin{bmatrix} \mathbf{f}_k^{b(1)} \\ \vdots \\ \mathbf{f}_k^{b(R)} \end{bmatrix} \quad \text{and} \quad \boldsymbol{g}_k^b = \begin{bmatrix} (\boldsymbol{\omega}_{ib}^{b(1)})_k \\ \vdots \\ (\boldsymbol{\omega}_{ib}^{sb(R)})_k \end{bmatrix}. \tag{8.4.5}$$

A quite natural error model for $(Y_{i,j})_k$ can be written as

$$(y_{i,j})_k = \mu_{j,k} + (\varepsilon_{i,j})_k \tag{8.4.6}$$

where $\mu_{j,k}$ denotes the true signal that should be observed at digital time t_k on the j-th axis of the s_i-frame, and $(\varepsilon_{i,j})_k$ the corresponding error. If we assume that $(\varepsilon_{i,j})_k \overset{iid}{\sim} \mathcal{N}(0,\sigma^2)$, then a natural estimator for $\mu_{j,k}$ is simply

$$\hat{\mu}_{j,k} = \frac{1}{R}\sum_{i=1}^{R}(y_{i,j})_k \tag{8.4.7}$$

and we have that

$$\hat{\mu}_{j,k} \sim N\left(\mu_{j,k}, \frac{\sigma^2}{R}\right). \tag{8.4.8}$$

Of course, $\hat{\boldsymbol{\mu}}_k = [\hat{\mu}_{1,k},\ \hat{\mu}_{2,k},\ \hat{\mu}_{3,k}]^T$ is equivalent to $(\boldsymbol{\omega}_{ib}^b)_k$ or $\mathbf{f}_k^b$ from Eq. (8.4.2). Eq. (8.4.7) computes an *unweighted synthetic IMU* by simply averaging the i accelerometers and i gyroscopes by assigning equal weights to each individual sensor.

However, the investigations in Stebler et al. [2011b], Waegli et al. [2010] revealed that the assumption of giving equal weights to all sensors is not realistic since the noise power can vary across sensors and in time. This was confirmed in the left panel of Figure 8.1 showing the accelerometer error signal acquired in dynamical conditions (see Chapter 7). It clearly reveals that noise level is non-constant.

This simple model expressed in Eq. (8.4.8) can be extended in various ways. For example, if we relax the assumption that the sensors have the same variance, and hence the same weight in the computation of $\hat{\mu}_{j,k}$, and assume that $(\varepsilon_{i,j})_k \overset{iid}{\sim} \mathcal{N}(0,\sigma_i^2)$, the estimation of $\mu_{j,k}$ becomes far more complex. Let $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^{3N+R}$ be defined as

$$\boldsymbol{\theta} = \left[\ \mu_{1,1} \quad \ldots \quad \mu_{3,N} \ \middle|\ \gamma_1 \quad \ldots \quad \gamma_R\ \right]^T \tag{8.4.9}$$

[1] The sensor frame of each IMU is supposed to be aligned with the b-frame using $\mathbf{r}_{x\to y}^x$ and $\mathbf{C}_x^y$ which are either known or were previously estimated.

where, for notational simplicity, $\gamma_i = \sigma_i^2$. The problem of computating a synthetic IMU in which the individual sensors are weighted according to their noise level requires the estimation of $\boldsymbol{\theta}$. The MLE for $\boldsymbol{\theta}$ is defined as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}}\, \ell\left(\boldsymbol{\theta}|\mathbf{y}\right). \tag{8.4.10}$$

Unfortunately, the maximization problem expressed by Eq. (8.4.10) is numerically challenging and therefore often difficult to perform in practice. However, by making the additional assumption that the function $-\ell\left(\boldsymbol{\theta}|\mathbf{y}\right)$ is convex in $\boldsymbol{\theta}$ (i.e. there is a unique minimum and no local maxima), we may solve Eq. (8.4.10) iteratively using the *coordinate descent algorithm* (see e.g. Brent [2002]). This procedure finds the minimum of a function by line search along each coordinate direction at the current point in each iteration. This implies that we will solve iteratively the following problems:

$$\begin{aligned}
\hat{\mu}_{1,1}^{(m)} &= \underset{\mu_{1,1}\in\mathbb{R}}{\operatorname{argmax}}\, \ell\left(\mu_{1,1}|\mathbf{y}, \hat{\mu}_{1,2}^{(m-1)}, ..., \hat{\mu}_{3,N}^{(m-1)}, \hat{\gamma}_1^{(m-1)}, ..., \hat{\gamma}_R^{(m-1)}\right) \\
&\vdots \\
\hat{\mu}_{3,N}^{(m)} &= \underset{\mu_{3,N}\in\mathbb{R}}{\operatorname{argmax}}\, \ell\left(\mu_{3,N}|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, ..., \hat{\mu}_{3,N-1}^{(m)}, \hat{\gamma}_1^{(m-1)}, ..., \hat{\gamma}_R^{(m-1)}\right) \\
\hat{\gamma}_1^{(m)} &= \underset{\gamma\in\mathbb{R}^+}{\operatorname{argmax}}\, \ell\left(\gamma_1|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, ..., \hat{\mu}_{3,N}^{(m)}, \hat{\gamma}_2^{(m-1)}, ..., \hat{\gamma}_R^{(m-1)}\right) \\
&\vdots \\
\hat{\gamma}_R^{(m)} &= \underset{\gamma\in\mathbb{R}^+}{\operatorname{argmax}}\, \ell\left(\gamma_1|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, ..., \hat{\mu}_{3,N}^{(m)}, \hat{\gamma}_1^{(m)}, ..., \hat{\gamma}_{R-1}^{(m)}\right)
\end{aligned} \tag{8.4.11}$$

were $m = 0, 1, 2, \ldots$ is the mth iteration. Moreover, it is easy to show that for $k = 1, \ldots, N$, axes $j = 1, 2, 3$ and sensors $i = 1, \ldots, R$ we have

$$\begin{aligned}
\hat{\mu}_{j,k}^{(m)} &= \frac{\sum_{i=1}^{R} \frac{(y_{i,j})_k}{\hat{\gamma}_i^{(m-1)}}}{\sum_{i=1}^{R} \frac{1}{\hat{\gamma}_i^{(m-1)}}}, \\
\hat{\gamma}_i^{(m)} &= \frac{1}{3N} \sum_{j=1}^{3} \sum_{k=1}^{N} \left((y_{i,j})_k - \hat{\mu}_{j,k}^{(m)}\right)^2.
\end{aligned} \tag{8.4.12}$$

Eq. (8.4.12) expresses the computation of a synthetic averaged sensor $\hat{\mu}_{j,k}^{(m)}$ in which each individual device is weighted by $\hat{\gamma}_i^{(m)}$. We will designate such a type of IMU as a *weighted Synthetic IMU*. As $\hat{\mu}_{j,k}^{(m)}$ only depends of the estimated variances from the previous iteration, we only have to initialize the values in $\gamma_i^{(0)}$. A natural choice for that is $\gamma_i^{(0)} = \gamma_0$ for $i = 1, ..., R$ in which case $\tilde{\mu}_{j,k}^{(0)}$ reduces to $\hat{\mu}_{j,k}$ defined in Eq. (8.4.7), i.e. the unweighted synthetic IMU. This iterative procedure guarantees to obtain the MLE provided the function $-\ell\left(\boldsymbol{\theta}|\mathbf{y}\right)$ is convex. In the sequel, we study two forms of weighted synthetic IMU computers in which we not only assume that the weights vary according to the sensor, but also with time (i.e. $\sigma_{i,k}^2$).

8.4.2 ARMA-GARCH based Synthetic IMU Computer

In the second part of this thesis (see also El-Sheimy et al. [2008], Guerrier [2009], Hou [2004], Waegli et al. [2010]), we demonstrated the presence of colored noise in MEMS-based inertial sensors. This directly contradicts that $(\varepsilon_{i,j})_k \overset{iid}{\sim} \mathcal{N}(0, \sigma_i^2)$. In the case of inertial sensors, ARMA$(\cdot,\cdot)$ models seem to quite well approximate the autocorrelated noise components when forming the synthetic observations Nassar [2003], Nassar et al. [2004], Park and Gao [2006]. Nevertheless, it was shown in Waegli et al. [2010] that, when applied on inertial sensor error signals, ARMA$(\cdot,\cdot)$ residuals are not *iid* distributed as they are not homoskedastic. The variance of the residuals varies in time and, hence, needs to be estimated. GARCH$(\cdot,\cdot)$ models, introduced in Bollerslev [1986], offer a suitable solution for such a problem. Therefore, we will consider an ARMA$(p_{i,j}, q_{i,j})$ - GARCH$(r_{i,j}, s_{i,j})$ model to describe $\mathbf{y}$.

We write the ARMA$(p_{i,j}, q_{i,j})$ - GARCH$(r_{i,j}, s_{i,j})$ model as:

$$\begin{aligned} y_k &= \mu_{j,k} + \varepsilon_k \\ \varepsilon_k &= \nu_k + \sum_{g=1}^{p_{i,j}} a_g \varepsilon_{k-g} + \sum_{g=1}^{q_{i,j}} b_g u_{k-g} + \sqrt{\gamma_k} u_k \\ \gamma_k &= c_0 + \sum_{g=1}^{r_{i,j}} c_g u_{k-g}^2 + \sum_{g=1}^{s_{i,j}} d_g \gamma_{k-g} \\ u_k &\overset{iid}{\sim} \mathcal{N}(0,1) \end{aligned} \tag{8.4.13}$$

were $\{a_g : g = 1, \ldots, p_{i,j}\}$ and $\{b_g : g = 1, \ldots, q_{i,j}\}$ are the ARMA$(\cdot,\cdot)$ coefficients (see Chapter 4, Section 4.7), and $\{c_g : g = 0, \ldots, r_{i,j}\}$ and $\{d_g : g = 1, \ldots, s_{i,j}\}$ are the GARCH$(\cdot,\cdot)$ coefficients. For the sake of readibility, we omitted some i, j indices in Eq. (8.4.13) and did the following notation simplifications:

$$\begin{aligned} &(y_{i,j})_k = y_k,\ (a_{i,j})_g = a_g,\ (b_{i,j})_g = b_g,\ (\varepsilon_{i,j})_k = \varepsilon_k,\ (\nu_{i,j})_k = \nu_k \\ &(u_{i,j})_k = u_k,\ (c_{i,j})_k = c_k,\ (\gamma_{i,j})_k = \gamma_k,\ (d_{i,j})_k = d_k. \end{aligned} \tag{8.4.14}$$

For the moment assume that the orders of the ARMA$(\cdot,\cdot)$ and GARCH$(\cdot,\cdot)$ models, i.e. the values of $p_{i,j}$, $q_{i,j}$, $r_{i,j}$ and $s_{i,j}$, are known, and let $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^z$ be the vector of parameters associated to model (8.4.13). Of course we may write again the following optimization problem

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}}\ \ell(\boldsymbol{\theta}|\mathbf{y})$$

but the maximization of $\ell(\boldsymbol{\theta}|\mathbf{y})$ can be very difficult in practice. Hopefully, by making again the assumption that $-\ell(\boldsymbol{\theta}|\mathbf{y})$ is convex in $\boldsymbol{\theta}$ we may use an approach called the *block coordinate descent algorithm*. This algorithm is very similar to the standard coordinate descent method except that parts of $\boldsymbol{\theta}$ (not necessarily of dimension 1) are iteratively optimized. A natural choice for the part of $\boldsymbol{\theta}$ to be optimized is every $\mu_{j,k}$, the parameters of each ARMA$(\cdot,\cdot)$ model, denoted as $\boldsymbol{\zeta}_{i,j}$, and the parameters of each GARCH$(\cdot,\cdot)$ model, denoted as $\boldsymbol{\vartheta}_{i,j}$. Therefore, we have

$$\begin{aligned} \boldsymbol{\zeta}_{i,j} &= \left[\ (a_{i,j})_1 \quad \ldots \quad (a_{i,j})_{p_{i,j}} \ \middle|\ (b_{i,j})_1 \quad \ldots \quad (b_{i,j})_{q_{i,j}}\ \right]^T \\ \boldsymbol{\vartheta}_{i,j} &= \left[\ (c_{i,j})_1 \quad \ldots \quad (c_{i,j})_{r_{i,j}} \ \middle|\ (d_{i,j})_1 \quad \ldots \quad (d_{i,j})_{s_{i,j}}\ \right]^T \end{aligned} \tag{8.4.15}$$

and the following iterative problems:

$$
\begin{aligned}
\hat{\mu}_{1,1}^{(m)} &= \underset{\mu_{1,1}\in\mathbb{R}}{\operatorname{argmax}}\ \ell\left(\mu_{1,2}|\mathbf{y}, \hat{\mu}_{1,2}^{(m-1)}, ..., \hat{\mu}_{3,N}^{(m-1)}, \boldsymbol{\zeta}_{1,1}^{(m-1)}, ..., \boldsymbol{\vartheta}_{R,3}^{(m-1)}\right) \\
&\vdots \\
\hat{\mu}_{3,N}^{(m)} &= \underset{\mu_{3,N}\in\mathbb{R}}{\operatorname{argmax}}\ \ell\left(\mu_{3,N}|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, ..., \hat{\mu}_{3,N-1}^{(m)}, \boldsymbol{\zeta}_{1,1}^{(m-1)}, ..., \boldsymbol{\vartheta}_{R,3}^{(m-1)}\right) \\
\hat{\boldsymbol{\zeta}}_{1,1}^{(m)} &= \underset{\boldsymbol{\zeta}_{1,1}\in\boldsymbol{A}}{\operatorname{argmax}}\ \ell\left(\boldsymbol{\zeta}_{i,j}|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, ..., \hat{\mu}_{3,N}^{(m)}, \boldsymbol{\zeta}_{2,1}^{(m-1)}, ..., \boldsymbol{\zeta}_{R,3}^{(m-1)}, \boldsymbol{\vartheta}_{1,1}^{(m-1)}, ..., \boldsymbol{\vartheta}_{R,3}^{(m-1)}\right) \\
&\vdots \\
\hat{\boldsymbol{\zeta}}_{R,3}^{(m)} &= \underset{\boldsymbol{\zeta}_{R,3}\in\boldsymbol{A}}{\operatorname{argmax}}\ \ell\left(\boldsymbol{\zeta}_{i,j}|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, ..., \hat{\mu}_{3,N}^{(m)}, \boldsymbol{\zeta}_{1,1}^{(m)}, ..., \boldsymbol{\zeta}_{R,2}^{(m)}, \boldsymbol{\vartheta}_{1,1}^{(m-1)}, ..., \boldsymbol{\vartheta}_{R,3}^{(m-1)}\right) \\
\hat{\boldsymbol{\vartheta}}_{1,1}^{(m)} &= \underset{\boldsymbol{\vartheta}_{1,1}\in\boldsymbol{B}}{\operatorname{argmax}}\ \ell\left(\boldsymbol{\vartheta}_{1,1}|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, ..., \hat{\mu}_{3,N}^{(m)}, \boldsymbol{\zeta}_{1,1}^{(m)}, ..., \boldsymbol{\zeta}_{R,2}^{(m)}, \boldsymbol{\vartheta}_{2,1}^{(m-1)}, ..., \boldsymbol{\vartheta}_{R,3}^{(m-1)}\right) \\
&\vdots \\
\hat{\boldsymbol{\vartheta}}_{R,3}^{(m)} &= \underset{\boldsymbol{\vartheta}_{R,3}\in\boldsymbol{B}}{\operatorname{argmax}}\ \ell\left(\boldsymbol{\vartheta}_{R,3}|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, ..., \hat{\mu}_{3,N}^{(m)}, \boldsymbol{\zeta}_{1,1}^{(m)}, ..., \boldsymbol{\zeta}_{R,2}^{(m)}, \boldsymbol{\vartheta}_{1,1}^{(m)}, ..., \boldsymbol{\vartheta}_{R,2}^{(m)}\right).
\end{aligned}
\tag{8.4.16}
$$

A closed form formula for the estimator $\hat{\mu}_{j,k}^{(m)}$ with $k = 1, \ldots, N$, for axes $j = 1, 2, 3$ and sensors $i = 1, \ldots, R$ is given by

$$
\hat{\mu}_{j,k}^{(m)} = \frac{\sum_{i=1}^{R} \frac{(y_{i,j})_k - (\hat{\eta}_{i,j})_k^{(m-1)}}{\hat{\gamma}_i^{(m-1)}}}{\sum_{i=1}^{R} \frac{1}{\hat{\gamma}_i^{(m-1)}}}
\tag{8.4.17}
$$

where

$$
\begin{aligned}
(\hat{\eta}_{i,j})_k^{(m-1)} &= \mathbb{E}\left[\varepsilon_k | \hat{\varepsilon}_{k-1}^{(m-1)}, ..., \hat{\varepsilon}_1^{(m-1)}, \hat{\boldsymbol{\zeta}}_{i,j}^{(m)}\right] \\
&= \hat{\nu}_k^{(m-1)} + \sum_{g=1}^{p_{i,j}} \hat{a}_g^{(m-1)} \varepsilon_{k-g}^{(m-1)} + \sum_{g=1}^{q_{i,j}} \hat{b}_g^{(m-1)} \hat{u}_{k-g}^{(m-1)}.
\end{aligned}
\tag{8.4.18}
$$

Again, some i, j indices were omitted in the above relation according to Eq. (8.4.14). Eq. (8.4.17) and (8.4.18) express the final form of the *ARMA-GARCH based synthetic IMU* computer in which individual sensor weights $\gamma_i^{(m)}$ are computed by the GARCH$(\cdot, \cdot)$ model. There exist no closed form expression for $\hat{\boldsymbol{\zeta}}_{i,j}^{(m)}$ and $\hat{\boldsymbol{\vartheta}}_{i,j}^{(m)}$ but it can be shown that $\hat{\boldsymbol{\zeta}}_{i,j}^{(m)}$ is equivalent to the MLE of an ARMA$(\cdot, \cdot)$ model with *iid* normally distributed residuals computed on $(y_{i,j})_k - \hat{\mu}_{j,k}^{(m)}$. It can also be shown that $\hat{\boldsymbol{\vartheta}}_{i,j}^{(m)}$ is equivalent to the MLE of a GARCH$(\cdot, \cdot)$ model with Gaussian innovations based on the observations $(y_{i,j})_k = \hat{\mu}_{j,k}^{(m)} - (\hat{\eta}_{i,j})_k^{(m)}$. For details of the MLE of ARMA$(\cdot, \cdot)$ and GARCH$(\cdot, \cdot)$ models, we refer to Hamilton [1994].

A natural choice for the initial values of $\hat{\boldsymbol{\theta}}^{(0)}$ is $\hat{\mu}_{j,k}^{(0)} = \hat{\mu}_{j,k}$ (i.e. the unweighted synthetic IMU defined in Eq. (8.4.7)), $\hat{\nu}_{i,j}^{(0)} = 0$, $(\hat{a}_{i,j})_g^{(0)} = 0$, $(\hat{b}_{i,j})_g^{(0)} = 0$, $(\hat{c}_{i,j})_0^{(0)} = \hat{\gamma}_i^{(\infty)}$ (where $\hat{\gamma}_i^{(\infty)}$ denotes the values of $\hat{\gamma}_i$ at convergence of the algorithm defined in Eq. (8.4.12)), $(\hat{c}_{i,j})_g^{(0)} = 0$ and

$(\hat{d}_{i,j})_g^{(0)} = 0$. Similarly to the previous section, the iterative approach defined in Eq. (8.4.16) should be repeated until convergence of $\hat{\boldsymbol{\theta}}^{(m)}$ is achieved.

In practice the order values of the ARMA$(\cdot,\cdot)$ and GARCH$(\cdots,\cdot)$ models, i.e. the $p_{i,j}$, $q_{i,j}$, $r_{i,j}$ and $s_{i,j}$ values, are unknown. We propose here an *ad-hoc* approach for finding reasonable values for these coefficients, although it is not a real-time procedure.

Algorithm 8.4.1. This procedure enables to find values for $p_{i,j}$, $q_{i,j}$, $r_{i,j}$ and $s_{i,j}$:

1. Compute a reasonable estimates of $\mu_{j,k}$. For example, we may use the unweighted synthetic sensor $\hat{\mu}_{j,k}$ defined in Eq. (8.4.7) or $\hat{\mu}_{j,k}^{(\infty)}$ defined as the values of $\hat{\mu}_{j,k}$ at convergence of Eq. (8.4.12).

2. Estimate for each sensor $i = 1, \ldots, R$ and each axis $j = 1, 2, 3$ all possible ARMA$(\cdot,\cdot)$ models, by assuming *iid* normally distributed residuals. For that, use

$$(y_{i,j})_k - \hat{\mu}_{j,k} \text{ or } (y_{i,j})_k - \hat{\mu}_{j,k}^{(\infty)} \tag{8.4.19}$$

   satisfying $0 \leq p_{i,j} \leq p_{max}$ and $0 \leq q_{i,j} \leq q_{max}$.

3. Select the values of $p_{i,j}$ and $q_{i,j}$ such that they minimize a prediction criterion such as the AIC (proposed in Akaike [1974]) or the BIC (proposed in Schwarz [1978]), i.e.

$$[\hat{p}_{i,j}\ \hat{q}_{i,j}] = \underset{\substack{0\leq p_{i,j}\leq p_{max}\\ 0\leq q_{i,j}\leq q_{max}}}{\operatorname{argmin}} f(N)(p_{i,j} + q_{i,j} + 1) - 2\ell^{\star}_{p_{i,j},q_{i,j}} \tag{8.4.20}$$

   where $\ell^{\star}_{p_{i,j},q_{i,j}}$ denotes the log-likelihood associated to the ARMA$(p_{i,j}, q_{i,j})$ model based on the sequences of Eq. (8.4.19) and $f(N)$ is a penalty function. By taking $f(N) = 2$, the vector $[\hat{p}_{i,j}\ \hat{q}_{i,j}]$ defined in Eq. (8.4.20) corresponds to the values of $p_{i,j}$ and $q_{i,j}$ that minimize the prediction error measured by the AIC, while using $f(N) = \log(N)$ corresponds to the BIC.

4. Compute the residuals of the ARMA$(\cdot,\cdot)$ models for $i = 1, \ldots, R$ and $j = 1, 2, 3$ with orders defined by Eq. (8.4.20). Estimate for each set of residuals all possible GARCH$(\cdot,\cdot)$ models, assuming Gaussian innovations, satisfying $0 \leq r_{i,j} \leq r_{max}$ and $0 \leq s_{i,j} \leq s_{max}$, and select the values of $r_{i,j}$ and of $s_{i,j}$ as the results of

$$[\hat{r}_{i,j}\ \hat{s}_{i,j}] = \underset{\substack{0\leq r_{i,j}\leq r_{max}\\ 0\leq s_{i,j}\leq s_{max}}}{\operatorname{argmin}} f(N)(r_{i,j} + s_{i,j} + 1) - 2\ell^{\circ}_{r_{i,j},s_{i,j}} \tag{8.4.21}$$

   where $\ell^{\circ}_{r_{i,j},s_{i,j}}$ denotes the log-likelihood associated to the GARCH$(r_{i,j}, s_{i,j})$ based on residuals of the ARMA$(\cdot,\cdot)$ models selected by Eq. (8.4.20) and $f(N)$ is the same penalty function that was used in Eq. (8.4.20).

Choosing between the AIC and BIC, or another criterion with similar properties, is a non-trivial task. For example if we assume that the true model is of *finite dimension* and is included among the candidate models, then the BIC should be preferred since it is a consistent criterion (i.e. it will select the true model with a probability approaching 1 as $N \to \infty$). However, if we assume that the true model is of *infinite dimension* or that this model is not included in the set of candidate models, then the AIC should be preferred. Indeed, the AIC is an *efficient* criterion, meaning that it will select the "closest" model to the true underlying (generating) model.

8.4.3 Markovian Regime-Switching based Synthetic IMU Computer

While in Section 8.4.2 as well as in Stebler et al. [2011b], Waegli et al. [2010] GARCH$(\cdot,\cdot)$ models were introduced as a direct noise estimation technique for synthetic IMUs able to estimate changes of sensor variances, a single ARMA$(\cdot,\cdot)$ model was used for modeling the autocorrelated signal part. We can now go further by treating the case where the structure of the inertial sensor error correlation varies in time. The right panel of Figure 8.2 depicts the estimated ACS of two portions of the accelerometer error signal drawn in the left panel of Figure 8.1. The two portions correspond to time intervals during which the vehicule was not moving (gray curve in ACS plot) and moving (black curve in ACS plot). From the two curves, different time correlations may be suspected. Accounting for such effects could be performed by using two sets of coefficients for modeling the autocorrelated part. This section deals with such issues using *Markov Regime-Switching models* (MRS). We introduce them as an alternative to the ARMA$(\cdot,\cdot)$-GARCH$(\cdot,\cdot)$ modeling procedure, as an attempt to account not only for time-varying noise levels, but also for varying time correlations in the noise signal.

The problem of modeling changes in *regimes* (sometimes also called *states*) in time series has been of interest in many different fields. It has, for example, been used in medicine to model influenza epidemics Martínez-Beneito et al. [2008] or in economics with U.S. gross national product Hamilton [1989], McCulloch and Tsay [1993]. In the context of state space models, Shumway and Stoffer [1991] applied successfully this approach for the tracking of large numbers of moving targets. Roughly speaking, regime-switching models can be distinguished between two principle classes:

1. Models which assume that the regimes can be characterized by an *observed* variable. Therefore, the regimes are purely deterministic, given a realization of the observed variable on which the regime depends. *Threshold Auto-Regressive models* (TAR) proposed by Tong and Lim [1980] are examples of such models.

2. Models which assume that the regimes are *not observed*, but can be determined by an underlying unobserved stochastic process. This implies that the regimes are of stochastic nature. *Markov-Switching models* advocated by Hamilton [1989] are an example of models among which some are governed by a first-order Markov process. Detailed reviews of these models can, for example, be found in Franses and Van Dijk [2000], Hamilton [1994], Shumway and Stoffer [2000]

Formally, we will consider the following MRS model for digital times $\{t_k : k = 1, \ldots, N\}$:

$$
\begin{aligned}
y_k &= \mu_{j,k} + \varepsilon_k \\
\varepsilon_k &= \sum_{s=1}^{S^\star_{i,j}} \left[I_{(S_k=s)} \left(\nu_{s,k} + \sum_{g=1}^{d_s} a_{g,s}\, \varepsilon_{k-g} + \sigma_s\, u_k \right) \right] \\
I_{(S_k=s)} &= \begin{cases} 1 & \text{if } S_k = s \\ 0 & \text{if } S_k \neq s \end{cases} \\
u_k &\overset{iid}{\sim} N(0,1)
\end{aligned}
\tag{8.4.22}
$$

where $S_{i,j}$ is a random variable denoting the current regime such that $S_{i,j} \in \left\{1, ..., S^{\star}_{i,j}\right\}$ with $S^{\star}_{i,j}$ denoting the number of regimes present in sensor i on axis j. Note that some i, j indices were omitted for readibility reasons and therefore,

$$\begin{array}{l}(y_{i,j})_k = y_k,\ (a_{i,j})_{g,s} = a_{g,s},\ (\varepsilon_{i,j})_k = \varepsilon_k,\ (\nu_{i,j})_{s,k} = \nu_{s,k},\ (d_{i,j})_s = d_s \\ (\sigma_{i,j})_s = \sigma_s,\ (u_{i,j})_k = u_k,\ (S_{i,j})_k = S_k.\end{array} \tag{8.4.23}$$

have to be considered in Eq. (8.4.22). The second equation in Model (8.4.22) models the residuals ε_k as an AR(d_s) process (see Chapter 4, Section 4.7) whose order $\{d_s : s = 1, \ldots, S^{\star}_{i,j}\}$ and coefficients $\{a_{g,s} : g = 1, \ldots, d_s\ ; s = 1, \ldots, S^{\star}_{i,j}\}$ change according to regime S_k. To define completely model (8.4.22), the properties of the regime's process S_k need to be specified. We will assume that S_k is a first-order Gauss-Markov process as advocated in Hamilton [1989]. This implies that the current regime (i.e. S_k) only depends on the regime one period ago (i.e. S_{k-1}). Therefore, such a model is completely defined with $3R$ transition matrices $\mathbf{P}_{i,j}$ which contain the probabilities of moving from one regime to the other for each sensor i and each axis j, i.e.

$$\mathbf{P}_{i,j} = \left[p^{i,j}_{l,h}\right]_{\substack{l=1,\ldots,S^{\star}_{i,j} \\ h=1,\ldots,S^{\star}_{i,j}}}$$

where

$$p^{i,j}_{l,h} = P\left(S_k = l | (S_{k-1} = h\right). \tag{8.4.24}$$

Now, we let $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^v$ be the vector of parameters associated to model (8.4.22). Then, we aim to maximize the likelihood function $\ell\left(\boldsymbol{\theta}|\mathbf{y}\right)$ to obtain the point estimate $\hat{\boldsymbol{\theta}}$. Of course, this problem is quite complex. Therefore, we will consider for the moment a simpler problem. Assume that the values of $\mu_{j,k}$ are known. Then we could directly observe ε_k which considerably simplifies our problem. Assuming further that the number of regimes $S^{\star}_{i,j}$ and the AR($\cdot$) order d_s are either known or fixed (i.e. not estimated from the data), we may use the method proposed by Hamilton [1989] to estimate the remaining unknown parameters of Eq. (8.4.22). More precisely, these unknown parameters are

$$\boldsymbol{\theta}^{\star} = \left[\begin{array}{cccc|ccc|ccc} (\nu_{1,1})_1 & \ldots & (\nu_{N,3})_{S^{\star}_{N,3}} & | & (a_{1,1})_{1,1} & \ldots & (a_{(d_{N,3})_{S^{\star}_{N,3}},N})_{3,S^{\star}_{N,3}} & | & (\sigma_{1,1})_1 & \ldots & (\sigma_{N,3})_{S^{\star}_{N,3}} \end{array}\right]^T \tag{8.4.25}$$

and its estimation will be explained in the sequel.

Since we do not observe the regimes directly, the inference on the current state at time t_k for the sensor i on the axis j is performed through the observed behaviour of y_k. Indeed, the estimated probability of being in a given regime can be obtained as

$$\xi^{i,j}_{s|k} = P\left(S_k = s | \boldsymbol{\Omega}_k, \boldsymbol{\theta}^{\star}, \boldsymbol{\mu}\right) \tag{8.4.26}$$

where $\boldsymbol{\Omega}_k$ denotes the information set at digital times $\{t_k : k = 1, \ldots, N\}$ and

$$\boldsymbol{\mu} = \left[\begin{array}{ccc} \mu_{1,k} & \mu_{2,k} & \mu_{3,k} \end{array}\right]^T.$$

If the system is in the regime s, then its density at time t_k is given by

$$\begin{aligned}\eta_{s|k}^{i,j} &= f\left(y_k|S_k = s, \boldsymbol{\Omega}_{k-1}, \boldsymbol{\theta}^\star, \boldsymbol{\mu}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_s} \exp\left[\frac{-\left(y_k - \mu_{j,k} - \nu_s - \sum_{g=1}^{d_s} a_{g,s}\ y_{k-g}\right)^2}{2\sigma_s^2}\right]\end{aligned} \tag{8.4.27}$$

where notation simplifications have been done according to Eq. (8.4.23), and the following additional simplifications

$$p_{l,h}^{i,j} = p_{l,h},\ \ \xi_{s|k-1}^{i,j} = \xi_{s|k-1}, \quad \text{and} \quad \eta_{s|k}^{i,j} = \eta_{s|k}. \tag{8.4.28}$$

are applied in the sequel. Despite the fact that the true current system regime is unknown, using the Bayes theorem enables to write the density of y_k as

$$f(y_k|\boldsymbol{\Omega}_{k-1}, \boldsymbol{\theta}^\star, \boldsymbol{\mu}) = \sum_{l=1}^{S_{i,j}^\star} \sum_{h=1}^{S_{i,j}^\star} p_{l,h} \cdot \xi_{l|k-1} \cdot \eta_{h|k}. \tag{8.4.29}$$

The result of Eq. (8.4.29) allows to compute the updated regime probability, i.e.

$$\xi_{s|k} = \frac{\sum_{l=1}^{S_{i,j}^\star} p_{l,h} \cdot \xi_{l|k-1} \cdot \eta_{h|k}}{f(y_k|\boldsymbol{\Omega}_{k-1}, \boldsymbol{\theta}^\star)}. \tag{8.4.30}$$

The log-likelihood of Model (8.4.22) is defined as

$$\ell\left(\boldsymbol{\theta}^\star|\mathbf{y}, \boldsymbol{\mu}\right) = \sum_{i=1}^{R} \sum_{j=1}^{3} \left(\log f\left(y_1|\boldsymbol{\theta}^\star, \boldsymbol{\mu}\right) + \sum_{k=2}^{N} \log f\left(y_k|\boldsymbol{\Omega}_{k-1}, \boldsymbol{\theta}^\star, \boldsymbol{\mu}\right)\right). \tag{8.4.31}$$

By executing Eq. (8.4.27), (8.4.29) and (8.4.30), we can evaluate the approximate log-likelihood of the observed data as

$$\ell^\star\left(\boldsymbol{\theta}^\star|\mathbf{y}^\star, \boldsymbol{\mu}\right) = \sum_{i=1}^{R} \sum_{j=1}^{3} \sum_{k=2}^{N} \log f\left(y_k|\boldsymbol{\Omega}_{k-1}, \boldsymbol{\theta}^\star, \boldsymbol{\mu}\right) \tag{8.4.32}$$

where $\mathbf{y}^\star = \{(y_{i,j})_k : k = 2, \ldots, N;\ i = 1, \ldots, R;\ j = 1, 2, 3\}$ by arguing that $\ell\left(\boldsymbol{\theta}^\star|\mathbf{y}, \boldsymbol{\mu}\right)$ defined in Eq. (8.4.31) and $\ell^\star\left(\boldsymbol{\theta}^\star|\mathbf{y}^\star, \boldsymbol{\mu}\right)$ defined in Eq. (8.4.32) are extremely close in large samples. Beside the computation of $\ell^\star\left(\boldsymbol{\theta}^\star|\mathbf{y}^\star, \boldsymbol{\mu}\right)$, the initial $\xi_{s|0}$ are required for all possible s, i and j. Several options are available for this starting value. Assuming the Markov chain to be ergodic, the unconditional probability can be used to find $\xi_{s|0}$ (see e.g. Hamilton [1994] for more details). Alternatively, one could simply set $\xi_{s|0} = 1/S_{i,j}^\star$ for all s, i and j. Therefore, an estimate of $\boldsymbol{\theta}$ can be obtained numerically by solving the following problem:

$$\hat{\boldsymbol{\theta}}^\star = \underset{\boldsymbol{\theta}^\star \in \boldsymbol{\Theta}^\star}{\operatorname{argmax}}\ \ell^\star\left(\boldsymbol{\theta}^\star|\mathbf{y}^\star, \boldsymbol{\mu}\right). \tag{8.4.33}$$

Alternatively, Hamilton [1989] proposed an iterative procedure which can be used to find $\hat{\boldsymbol{\theta}}^\star$. Such approach is in general more stable and less computationally intensive than solving Eq. (8.4.33). It turns out that this procedure is an application of the EM algorithm developed by

Dempster et al. [1977]. The idea of Hamilton's approach is to express some portions of $\boldsymbol{\theta}^\star$ as functions of $\hat{\boldsymbol{\theta}}^\star$ and then, given starting values for the parameters, to compute iteratively each portion of $\boldsymbol{\theta}^\star$ until convergence occurs. Note that once $\hat{\boldsymbol{\theta}}^\star$ has been computed, we can compute forecasted and smoothed inferences for the regimes. This allows to compute (smoothed) probabilities for the sensor i on the axis j at digital time t_k to be in the regime s. An efficient algorithm to compute these probabilities has been proposed in Kim [1994] and complete discussion on the subject can be found in Hamilton [1994].

We may now relax the assumption that $\boldsymbol{\mu}$ is known. Similarily to what was done in the previous section, the block coordinate descent algorithm can be used to solve

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}}\ \ell^\circ\left(\boldsymbol{\theta}|\mathbf{y}\right) \tag{8.4.34}$$

where ℓ° denotes the likelihood of Model (8.4.22). A natural choice for the part of $\boldsymbol{\theta}$ to be optimized is each $\mu_{j,k}$ and $\boldsymbol{\theta}^\star$. Therefore, we have the following iterative problems:

$$\begin{aligned}
\hat{\mu}_{1,1}^{(m)} &= \underset{\mu_{1,1}\in\mathbb{R}}{\operatorname{argmax}}\ \ell\left(\mu_{1,1}|\mathbf{y}, \hat{\mu}_{1,2}^{(m-1)}, \ldots, \hat{\mu}_{3,N}^{(m-1)}, (\hat{\boldsymbol{\theta}}^\star)^{(m-1)}\right)\\
&\vdots\\
\hat{\mu}_{3,N}^{(m)} &= \underset{\mu_{3,N}\in\mathbb{R}}{\operatorname{argmax}}\ \ell\left(\mu_{3,N}|\mathbf{y}, \hat{\mu}_{1,1}^{(m)}, \ldots, \hat{\mu}_{3,N-1}^{(m)}, (\hat{\boldsymbol{\theta}}^\star)^{(m-1)}\right)\\
(\hat{\boldsymbol{\theta}}^\star)^{(m)} &= \underset{\boldsymbol{\theta}^\star\in\boldsymbol{\Theta}^\star}{\operatorname{argmax}}\ \ell^\star\left(\boldsymbol{\theta}^\star|\mathbf{y}^\star, \hat{\mu}_{1,1}^{(m)}, \ldots, \hat{\mu}_{3,N}^{(m)}\right)
\end{aligned} \tag{8.4.35}$$

were $m = 0, 1, 2, \ldots$ is the mth iteration. Fortunately, $\hat{\mu}_{j,k}^{(m)}$ can be obtained analytically as

$$\hat{\mu}_{j,k}^{(m)} = \frac{\sum_{i=1}^{R} \frac{(y_{i,j})_k - (\kappa_{i,j})_k^{(m-1)}}{\gamma_i^{(m-1)}}}{\sum_{i=1}^{R} \frac{1}{\gamma_i^{(m-1)}}} \tag{8.4.36}$$

where

$$\begin{aligned}
(\kappa_{i,j})_k^{(m-1)} &= \mathbb{E}\left[\varepsilon_k|\varepsilon_{k-1}, \ldots, \varepsilon_1, (\hat{\boldsymbol{\theta}}^\star)^{(m-1)}\right]\\
(\gamma_{i,j})_k^{(m-1)} &= \operatorname{var}\left[\varepsilon_k|\varepsilon_{k-1}, \ldots, \varepsilon_1, (\hat{\boldsymbol{\theta}}^\star)^{(m-1)}\right].
\end{aligned} \tag{8.4.37}$$

These quantities can be estimated as

$$\begin{aligned}
(\hat{\kappa}_{i,j})_k^{(m-1)} &= \sum_{s=1}^{S_{i,j}^\star}\left[\hat{\xi}_{s|k}^{(m-1)}\left(\hat{\nu}_{s,k}^{(m-1)} + \sum_{g=1}^{d_s}\hat{a}_{g,s}^{(m-1)}\ \left(y_k - \hat{\mu}_{j,k}^{(m-1)}\right)\right)\right]\\
(\hat{\gamma}_{i,j})_k^{(m-1)} &= \sum_{s=1}^{S_{i,j}^\star}\hat{\xi}_{s|k}^{(m-1)}\ (\hat{\sigma}_k^2)^{(m-1)}.
\end{aligned} \tag{8.4.38}$$

By using in Eq. (8.4.36) the estimates for $(\kappa_{i,j})_k^{(m-1)}$ and $(\gamma_{i,j})_k^{(m-1)}$ defined in Eq. (8.4.38), one can compute $\hat{\mu}_{j,k}^{(m)}$ and solve (8.4.35) iteratively until convergence. For this model the choice

of $\hat{\boldsymbol{\theta}}^{(0)}$ might be slightly more difficult than for the previous models. We recommend to use $\hat{\mu}_{j,k}^{(0)} = \hat{\mu}_{j,k}$ (where $\hat{\mu}_{j,k}$ is defined in Eq. (8.4.7)) and use any plausible starting values for $\boldsymbol{\theta}^{\star\,(0)}$. Finally, in practice the values of $S_{i,j}^{\star}$ and d_s are unknown and should somehow be determined for the data at hand. We propose here to use the same kind of procedure as proposed in Algorithm 8.4.1 (see Section 8.4.2).

Algorithm 8.4.2. This procedure enables to find values for $S_{i,j}^{\star}$ and d_s:

1. Compute a reasonable estimates of $\mu_{j,k}$. For example, use $\hat{\mu}_{j,k}$ defined in Eq. (8.4.7) or $\hat{\mu}_{j,k}^{(\infty)}$ defined as the values of $\hat{\mu}_{j,k}$ at convergence of Eq. (8.4.12).
2. Estimate for each values of $i = 1, \ldots, R$ and $j = 1, 2, 3$ all possible MRS AR$(\cdot)$ models based on $(y_{i,j})_k - \hat{\mu}_{j,k}$ (or on $(y_{i,j})_k - \hat{\mu}_{j,k}^{(\infty)}$) satisfying $0 \leq d_s \leq d_{max}$ and $1 \leq S_{i,j}^{\star} \leq s_{max}^{\star}$.
3. Select values for d_s and $S_{i,j}^{\star}$ such that they minimize

$$\left[\hat{d}_s \;\hat{S}_{i,j}^{\star}\right] = \underset{\substack{0 \leq d_s \leq d_{max} \\ 1 \leq S_{i,j}^{\star} \leq s_{max}^{\star}}}{\operatorname{argmin}} \; f(N)\,(d_s + 2)\,S_{i,j}^{\star} - 2\ell_{d_s,S_{i,j}^{\star}}^{\star} \tag{8.4.39}$$

   where $\ell_{d_s,S_{i,j}^{\star}}^{\star}$ denotes the log-likelihood as defined in Eq. (8.4.32) associated to an MRS AR$(\cdot)$ process with values d_s and $S_{i,j}^{\star}$ and assuming the values of $\boldsymbol{\mu}$ to be known. For the penalty function $f(N)$ we recommend using either $f(N) = 2$ or $f(N) = \log(N)$ which respectively correspond to the AIC and the BIC. The resulting values for $S_{i,j}^{\star}$ (and d_s) should however be considered with caution has it was reported in Smith et al. [2006] that the AIC tends to retain too many states and variables while the BIC underestimates the correct number of states. Therefore, the selection performance of $S_{i,j}^{\star}$ and d_s might be improved by using the *Markov Switching Criterion* (MSC) in which $f(N)\,(d_s + 2)\,S_{i,j}^{\star}$ from Eq. (8.4.39) is replaced by a more suitable quantity (see Smith et al. [2006] for details).

8.4.4 Extended Kalman Filtering using a Synthetic IMU

Several remarks can be made when using a synthetic IMU in a navigation filter:

1. Theoreticaly, the residuals $\{\varepsilon_k : k = 1, \ldots, N\}$ of models (8.4.13) and (8.4.22) are supposed to be *i.i.d.* such that $u_k \sim \mathcal{N}(0, 1)$ if the true model parameters are known. This would imply that the resulting synthetic IMU signals are perfectly whitened when introduced into the EKF, and thus no additional error states are required in the filter. In practice however, the true model (ARMA-GARCH and MRS) parameters are unknown and hence must be estimated. The resulting estimation error implies that the estimated residuals $\hat{\epsilon}_k$ are not *i.i.d.*, meaning that some correlation may still be present. Moreover, if the constant ν_k is not estimated, the estimated residuals may also contain a bias. These two issues can be handled in the EKF by augmenting the state vector with additional error states as a random bias or a first-order Gauss-Markov process for example. Note that if the parameters of the ARMA$(\cdot,\cdot)$-GARCH$(\cdot,\cdot)$ and the MRS are unknown (and hence have to be estimated), the computation of the synthetic IMU is only feasible in post-processing procedures.

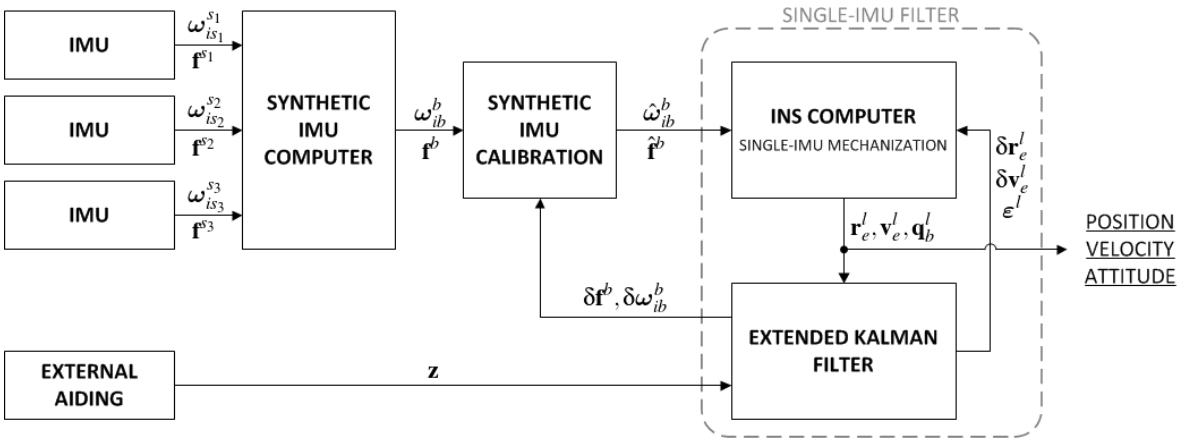

Figure 8.3: Principle of mechanization based on a synthetic IMU.

Table 8.1: Summary of the main differences between the ARMA-GARCH based and the MRS based synthetic IMU computer.

	ARMA-GARCH	**MRS**
Effort to automate parameter estimation and model selection	−	−
Computational effort	−	−
The volatility of each sensor is modeled	+	+
Predictable part of the errors can be linked to dynamical condition	−	+
Residual level can be linked to dynamical condition	+	+
Quality of the variance modeling	+	−
Quality of the auto-regressive noise component modeling	−	+

2. Synthetic IMU computers deliver time-variant gyroscope and accelerometer noise levels which can be used in the $\mathbf{Q}_k$ system noise matrix of the EKF, thus making the stochastic model in the navigation filter adapt to reality.

The principle of the integrated navigation in which a synthetic IMU computer is used is schematized in Figure 8.3. Note that individual accelerometer and gyroscope biases cannot be estimated and fed back to the corresponding sensors. Table 8.1 summarizes the main differences between the ARMA-GARCH and MRS synthetic IMU computer and can be seen as an extension of the study proposed in Stebler et al. [2011b], Waegli et al. [2010].

The benefits of the ARMA-GARCH method might be larger when the noise level of the inertial sensors varies significantly in time and among the sensors. On the other side, the benefits of the MRS method might be larger if the noise correlation structure is suspected to change over time. If none of these features are fullfilled, the unweighted synthetic IMU may be sufficient.

8.5 Redundancy in State Space

As mentioned in Section 8.2, redundancy can be handled also at the state level. In this section, we first describe how to implement a generalized filter based on extended mechanization

equations. In the context of integrated inertial navigation, we will call it the *generalized EKF*. Then we develop another approach for multiple IMU integration in which the navigation solutions of individual IMUs are estimated separately and the results are compared at regular time intervals. Such a filter will be referred to as the *geometrically-constrained EKF*.

8.5.1 Extended Mechanization

System error dynamics

The alignment of the redundant signals in the b-frame can be performed directly at mechanization level. Consider again the $(3R \times 1)$ vectors $\boldsymbol{a}$ and $\boldsymbol{g}$ of Eq. (8.4.1) which contain the rotated specific force and angular rate signals of each conventional IMU. Their corresponding errors are written as

$$\delta\mathbf{x}_a = \begin{bmatrix} \delta\mathbf{f}^{s_1} \\ \vdots \\ \delta\mathbf{f}^{s_R} \end{bmatrix} \quad \text{and} \quad \delta\mathbf{x}_g = \begin{bmatrix} \delta\boldsymbol{\omega}_{is_1}^{s_1} \\ \vdots \\ \delta\boldsymbol{\omega}_{is_R}^{s_R} \end{bmatrix}. \tag{8.5.1}$$

When accounting for R IMUs, the attitude error model which was defined for a single IMU mechanization in Eq. (1.7.14) becomes

$$\dot{\boldsymbol{\varepsilon}}^l = \mathbf{C}_b^l \boldsymbol{\Pi}_g \delta\mathbf{x}_g - \delta\boldsymbol{\omega}_{il}^l - \boldsymbol{\omega}_{il}^l \times \boldsymbol{\varepsilon}^l \tag{8.5.2}$$

where $\boldsymbol{\Pi}_g$ was defined in Eq. (8.4.3). Note that $\boldsymbol{\omega}_{lb}^b$ expressed in Eq. (1.5.16) is modified as

$$\boldsymbol{\omega}_{lb}^b = \boldsymbol{\Pi}_g \boldsymbol{g} - \mathbf{C}_l^b \boldsymbol{\omega}_{il}^l. \tag{8.5.3}$$

The modified velocity equation given in Eq. (1.5.9) becomes

$$\dot{\mathbf{v}}_e^l = \mathbf{C}_b^l \boldsymbol{\Pi}_a \boldsymbol{a} + \mathbf{g}^l - \left(\boldsymbol{\omega}_{el}^l + 2\boldsymbol{\omega}_{ie}^l\right) \times \mathbf{v}_e^l \tag{8.5.4}$$

from which the velocity error model can be deduced as

$$\delta\dot{\mathbf{v}}_e^l = -\left(\mathbf{C}_b^l \boldsymbol{\Pi}_a \boldsymbol{a}\right) \times \boldsymbol{\varepsilon}^l - \left(\boldsymbol{\omega}_{ie}^l + \boldsymbol{\omega}_{il}^l\right) \times \delta\mathbf{v}_e^l - \left(\delta\boldsymbol{\omega}_{ie}^l + \delta\boldsymbol{\omega}_{il}^l\right) \times \mathbf{v}_e^l + \mathbf{C}_b^l \boldsymbol{\Pi}_a \delta\mathbf{x}_a + \delta\mathbf{g}^l. \tag{8.5.5}$$

were $\boldsymbol{\Pi}_a$ was defined in Eq. (8.4.3). The position error model defined in Eq. (1.7.24) remains unchanged. Therefore, the final redundant INS error model can be written as

$$\begin{bmatrix} \delta\dot{\mathbf{r}}_e^l \\ \delta\dot{\mathbf{v}}_e^l \\ \dot{\boldsymbol{\varepsilon}}^l \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-1}\delta\dot{\mathbf{v}}_e^l - \mathbf{D}^{-1}\mathbf{D}_r\delta\mathbf{r}_e^l \\ -\left(\mathbf{C}_b^l \boldsymbol{\Pi}_a \boldsymbol{a}\right) \times \boldsymbol{\varepsilon}^l - \left(\boldsymbol{\omega}_{ie}^l + \boldsymbol{\omega}_{il}^l\right) \times \delta\mathbf{v}_e^l - \left(\delta\boldsymbol{\omega}_{ie}^l + \delta\boldsymbol{\omega}_{il}^l\right) \times \mathbf{v}_e^l + \mathbf{C}_b^l \boldsymbol{\Pi}_a \delta\mathbf{x}_a + \delta\mathbf{g}^l \\ \mathbf{C}_b^l \boldsymbol{\Pi}_g \delta\mathbf{x}_g - \delta\boldsymbol{\omega}_{il}^l - \boldsymbol{\omega}_{il}^l \times \boldsymbol{\varepsilon}^l \end{bmatrix} \tag{8.5.6}$$

which can be further expanded in state space notation:

$$\delta\dot{\mathbf{x}}^l = \mathbf{F}\delta\mathbf{x}^l + \mathbf{G}\mathbf{q} \tag{8.5.7}$$

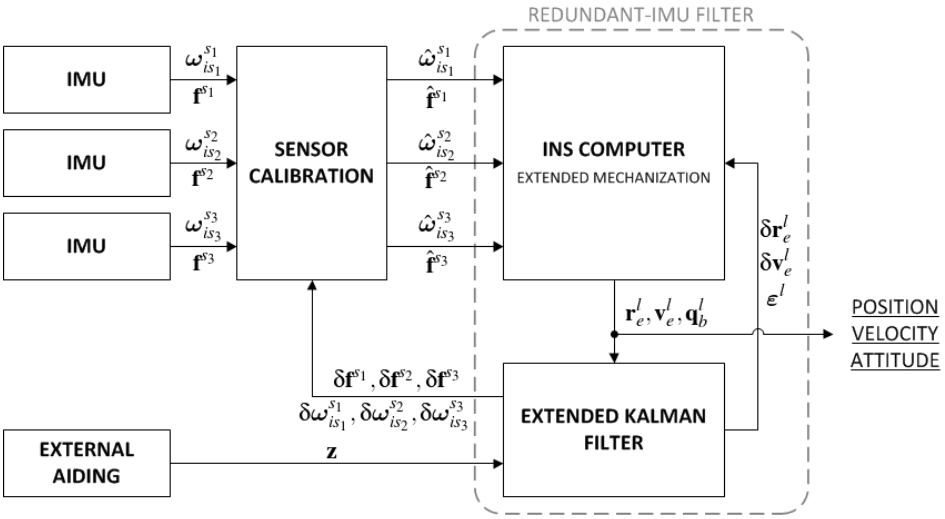

Figure 8.4: Principle of the generalized extended Kalman filter based on the extended mechanization.

where $\delta\dot{\mathbf{x}}^l$ represents the error in the nominal navigation state. Thus, the state augmented version is

$$\begin{bmatrix} \delta\dot{\mathbf{r}}_e^l \\ \delta\dot{\mathbf{v}}_e^l \\ \dot{\boldsymbol{\varepsilon}}^l \\ \delta\dot{\mathbf{x}}_a \\ \delta\dot{\mathbf{x}}_g \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{rr} & \mathbf{F}_{rv} & \mathbf{F}_{r\varepsilon} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{vr} & \mathbf{F}_{vv} & \mathbf{F}_{v\varepsilon} & \mathbf{C}_b^l\boldsymbol{\Pi}_a & \mathbf{0} \\ \mathbf{F}_{\varepsilon r} & \mathbf{F}_{\varepsilon v} & \mathbf{F}_{\varepsilon\varepsilon} & \mathbf{0} & \mathbf{C}_b^l\boldsymbol{\Pi}_g \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{aa} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{gg} \end{bmatrix} \begin{bmatrix} \delta\mathbf{r}_e^l \\ \delta\mathbf{v}_e^l \\ \boldsymbol{\varepsilon}^l \\ \delta\mathbf{x}_a \\ \delta\mathbf{x}_g \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\Pi}_a\mathbf{C}_b^l & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Pi}_g\mathbf{C}_b^l & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_g \end{bmatrix} \begin{bmatrix} -\boldsymbol{\nu}_a \\ \boldsymbol{\nu}_g \\ \boldsymbol{w}_a \\ \boldsymbol{w}_g \end{bmatrix}. \tag{8.5.8}$$

where the other elements of the model were defined in Chapter 1 (see Section 1.7.4). From Eq. (8.5.8) it can be seen that biases of each individual IMU can be modeled in the augmented part of the state.

Generalized Extended Kalman Filter

The filter prediction stage developed in Section 2.3 has to be modified as follows. The classical strapdown mechanization algorithm remains identical with the exception of Eq. (2.3.6) and (2.3.10) which become

$$\mathbf{u}_k = \int_{t_{k-1}}^{t_k} \dot{\mathbf{u}}(t)\, dt \approx \boldsymbol{\Pi}_g \cdot \boldsymbol{g}_k \cdot \Delta t_k \tag{8.5.9}$$

and

$$(\Delta\mathbf{v}_{\mathbf{f}}^l)_k = \int_{t_{k-1}}^{t_k} \mathbf{C}_b^l\boldsymbol{\Pi}_a\boldsymbol{a}\, dt \approx (\mathbf{C}_b^l)_k \cdot \boldsymbol{\Pi}_a \cdot \boldsymbol{a}_k \cdot \Delta t, \tag{8.5.10}$$

respectively, with $\Delta t_k = t_k - t_{k-1}$, $(\mathbf{C}_b^l)_k = (\mathbf{C}_b^l)_{k|k-1}$ and digital times $\{t_k : k = 1, \ldots, N\}$. The field computations in the prediction-correction integration algorithm of Section 2.3.2 must account for redundancy thanks to the following modifications. The digital attitude field of Eq. (2.3.20) remains the same except that

$$(\boldsymbol{\omega}_{lb}^b)_k = \boldsymbol{\Pi}_g \cdot \boldsymbol{g}_k - (\mathbf{C}_l^b)_k (\boldsymbol{\omega}_{il}^l)_k \tag{8.5.11}$$

has to be considered. The digital velocity field defined in Eq. (2.3.25) is transformed to

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, t_k) = (\mathbf{C}_b^l)_k \cdot \boldsymbol{\Pi}_a \cdot \boldsymbol{a}_k - \left[2(\boldsymbol{\omega}_{ie}^l)_k + (\boldsymbol{\omega}_{el}^l)_k\right] \times (\mathbf{v}_e^l)_k + (\mathbf{g}^l)_k \tag{8.5.12}$$

while the position field of Eq. (2.3.28) remains unchanged.

In the filter update stage described in Section 2.4, the linearized absolute position update model of Eq. (2.4.3) accounting for redundancy is given by

$$\mathbf{H}_\mathbf{r} = \left[\ \mathbf{I}_{3\times3} + \mathbf{D}_a \ \middle|\ \mathbf{0}_{3\times3} \ \middle|\ -\mathbf{D}^{-1}[\mathbf{a}^l\times] \ \middle|\ \mathbf{0}_{3\times3R}\ \right] \tag{8.5.13}$$

while the linearized absolute velocity update model given in Eq. (2.4.8) becomes

$$\mathbf{H}_\mathbf{v} = \left[\ [\mathbf{a}^l\times]\mathbf{D}_1 \ \middle|\ \mathbf{I}_{3\times3} + [\mathbf{a}^l\times]\mathbf{D}_2 \ \middle|\ [\mathbf{a}^l\times]\left(\boldsymbol{\Omega}_{il}^l + 2\boldsymbol{\Omega}_{lb}^b\right) \ \middle|\ \mathbf{0}_{3\times3R} \ \middle|\ -[\mathbf{a}^l\times]\mathbf{C}_b^l\boldsymbol{\Pi}_a\ \right]. \tag{8.5.14}$$

The principle of the extended navigation is schematized in Figure 8.4. Note that as already mentioned, the sensor biases can be estimated for each individual gyroscopes and accelerometer. Hence, in contrary to the synthetic IMU approach, the generalized EKF architecture allows that the modeled errors are fed back into the individual sensors.

8.5.2 Geometrically-Constrained Navigation

System error dynamics

Consider again R IMUs. They can be processed using separate blocks, each based on a single IMU mechanization and a particular value $(\mathbf{x}^l)^{(i)}$ for the ith block with $i = 1, \ldots, R$. Let $\boldsymbol{C}$ be defined as a matrix whose rows consist of all possible combinations of the R IMUs, i.e. the elements of the sequence $\{1, 2, \ldots, R\}$, taken two elements at a time. Therefore, $\boldsymbol{C}$ has size $\left(\frac{R!}{((R-2)!2!)} \times 2\right)$. For example, $\boldsymbol{C}$ is given by

$$\boldsymbol{C} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 2 & 3 \end{bmatrix} \tag{8.5.15}$$

when considering $R = 3$. The inertial navigation model associated to the ith block is given by

$$(\dot{\mathbf{x}}^l)^{(i)} = \begin{bmatrix} (\dot{\mathbf{r}}_e^l)^{(i)} \\ (\dot{\mathbf{v}}_e^l)^{(i)} \\ \dot{\mathbf{q}}_{s_i}^l \\ \hline \dot{\mathbf{r}}_{s_i \to s_j}^{s_i} \\ \dot{\mathbf{C}}_{s_j}^{s_i} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{(i)}^{-1}(\mathbf{v}_e^l)^{(i)} \\ \mathbf{C}_{s_i}^l \mathbf{f}^{s_i} - \left[2(\boldsymbol{\omega}_{ie}^l)^{(i)} + (\boldsymbol{\omega}_{el}^l)^{(i)}\right] \times (\mathbf{v}_e^l)^{(i)} + (\mathbf{g}^l)^{(i)} \\ \frac{1}{2}\mathbf{q}_{s_i}^l \otimes \left[\boldsymbol{\omega}_{ls_i}^{s_i}\right]_q \\ \hline \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{8.5.16}$$

where $\mathbf{r}^{s_i}_{s_i \to s_j}$ is the leverarm vector relating the origin of the s_i-frame with all frames s_j such that the frame pairs (s_i, s_j) are the lines of $\boldsymbol{C}$ starting with i. Taking again the example in Eq. (8.5.15) and $i = 1$, we have the following pairs (s_1, s_2) and (s_1, s_3) in $(\mathbf{x}^l)^{(i)}$. The matrix $\mathbf{C}^{s_j}_{s_i}$ expresses the relative misalignment between s_i-frame and s_j-frames.

Considering no correlation between blocks, the error state space model becomes

$$\begin{bmatrix} (\delta\dot{\mathbf{x}}^l)^{(1)} \\ (\delta\dot{\mathbf{x}}^l)^{(2)} \\ \vdots \\ (\delta\dot{\mathbf{x}}^l)^{(R)} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^{(1)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^{(2)} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}^{(R)} \end{bmatrix} + \begin{bmatrix} \mathbf{G}^{(1)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{(2)} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}^{(R)} \end{bmatrix} \begin{bmatrix} \mathbf{q}^{(1)} \\ \mathbf{q}^{(2)} \\ \vdots \\ \mathbf{q}^{(R)} \end{bmatrix} \tag{8.5.17}$$

where each error state vector $(\delta\dot{\mathbf{x}}^l)^{(i)},\ i = 1, \ldots, R$ is expressed in its augmented form (see Eq. (1.7.27)) as

$$\begin{bmatrix} (\delta\dot{\mathbf{r}}^l_e)^{(i)} \\ (\delta\dot{\mathbf{v}}^l_e)^{(i)} \\ (\dot{\boldsymbol{\varepsilon}}^l)^{(i)} \\ \delta\dot{\mathbf{x}}^{(i)}_a \\ \delta\dot{\mathbf{x}}^{(i)}_g \\ \hline \delta\dot{\mathbf{r}}^{s_i}_{s_i \to s_j} \\ \dot{\boldsymbol{\psi}}^{s_i}_{s_i \to s_j} \end{bmatrix} = \left[\begin{array}{ccccc|cc} \mathbf{F}^{(i)}_{rr} & \mathbf{F}^{(i)}_{rv} & \mathbf{F}^{(i)}_{r\varepsilon} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}^{(i)}_{vr} & \mathbf{F}^{(i)}_{vv} & \mathbf{F}^{(i)}_{v\varepsilon} & \mathbf{C}^l_{s_i}\mathbf{F}^{(i)}_{va} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}^{(i)}_{\varepsilon r} & \mathbf{F}^{(i)}_{\varepsilon v} & \mathbf{F}^{(i)}_{\varepsilon\varepsilon} & \mathbf{0} & \mathbf{C}^l_{s_i}\mathbf{F}^{(i)}_{\varepsilon g} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}^{(i)}_{aa} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}^{(i)}_{gg} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{r_{ij}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{\psi_{ij}} \end{array}\right] \begin{bmatrix} (\delta\mathbf{r}^l_e)^{(i)} \\ (\delta\mathbf{v}^l_e)^{(i)} \\ (\boldsymbol{\varepsilon}^l)^{(i)} \\ \delta\mathbf{x}^{(i)}_a \\ \delta\mathbf{x}^{(i)}_g \\ \hline \delta\mathbf{r}^{s_i}_{s_i \to s_j} \\ \boldsymbol{\psi}^{s_i}_{s_i \to s_j} \end{bmatrix}$$
$$+ \left[\begin{array}{cccc|cc} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}^l_{s_i} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^l_{s_i} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}^{(i)}_a & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}^{(i)}_g & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_{r_{ij}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_{\psi_{ij}} \end{array}\right] \begin{bmatrix} -\boldsymbol{\nu}^{(i)}_a \\ \boldsymbol{\nu}^{(i)}_g \\ \boldsymbol{w}^{(i)}_a \\ \boldsymbol{w}^{(i)}_g \\ \hline \mathbf{w}_{r_i} \\ \mathbf{w}_{\psi_i} \end{bmatrix}. \tag{8.5.18}$$

The elements of the upper part of the model are given in Section 1.7.4, considering the states associated to each block i, and $\mathbf{C}^l_b = \mathbf{C}^l_{s_i}$. Although supposed to be constant (see Eq. (8.5.16)), we model the inter-IMU leverarm error state $\delta\mathbf{r}^{s_i}_{s_i \to s_j}$ as a first-order Gauss-Markov process with inverse correlation time $t_c = 1/\beta$ "fixed" to infinity. Regarding inter-IMU misalignment angles, we supposed that initial approximations of $\mathbf{C}^{s_j}_{s_i}$ are available. This enables the small angles approximation of $\mathbf{C}^{s_j}_{s_i} \approx \mathbf{I} - [\boldsymbol{\psi}^{s_i}_{s_i \to s_j} \times]$ according to Eq. (1.2.18) in which $\boldsymbol{\psi}^{s_i}_{s_i \to s_j}$ represent the misalignment angles between the s_i-frame and the s_j-frames expressed in the s_i-frame.

Geometrically-Constrained Extended Kalman Filter

An EKF based on the developed dynamic model can be designed, with the particularity that it can be fed not only with external measurements (e.g. absolute position or velocity), but also with pseudo-measurements expressing the relative alignment between the individual IMUs kept as a constant. The resulting geometrically-constrained EKF principle is schematized in Figure

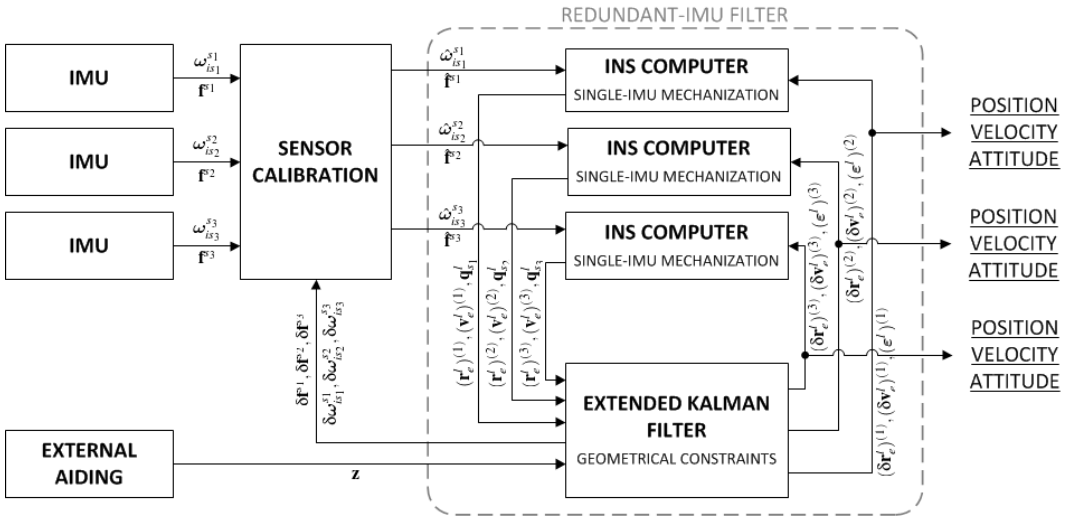

Figure 8.5: Principle of the geometrically-constrained extended Kalman filter.

8.5. As in the case of the generalized EKF, the geometrically-constrained EKF allows estimating the individual sensor biases and detecting defective sensors. The strapdown mechanization algorithms for each block i are identical to the single-IMU algorithms described in Chapter 2. In the sequel, we develop the associated (pseudo-)measurement models.

Relative Coordinate Update The relative position measurement model at digital time $\{t_k : k \in \mathbb{Z}\}$ between IMUs i and j can be written as

$$\mathbf{0} - \boldsymbol{\nu}_k^{(ij)} = (\mathbf{r}_e^l)_k^{(i)} + (\mathbf{D}^{-1})_k^{(i)} (\mathbf{R}^l)_k^{(i)} - (\mathbf{r}_e^l)_k^{(j)} \tag{8.5.19}$$

where $\boldsymbol{\nu}_k^{(ij)}$ are residuals modeled as white noise, and $(\mathbf{R}^l)_k^{(i)} = \left[(\mathbf{C}_{s_i}^l)_k (r_{s_i \to s_j}^{s_i})_k \times \right]$. Linearization of this equation yields the following design matrix:

$$\begin{aligned} \mathbf{H}_k^{(ij)} = \Big[\; & \mathbf{I} + (\mathbf{D}_a)_k^{(i)} \quad \mathbf{0}_{3\times 3} \quad -(\mathbf{D}^{-1})_k^{(i)} (\mathbf{R}^l)_k^{(i)} \quad \mathbf{0}_{3\times Z_a^i} \quad \mathbf{0}_{3\times Z_g^i} \quad (\mathbf{D}^{-1})_k^{(i)} (\mathbf{C}_{s_i}^l)_k \quad \mathbf{0}_{3\times 3} \; \Big| \cdots \\ & \cdots \Big| \; -\mathbf{I}_{3\times 3} \quad \mathbf{0}_{3\times 3} \quad \mathbf{0}_{3\times 3} \quad \mathbf{0}_{3\times Z_a^j} \quad \mathbf{0}_{3\times Z_g^j} \quad \mathbf{0}_{3\times 3} \quad \mathbf{0}_{3\times 3} \; \Big] \end{aligned} \tag{8.5.20}$$

where $Z_\bullet^\bullet$ are the number of error states associated to the accelerometers (a) and gyroscopes (g) biases in the ith and jth block, and the auxiliary matrix $\mathbf{D}_a^{(i)}$ is given in Eq. (2.4.4) in which $(\mathbf{x}^l)^{(i)}$ is considered. Note that the first half of the design matrix $\mathbf{H}_k^{(ij)}$ corresponds to the $(\delta\mathbf{x}^l)^{(i)}$ state while the second half is aligned on the $(\delta\mathbf{x}^l)^{(j)}$ state.

Relative Velocity Update The relative velocity measurement model at digital time $\{t_k : k \in \mathbb{Z}\}$ between IMUs i and j can be written as

$$\mathbf{0} - \boldsymbol{\nu}_k^{(ij)} = (\mathbf{v}_e^l)_k^{(i)} + (\boldsymbol{\Omega}_{ls_i}^l)_k (\mathbf{C}_{s_i}^l)_k (\mathbf{r}_{s_i \to s_j}^{s_i})_k - (\mathbf{v}_e^l)_k^{(j)} \tag{8.5.21}$$

where again $\boldsymbol{\nu}_k^{(ij)}$ are residuals modeled as white noise and $(\boldsymbol{\Omega}^l_{ls_i})_k = \left[(\boldsymbol{\omega}^l_{ls_i})_k\times\right]$ with

$$(\boldsymbol{\omega}^l_{ls_i})_k = (\mathbf{C}^l_{s_i})_k(\boldsymbol{\omega}^{s_i}_{is_i})_k - (\boldsymbol{\omega}^l_{il})^{(i)}_k. \tag{8.5.22}$$

After linearization, the design matrix becomes:

$$\begin{aligned}\mathbf{H}^{(ij)}_k = \Big[\ & (\mathbf{R}^l)^{(i)}_k(\mathbf{D}_1)^{(i)}_k \quad \mathbf{I} + (\mathbf{R}^l)^{(i)}_k(\mathbf{D}_2)^{(i)}_k \quad (\mathbf{R}^l)^{(i)}_k\left[(\boldsymbol{\Omega}^l_{il})_k + 2(\boldsymbol{\Omega}^l_{ls_i})_k\right] \quad \mathbf{0}_{3\times Z^i_a} \quad \mathbf{0}_{3\times Z^i_g} \\ & (\boldsymbol{\Omega}^l_{ls_i})_k(\mathbf{C}^l_{s_i})_k \quad \mathbf{0}_{3\times 3} \ \Big|\ \cdots \ \Big|\ \mathbf{0}_{3\times 3} \quad -\mathbf{I} \quad \mathbf{0}_{3\times 3} \quad \mathbf{0}_{3\times Z^j_a} \quad \mathbf{0}_{3\times Z^j_g} \quad \mathbf{0}_{3\times 3} \quad \mathbf{0}_{3\times 3}\ \Big]\end{aligned} \tag{8.5.23}$$

with auxiliary matrices

$$\mathbf{D}^{(i)}_1 = \left(\frac{\partial\boldsymbol{\Omega}^l_{il}}{\partial\mathbf{r}^l_e}\right)^{(i)} \quad \text{and} \quad \mathbf{D}^{(i)}_2 = \left(\frac{\partial\boldsymbol{\Omega}^l_{il}}{\partial\mathbf{v}^l_e}\right)^{(i)} \tag{8.5.24}$$

as given in Eq. (2.4.9) and (2.4.10).

Relative Orientation Update The relative orientation measurement model at digital time $\{t_k : k \in \mathbb{Z}\}$ between IMUs i and j can be expressed as:

$$\mathbf{0}_{3\times 3} - \boldsymbol{V}^{(ij)} = \mathbf{I}_{3\times 3} - \left[(\mathbf{C}^l_{s_i})_k\right]^T(\mathbf{C}^l_{s_j})_k(\mathbf{C}^{s_j}_{s_i})_k \tag{8.5.25}$$

where $\boldsymbol{V}^{(ij)}_k$ is a 3×3 diagonal matrix containing normaly distributed residuals and $(\mathbf{C}^{s_j}_{s_i})_k$ is obtained from $(\boldsymbol{\psi}^{s_i}_{s_i\to s_j})_k$. The analytically linearized expression of Eq. (8.5.25) is long and not that straightforward. An alternative for obtaining $\mathbf{H}^{(ij)}_k$ is numerical differentiation.

8.6 Realization of Distributed IMU

We built a DIMU composed of three *XSens MTx* IMUs and one *XSens MTi-G* IMU sampling at 100 Hz. The latter includes a L1 GPS receiver. Also, the gyroscopes of this IMU are *not* of the same type as the ones packaged in the *MTx* devices. In order to respect the third assumption in Section 8.3, only the inertial observations sensed by the *MTx* IMUs will be used in the processing. The three *MTx* units are connected to a *XBus Master* unit which is responsible for powering and triggering the individual IMUs, as well as for transmitting the data to a *Personal Computer* (PC). The four IMUs were mounted on a tetrahedron whose shape is known and corresponds to the optimal geometry for which the information volume is highest. More details regarding this setup can be found in Waegli [2009].

All devices sample in a common time frame. Similarily to the setup described in Section 7.4.1 of Chapter 7, the GPS clock contained in the *MTi-G* device yields access to a globally available timing and synchronization framework, the GPS time. An experimental setup was built such that all individual *MTx* IMUs sample simultaneously in time. With this respect, we implemented a real-time C++ based DIMU datalogger in which two synchronization strategies have been considered:

- The *Software Synchronization* mode in which time synchronization is performed by using the *Operating System* (OS) time $\{t_k^{OS} : k = 1, 2, \ldots\}$ as a timing base. The time offset between GPS time $\{t_m^{GPS} : m = 1, 2, \ldots\}$ and the OS time, denoted $\Delta t^{OS,GPS}$, is continuously estimated at each GPS package arrival to calibrate the IMU package time $\{t_k^{IMU} : k = 1, 2, \ldots\}$ based on the IMU sample counter interval. The synchronized time $\{t_k : k = 1, 2, \ldots\}$ can then be computed as

$$t_k = t_k^{IMU} + t_k^{OS} + \Delta t^{OS,GPS} \tag{8.6.1}$$

  where $\Delta t^{OS,GPS}$ is estimated at each GPS packet arrival using $\Delta t^{OS,GPS} = t_m^{GPS} - t_k^{OS}$. Note that due to the low bandwith featuring GNSS sensors, the m cycle is usually much slower than the k cycle. However, relying on data packets arrivals can be problematic due to variable processing speeds and transmission of the IMU packets on the sensor end. Furthermore, indeterminate timing behaviors from software-based synchronization on a non-real-time OS can induce timing errors higher than ten milliseconds Perry and Childs [2009].

- The *Hardware Synchronization* mode in which a periodic *Transistor-Transistor-Logic* (TTL) pulse (1 PPS) delivered by the GPS receiver serves as base for the measurements. This pulse is aligned to GPS time. The PPS signal is brought as input through a manufactured cable to the *XBus Master* as a scale that aligns the triggering of all *MTx* IMUs (i.e. their internal sensors). The absolute timing of the IMU messages is performed by the DIMU datalogger which exploits available GPS time messages and the guaranteed (by hardware) synchronous triggering of *MTx* internal sensors.

The precision of the relative timing is depicted in both panels of Figure 8.6 representing measured sampling periods $\Delta t_k = t_k - t_{k-1}$ for digital times $\{t_k : k = 1, \ldots, N\}$. The black curves with circles represent the measured sampling periods belonging to the *MTi-G* IMU. The gray curve shows the measured sampling interval belonging to one of the *MTx* devices forming the DIMU (gray curve). All devices sampled at 100 Hz. The upper and lower panels show results provided by the DIMU datalogger running in software and hardware synchronization mode, respectively. Despite the *MTi-G* uses its GPS to correct its internal clock bias, some jittering is observed from the deviations around the theoretical sampling period of 10 milliseconds. The sampling intervals of the *MTx* IMU match perfectly the ones belonging to the *MTi-G* (lower panel), which is not the case when using the software synchronization mode (upper panel). Although not shown here, this was also the case for all the remaining *MTx* devices forming the DIMU. From these results, it can safely be concluded that the relative timing capability is achieved with an acceptable precision (estimated below 1 millisecond) when performing hardware synchronization.

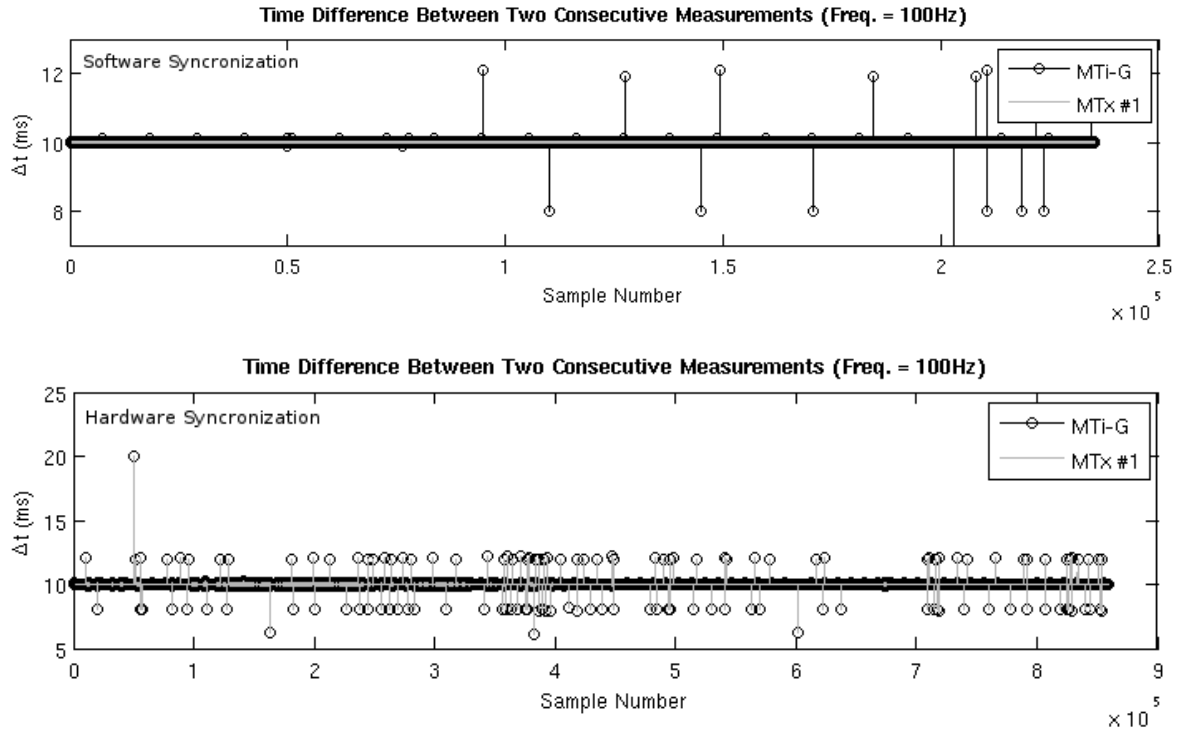

Figure 8.6: DIMU datalogger relative timing capability in software (upper panel) and hardware (lower panel) synchronization mode. Note the jittering affecting the MTi-G internal clock which enables to highlight the perfect alignment in hardware synchronization mode.

Chapter 9

Redundant INS/GPS Performance

9.1 Introduction

This chapter discusses the typical results obtained from the realized MEMS-based DIMU introduced in Chapter 8 and processed with the implemented redundant navigation filters. Since urban environment provides challenging conditions for integrated navigation, all the selected experiments presented in this chapter are issued from datasets collected with a car. Indeed, an inertial platform mounted on a car is subject to low dynamics (e.g. along the pitch and yaw vehicle axes) as well as to abrupt changes in dynamics (e.g. due to stops at crossroads). Moreover, a car may pass through tunnels, urban corridors and forests where the reception of GNSS signals is either not possible or of low quality. These elements may result in conditions in which the navigation quality may benefit from the employment of augmented redundant INS.

In Section 9.2, we present the experimental setup we designed for collecting reference and DIMU data with a car. Section 9.3 is dedicated to the description of the selected compaigns. Then we analyse the performance of the EKF based on different synthetic IMU computers in Section 9.4, and of the geometrically-constrained EKF in Section 9.5. Finally, an overall performance comparison is proposed in Section 9.6.

9.2 Experimental Setup

Since performance evaluation requires the existence of a reference, a system of considerably superior quality was operated and its output considered as the "truth" for comparison. The experimental setup from which observations were collected is depicted in Figure 9.1. In the sequel, we provide details on each of the delimited subsystems.

9.2.1 High-Grade (Reference) Navigation System

The high-precision navigation system is composed of a navigation-grade IMU (*Ixsea Airins* sampling at 100 Hz) and a geodetic-grade L1/L2 GPS/Glonass rover receiver (*JAVAD Alpha*

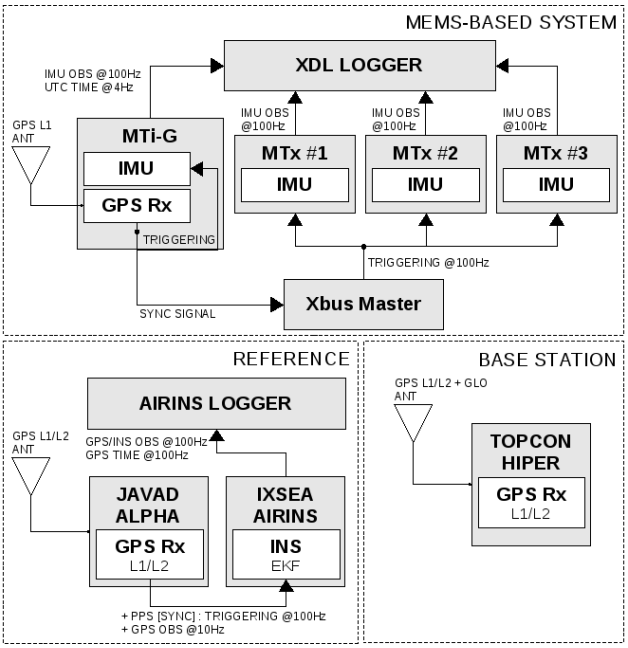

Figure 9.1: Instrumental setup used for evaluating the DIMU performance.

sampling at 10 Hz) connected to a L1/L2/Glonass active antenna (*JAVAD Legacy*). A L1/L2 GPS/Glonass receiver (*Topcon Hiper Pro* sampling at 10 Hz) is setup as base station for differential processing. A hardware synchronization scheme aligns the *Airins* time frame with GPS time using the PPS signal output from the *Alpha* receiver. Finally, a specially developed C++ based datalogger, designated as *Airins Logger*, tags the raw *Airins* samples with GPS time in real-time. The base and rover GPS receivers are combined to compute a post-processed double-differenced *Carrier-Phase* (CP) solution which feeds a single-IMU based INS/GNSS EKF. The reference navigation solution is then obtained using a smoother which combines forward and backward EKF solutions (see Chapter refch:fundamentals of integrated navigation).

9.2.2 Redundant (MEMS-based) Inertial Navigation System

The navigation system under test comprises two DIMUs of the type described in Section 8.6 (see right image in Figure 9.2). Hence, eight MEMS-based IMUs (two *XSens MTi-G* and six *XSens MTx* sampling at 100 Hz) mounted on two regular tetrahedrons with known geometry are forming the redundant MEMS-based INS. However, from the six *XSens MTx* units, two were identified to be badly calibrated for constant systematic errors and contained sensors with higher noise level. Since both were mounted on the same tetrahedron, we consider each DIMU as a separate system, meaning that all comparisons will be done using the correctly calibrated DIMU.

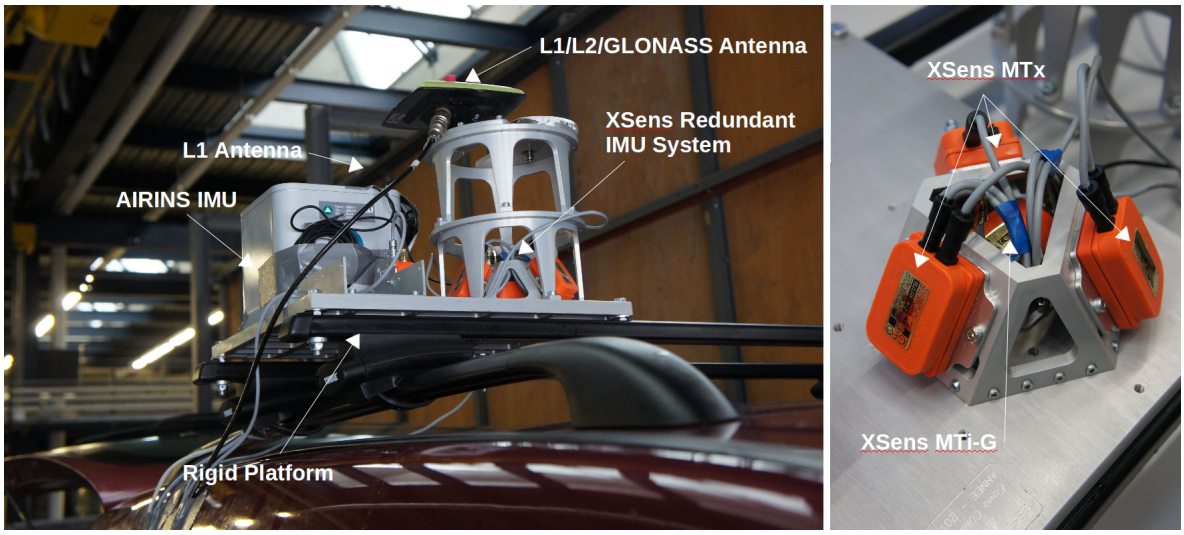

Figure 9.2: Navigation sensor platform mounted on the roof of the car (left picture) and view on the skew-redundant *XSens* inertial sensors forming the realized DIMU (right picture).

9.3 Campaign Description

Several runs were performed with a car in urban and sub-urban conditions. A rigid metallic platform on which both, the high-grade and MEMS-based redundant INS described in the previous section can be rigidly fixed has been designed and manufactured to be mounted on the roof of a car (see left image in Figure 9.2). Datasets issued from two measurement campaigns will be analysed.

The Urban Campaign The dataset depicted in the left panel of Figure 9.3 is issued from a track of approximately 10 km length collected in urban conditions. The car came across two tunnels (the first 40 seconds duration) and passed under several bridges. In these cases, no GNSS signals were available. Red lights and crossroads made the car stop many times during periods lasting from a few seconds to several minutes. A longer dataset acquired during the urban campaign will be used to illustrate the MRS synthetic IMU (see Figure 9.5).

The Campus Campaign The dataset depicted in the right panel of Figure 9.3 contains observations from a track on the university campus. The GNSS signal availability was very good during almost the complete run except when the car came across an underground parking garage. Note that the circles in the trajectory represent round-abouts.

On both panels in Figure 9.3, the reference trajectory together with the CP post-processed *differential GPS* (CP-DGPS) positions are drawn as a black curve and gray circles, respectively.

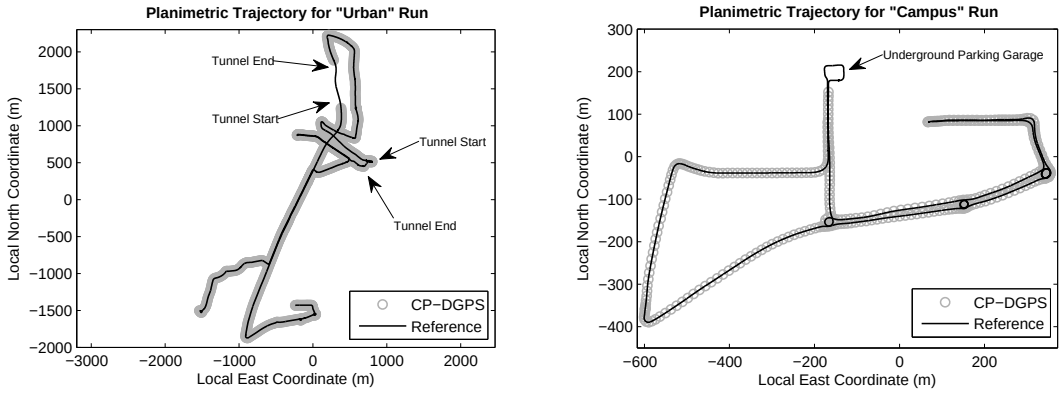

Figure 9.3: High precision reference trajectory of the *urban* (upper panel) and *campus* (lower panel) dataset collected with a car.

In the sequel, we process the observations issued from the augmented MEMS-based INS using the different redundancy architectures developed in Chapter 8. All individual sensors biases are modeled using models estimated by means of the GMWM method presented in Chapter 6.

9.4 Synthetic IMU Performance

Synthetic IMUs will be computed out of three calibrated *MTx* devices using the different weighting strategies developed in Chapter 8

9.4.1 Two-Regimes Markov Regime Switching Modeling

The results of the application of the MRS modeling on one gyroscope and accelerometer signal (Y-axis) of *MTx* #1 is presented as an example in Figure 9.4[1]. We considered a two regimes MRS, i.e. the regime $(S_{i,j})_k \in \{1, 2\}$ and $S^{\star}_{i,j} = 2$. We assume that these two regimes discriminate the static and the dynamic cases. This hypothesis is relevant and sufficient in the context of car navigation. Furthermore, the selection of the number of AR$(\cdot)$ coefficients was realized by retaining the model yielding the minimum BIC value among tested AR$(\cdot)$ processes with $d_s = 1, \ldots, 10$. In Figure 9.4, only a portion of the complete signal is shown for clarity. The upper panels show the estimated sensor (gyroscope/accelerometer) residuals (gray curves), i.e. $\hat{\varepsilon}_k = (y_{1,2})_k - \hat{\mu}_{2,k}$ for times $\{t_k : k = 1, \ldots, N\}$ where $(y_{1,2})_k$ is the sensor signal and $\hat{\mu}_{2,k}$ is the synthetic sensor computed using Eq. (8.4.7). The black curves are the noise level (standard deviation) obtained through MRS modeling, i.e. $(\hat{\gamma}_{1,2})_k$ from Eq. (8.4.38), which will weight the individual sensors. The lower panels depict the sensor signals $(y_{1,2})_k$ (gray curves) and the estimated smoothed probabilities $\hat{\xi}^{i,j}_{s|k}$ (black curves, see Eq. (8.4.26)) to be either in regime 1 $(s = 1)$ or regime 2 $(s = 2)$. The probabilities are not clearly separated since they are oscillating around 0.5. Hence, this could be interpreted as a difficulty of the MRS to discriminate both regimes. Consequently, the gyroscope residuals have a constant estimated noise level with time and hence the unweighted form may be sufficient in this case. However, the effects of the

[1]Identical results were obtained for all remaining sensors.

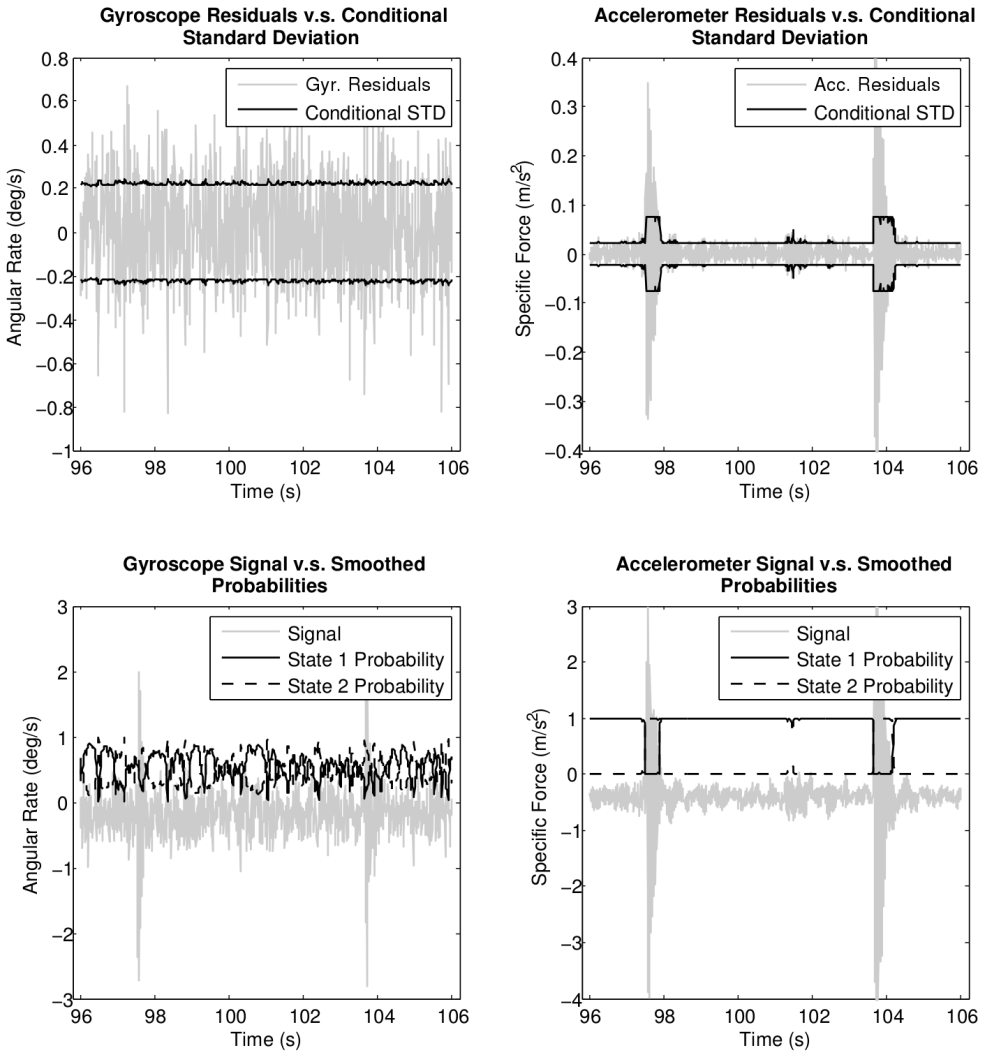

Figure 9.4: Result of the MRS modeling on *XSens MTx* gyroscope and accelerometer signals. The upper panels show the residuals $\{\hat{\varepsilon}_k\}$ computed on the Y-axis gyroscope (left column) and the Y-axis accelerometer (right column) as gray lines, together with the estimated noise level obtained through two-regimes MRS modeling (black curves). The lower panels compare the raw signals $(\{y_{i,j})_k\}$ (gray curves) with the estimated smoothed probabilities to be either in regime 1 or regime 2. The correlation between the detected regimes and the dynamics is clearly visible in the accelerometer case.

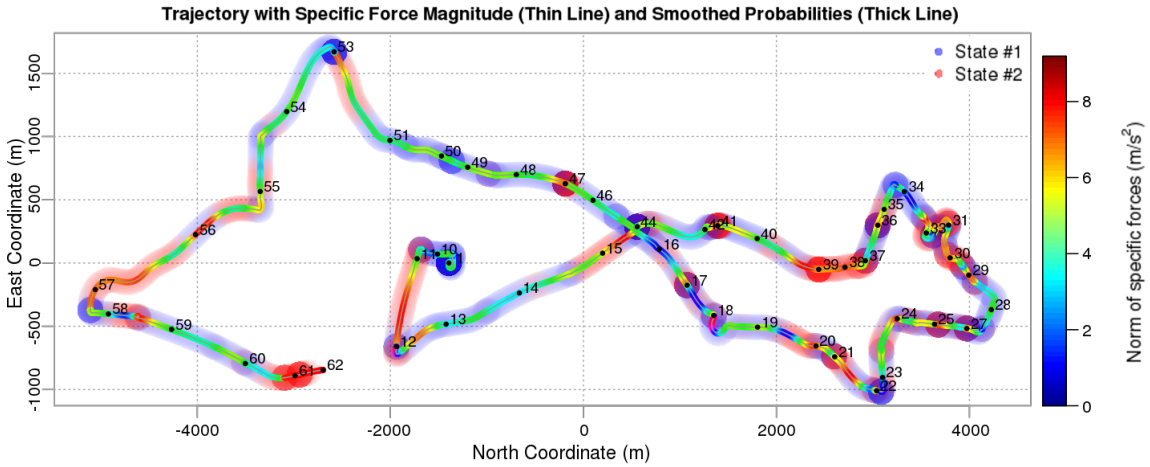

Figure 9.5: Map of the test trajectory on which the norm of the specific force is represented (colored thin line) and the smoothed probabilities issued from the regime-switching modeling are superposed (thicker transparent line). Note that due to the scale of the map, short term variations are not visible.

dynamics (through the vibrations) are clearly visible in the estimated accelerometer residuals $\hat{\varepsilon}_k$. The estimated noise level (upper right panel) is successfully changed during these periods according to the estimated smoothed probabilities which are clearly separated into 0.0 and 1.0 probabilities (lower right panel). In this case, the two regimes clearly discriminate the static (full black curve) from the dynamic periods (dashed black curve).

Figure 9.5 illustrates an example of trajectory acquired with a car during the *urban* campaign. The colors of the fine curve represent the norm of the b-frame specific force (used to represent dynamics), while the transparent superposed line provides the estimated smoothed probability to be in regime 1 (blue) or regime 2 (red). Note that due to the scale of the figure, only global trends are visible. The numbers along the trajectory represent the time of running from the starting point. In general, the "dynamic" regime is successfully detected at portions were strong linear accelerations or curves occured. Static periods at red lights are also clearly visible for example after 10, 44 or 53 minutes.

9.4.2 ARMA-GARCH Modeling

The results of ARMA-GARCH modeling are illustrated in Figure 9.6 for the estimated gyroscope and accelerometer residuals, $\hat{\varepsilon}_k$, on the same axis and time period as in the preceding section, i.e. $\hat{\varepsilon}_k = (y_{1,2})_k - \hat{\mu}_{2,k}$ for times $\{t_k : k = 1, \ldots, N\}$ where $(y_{1,2})_k$ is either the gyroscope or accelerometer signal (see Eq. (8.4.7)). The black curves depict the noise level $(\hat{\gamma}_{1,2})_k$ (representative for the individual sensor weighting) estimated by the GARCH$(\cdot,\cdot)$ on the ARMA$(\cdot,\cdot)$ residual sequence. Again, a constant noise level $(\hat{\gamma}_{1,2})_k$ is estimated on the gyroscope residuals (left panel). Note that due to the scale of the figure, the slight variations in $(\hat{\gamma}_{1,2})_k$ are not visible. However, the noise level of the accelerometer varies with dynamics (right panel). Compared to the MRS based computer, the noise level is better estimated in this case.

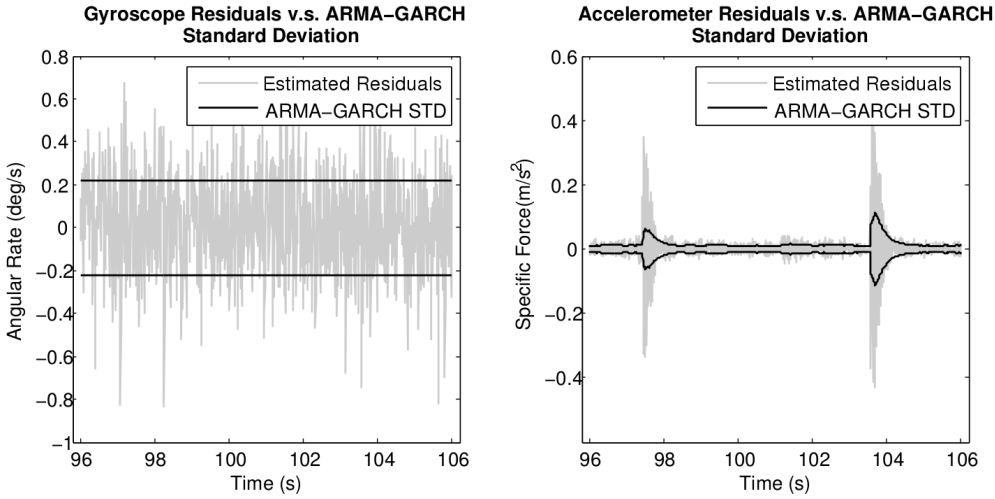

Figure 9.6: Result of the ARMA-GARCH modeling on MEMS-based gyroscope and accelerometer signals. The panels show the $\{\hat{\varepsilon}_k\}$ residuals computed on the Y-axis gyroscope (left panel) and the Y-axis accelerometer (right panel) as gray lines, together with the noise level $\hat{\gamma}_k$ (black curves) obtained through ARMA-GARCH modeling.

Since GARCH$(\cdot,\cdot)$ models are based on the assumption that the signal to model, i.e. the sensor residual sequence $\{\varepsilon_k : k = 1, \ldots\}$ of Eq. (8.4.13), is unpredictable (thus only composed of white noise), the ARMA modeling step has to fullfill this condition, considering from Chapter 7 and Stebler et al. [2011b], Waegli et al. [2010] that the inertial sensor errors are composed of colored noise. In Figure 9.7, the upper panels show the sensor residuals $\{\hat{\varepsilon}_k\}$ (black curves) and the ARMA$(\cdot,\cdot)$ residuals $\{u_k : k = 1, \ldots, N\}$ (gray curves) of Eq. (8.4.13) for the gyroscope and accelerometer over longer time period (8 minutes). The middle and lower panels illustrate the ACS and the Allan variance curves computed on both types of residuals. The whitening effect of the ARMA modeling is confirmed by a the ACS of $\{\hat{\varepsilon}_k\}$ which becomes close to a Dirac (gray curve) when computed on the ARMA$(\cdot,\cdot)$ residual sequence $\{u_k\}$, and an Allan variance curve shape (black curve with dots) computed on $\{\hat{\varepsilon}_k\}$ which tends to the line with slope $-1/2$ (this slope corresponds to a Gaussian white noise and is drawn as a dashed-black line on the same figure) when computed on $\{u_k\}$ (gray curve with triangles). For the gyroscope, this step can clearly be removed since the sensor residuals sequence $\{\hat{\varepsilon}_k\}$ is already white in this case.

9.4.3 Noise Reduction

The resulting noise reduction of the weighting process using MRS and ARMA$(\cdot,\cdot)$-GARCH$(\cdot,\cdot)$ models is compared to the unweighted synthetic IMU as well as to the magnitude of the estimated residuals of each individual sensor. Only gyroscope results will be presented here, since they are more important for achieving accurate navigation.

The upper panel of Figure 9.8 compares boxplots of the magnitude of the true angular rate errors (with respect to the reference provided by the *Airins* gyroscopes) of the individual *MTx* gyroscopes with the synthetic sensors. First, all three types of synthetic gyroscopes have reduced noise level, as could be expected from the theory. The values of the residual

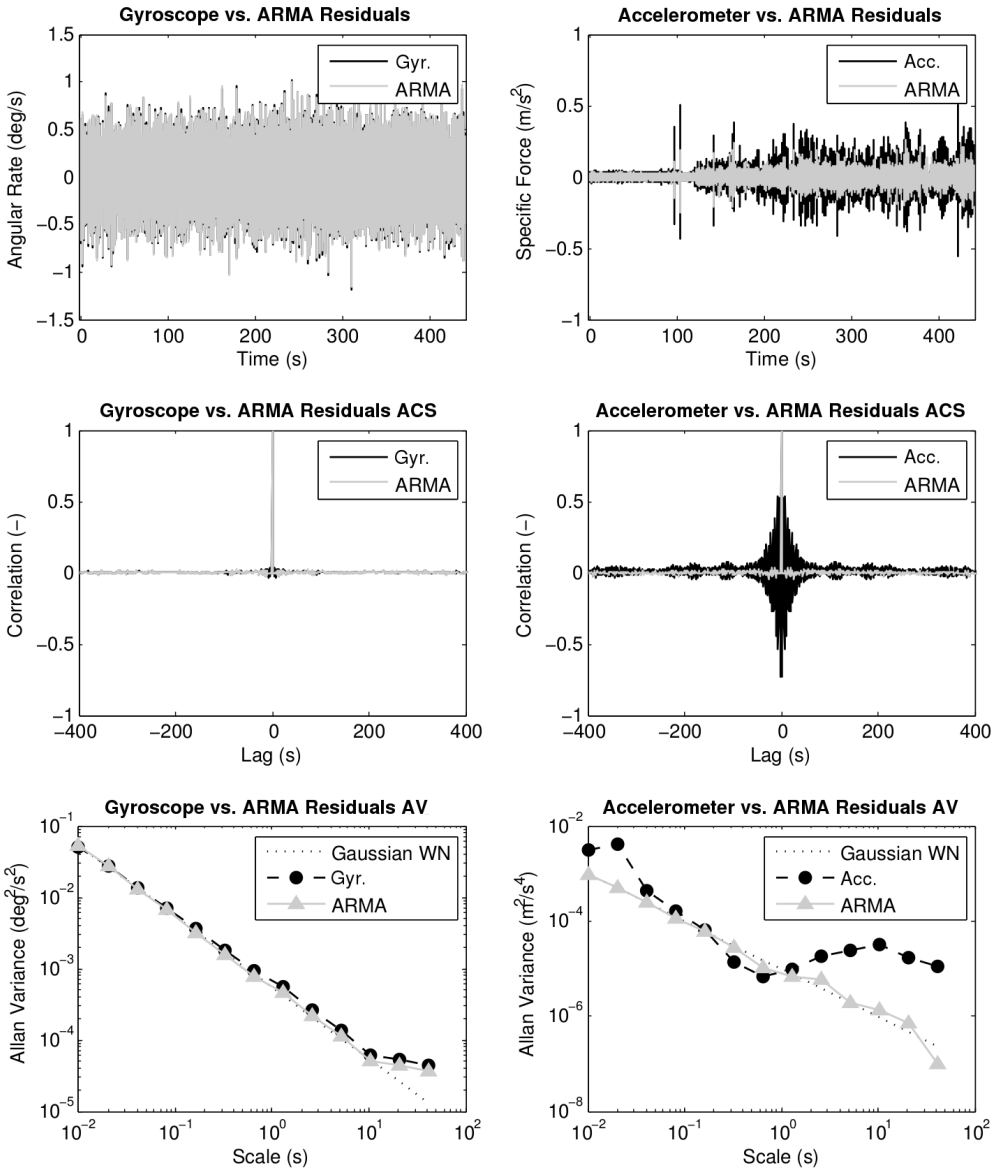

Figure 9.7: Demonstration of the whitening capability of the ARMA filters. The upper panels show the $\{\hat{\varepsilon}_k\}$ residuals (gray curves) and the residuals $\{u_k\}$ of the ARMA filters for the Y-axis gyroscope (left column) and the Y-axis accelerometer (right column). The middle and lower panels illustrate the corresponding autocorrelation and Allan variance sequences, respectively. Both information demonstrate the whiteness of the ARMA residuals $\{u_k\}$ used as input in the GARCH filters.

standard deviations are written in the boxplot. The observed noise level reduction is close to the asymptotic value of $\sqrt{3}$ (see Eq. (8.4.8)). However, since all synthetic IMU computation methods estimated rather constant noise levels in the gyroscopes (and thus equal weights), the ARMA-GARCH and MRS based synthetic IMUs have no reduced noise level compared to the unweighted form in this dataset. Therefore, the use of more sophisticated weighted synthetic IMU computers such as ARMA-GARCH and MRS is only necessary if the noise level is varying between sensors and in time. As an example, we considered a fourth *MTx* unit issued from the second DIMU. As previously explained, this sensor contains uncalibrated errors and its noise level is higher than the others. We then reprocessed the three forms of synthetic gyroscopes on the same (*campus*) trajectory. Again, we computed the residuals of the angular rate magnitude for each computed gyroscope and produced a boxplot (lower panel of Figure 9.8). The significantly higher noise level of *MTx* #4 with respect to the three others is clearly visible. In this case, the use of an ARMA-GARCH or MRS based synthetic computer is advantageous compared to the unweighted form as the noise level was reduced from 0.55 (deg/s) to 0.43 (deg/s) and 0.44 (deg/s) for the ARMA-GARCH and MRS based synthetic gyroscope, respectively.

For the accelerometers, the noise level variation is fully correlated to the vibrations. Since all accelerometers will sense the same vibrations, the estimated noise level will be increased for all of them. This further decreases the relative weighting of the individual IMUs and cancels the advantage of using the ARMA-GARCH or MRS based synthetic accelerometer against the unweighted synthetic accelerometer. Therefore, the benefits of using an ARMA-GARCH or MRS based synthetic IMU computer was limited in terms of noise reduction (not shown).

For the sake of completness, the standard deviations extracted from the synthetic IMU based EKF system noise matrix, $\mathbf{Q}_k$, are depicted in Figure 9.9 for the gyroscope noise level (left panel) and the accelerometer noise level (right panel). The benefits gained from using such type of adaptation in terms of navigation accuracy will be described in Section 9.6. Both plots are aligned with the estimated residuals plotted in Figure 8.1. By comparing the two figures, it can be seen that it is mainly the noise level of the synthetic accelerometer that accounts for the changing observation accuracy in the synthetic IMU mechanization process. The underestimated level of the accelerometer noise as issued from the calibration in static conditions and observed in the left panel of Figure 8.1 as a black curve is now adapted to realistic magnitudes when the platform is subject to vibrations.

9.5 Geometrically-Constrained EKF Performance

As previously mentioned, the geometrically-constrained INS/GNSS EKF is particularly usefull when the geometry of DIMU is poorely known. For demonstrating that, some bias (0.5 deg) was added in the misalignment angles defining the direction cosine matrices $\mathbf{C}_{s_i}^{s^j}$ (see Section 8.5.2) and two navigation solutions were computed. The first solution is issued from a geometrically-constrained EKF filter in which the misalignment angle corrections $\boldsymbol{\psi}_{s_i \to s_j}^{s_i}$ (see Eq. 8.5.18) are estimated in the state, while the second solution takes the wrong assumption on DIMU inner alignment. Figure 9.10 shows the effect on navigation performance in GNSS-denied conditions on the forward EKF solution. The true reference trajectory is shown as a thick gray curve in the upper panel, and GNSS solutions are depicted as gray circles. The map only shows a small

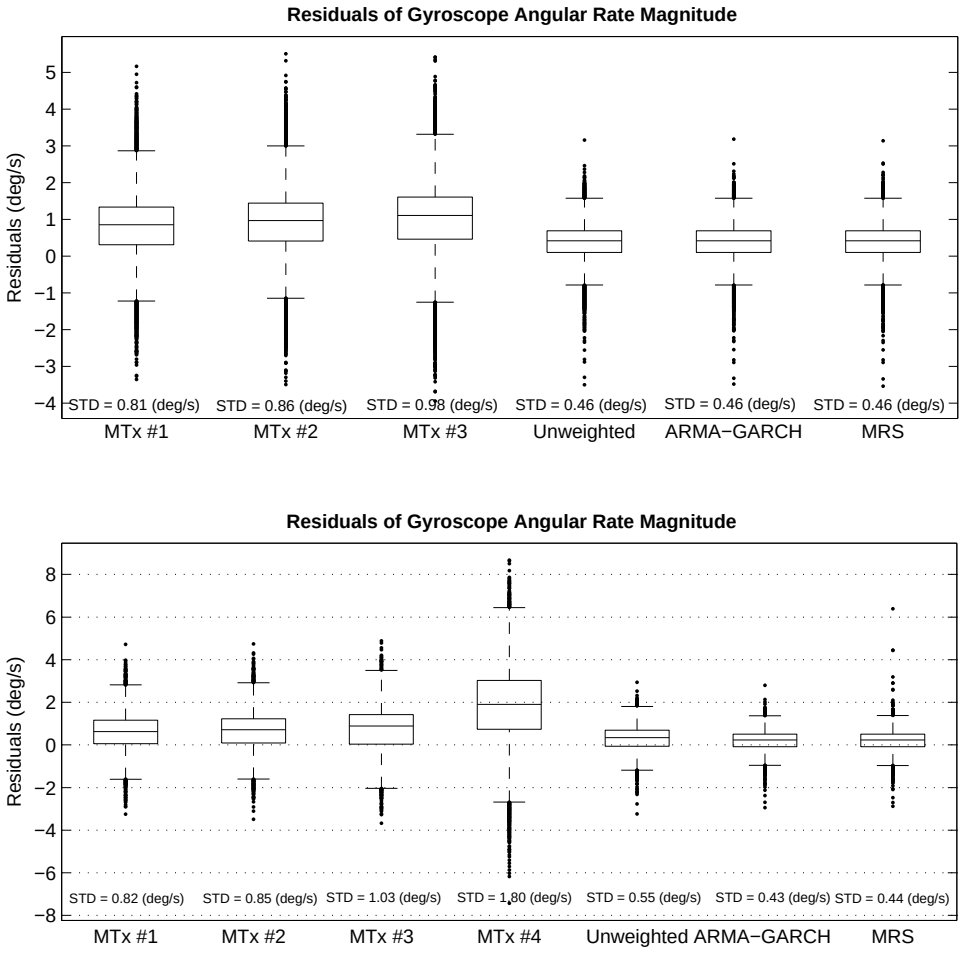

Figure 9.8: Noise reducation capability of the synthetic IMU computers for the *campus* trajectory. The panels show boxplots for the residuals of the magnitude of the MTx gyroscopes and the computed synthetic gyroscopes using three IMUs with identical noise levels (upper panel), and four IMUs among which one (MTx #4) has higher noise level (lower panel). The residuals standard deviations are noted below each corresponding boxplot.

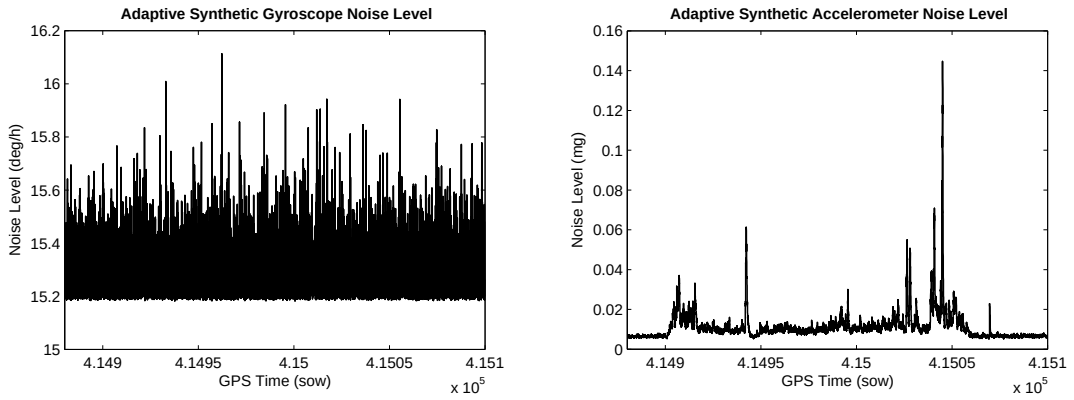

Figure 9.9: Standard deviations of the synthetic gyroscope (left panel) and accelerometer (right panel) used in the adaptive EKF system noise matrix.

portion of the complete trajectory in which additional GNSS outages are present. The effect of the bias in DIMU misalignment is highlighted in terms of planimetric and altimetric error in the lower two panels. When estimating the misalignment angle corrections (black curves), the positioning accuracy was generally increased. Such results were only obtained when the filter had sufficient time for correctly observing the states. This observability problem is intimately related to the dynamics encoutered by the platform. As stated in the introduction, the nature of car motion has little dynamics along the vehicle pitch and yaw axes. This implies that correct inter-IMU geometry estimation can only be achieved for sufficiently long datasets, large variations in dynamics, and conditions with good reception of GNSS signal.

The choice on the frequency of constraints within a DIMU system, i.e. the interval in performing the relative coordinate, velocity and attitude updates rate (see Section 8.5.2) can be made as a tradeoff between the computational load and the navigation performance. Using the same rate as the filter rate[2] implies updates at each IMU epoch. By keeping in mind that the geometrically-constrained EKF architecture involves high state dimensions with large matrices, it is obvious that this option is not optimal. For all processed datasets, the relative geometry updates were performed at the same rate as the filter covariance prediction rate, i.e. $\mathbf{P}_k^-$ in Eq. (1.9.4), which was set at 10 Hz. The effect of further decreasing this rate to 1 Hz is shown for the *campus* dataset in Figure 9.11 which presents RMSE[3] values in attitude computed with respect to the reference navigation solution provided by the high-grade navigation system. Two solutions are investigated: the first solution is issued from the geometrically-constrained EKF updated with correctly *known* (accurate) relative pseudo-measurements (designated as *known constraints* in the figure), while the second solution is issued from the same filter in which $(\boldsymbol{\psi}_{s_i \to s_j}^{s_i})_k$ for $k = 1, \ldots, N$ is estimated in the state and used for the $(\mathbf{C}_{s_i}^{s_j})_k$ cosine matrices at filter update stage (designated as *estimated constraints* in the figure). The RMSE values for the forward-filtered EKF and the (fixed-interval) smoothed navigation solutions were computed and are presented in the left and right panel of Figure 9.11, respectively. As no significant differences were observed in terms of position and velocity RMSE, only attitude RMSE are shown. The first bar (black) in each panel represents the averaged RMSE issued from the

[2]In this work, the general filter rate is equal to the IMU sampling rate, except for update times.

[3]Although RMSE is not a very representative quantity when evaluating accuracy of data containing inertial coasting periods, it still provides a good indication on the *relative* accuracy between two solutions.

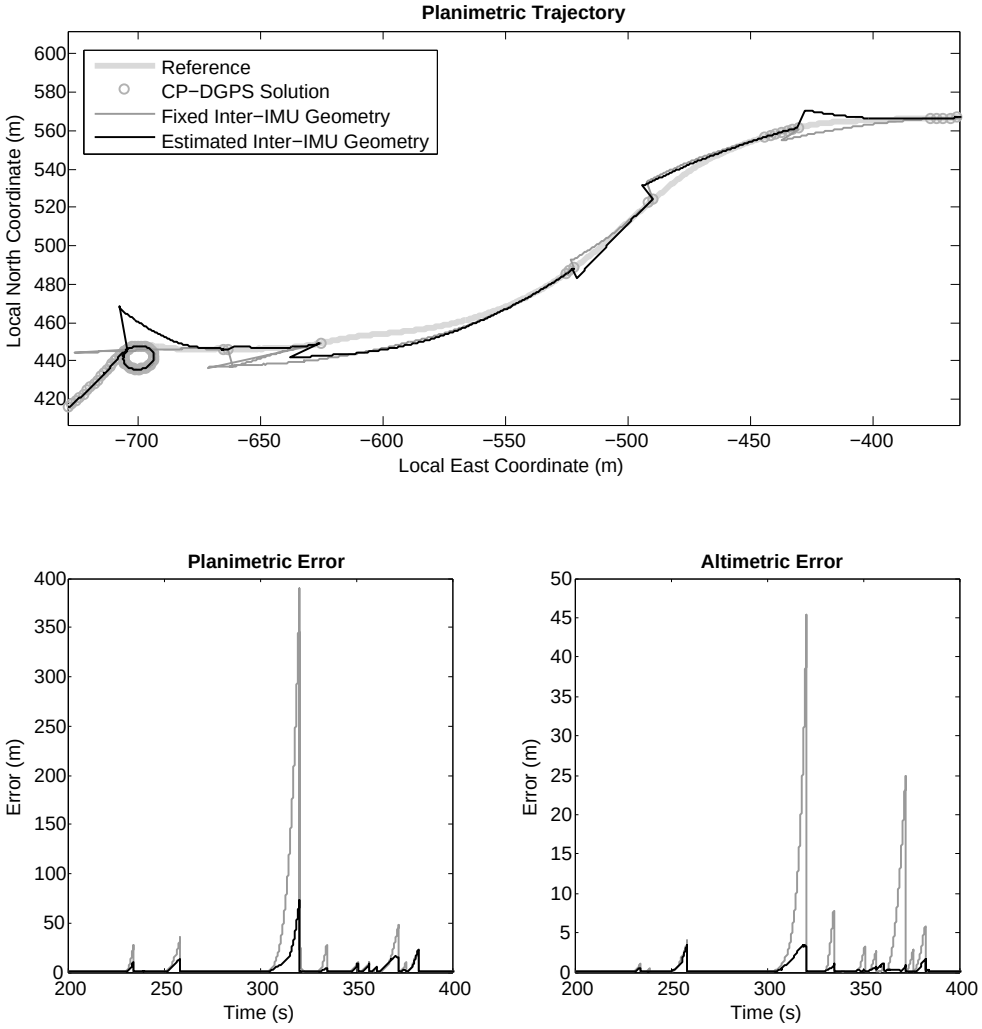

Figure 9.10: Influence of the poorely known inter-IMU boresight angles on the estimated trajectory during periods of poor GNSS signal availability. The upper panel depicts a portion of the true trajectory (thick gray curve) along with GNSS positions (circles) and the trajectories estimated using the geometrically-constrained EKF in which biased inter-IMU boresight angles are used as update (thin gray lines) and the filter in which the angles are estimated (black lines). The lower panels draw the planimetric (left panel) and altimetric (right panel) error of the solutions provided by the two filters.

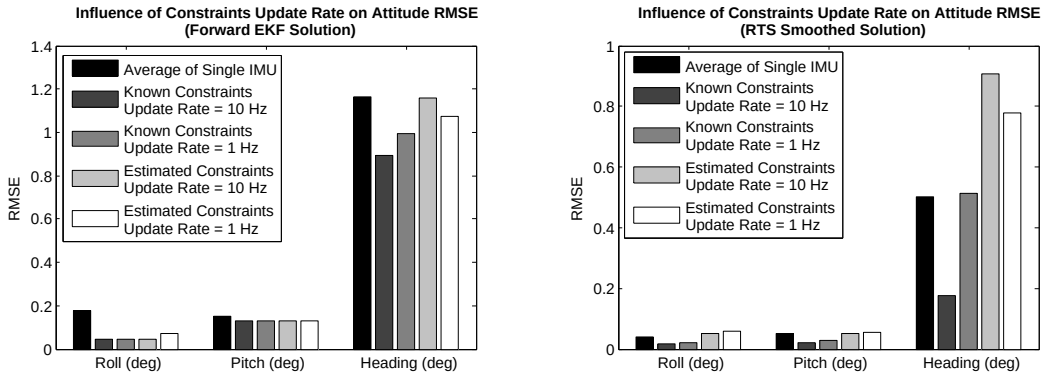

Figure 9.11: Influence of the inter-IMU geometry update rate on forward filtered (left panel) and smoothed (right panel) attitude estimation.

three individual single- IMU based EKF solutions. In the case of the forward EKF solution (left panel), there is an improvement of 75% when using fixed inter-IMU boresight angles at both rates. When estimating the boresight angles, the roll accuracy is improved by 75% and 61% at 10 Hz and 1 Hz rate, respectively. The pitch angle accuracy is improved by 13% for all four cases. The heading has respectively 23% and 14% less error when using 10 Hz and 1 Hz update rate in the case of known geometry. However, lower improvement can be noted when the boresight angles are estimated at either 10 Hz (0.6%) or 1 Hz (8%). In the case of the smoothed solution (right panel), roll, pitch and heading accuracy is improved by 60%, 62% and 65%, respectively, when using 10 Hz update rate with known boresight angles. At 1 Hz, these values decrease to 46% and 44% respectively for roll and pitch. The heading accuracy is 2% worse.

In summary, increasing the relative geometry update rate improves the navigation performance, especially in the case were the boresight and leverarms of DIMU are already known.

9.6 Overall Navigation Performance Comparison

9.6.1 General Accuracy Improvement

The navigation performance evaluated with respect to the reference solution will be denoted as the state accuracy (RMSE). Only the results for the *campus* datasets are presented, because similar conclusions can be drawn from all other acquired datasets. In the upcoming figures, the following legends will be used:

- *Single IMU (MTx #i)*: navigation solution provided by the INS/GNSS EKF based on the classical single-IMU mechanization of the ith *MTx* unit, with $i = 1, \ldots, R$;
- *Average of Single IMU*: average of the RMSE values obtained with the R single-IMU based EKF;

- *Federated Filter (no reset)*: results obtained with a federated filter without resets. This filter fuses R solutions of single-IMU based filters in least-squares sense. We refer to Bancroft [2010] who investigated these types of filters for INS/GNSS navigation;
- *Geometrically-Constrained (known calibration)*: results given by the geometrically-constrained EKF that is updated with inter-IMU geometry pseudo-measurements based on known (accurate) boresight angles and leverarms;
- *Geometrically-Constrained (unknown calibration)*: results given by the geometrically-constrained EKF that is updated with inter-IMU geometry pseudo-measurements based on estimated (in the state) boresight angles and leverarms;
- *Unweighted Synthetic IMU*: navigation solution computed with the unweighted synthetic IMU and EKF;
- *ARMA-GARCH Synthetic IMU*: navigation solution computed with the ARMA-GARCH synthetic IMU and EKF;
- *MRS Synthetic IMU*: navigation solution computed with the MRS synthetic IMU and EKF;
- *Extended Mechanization*: solution provided by the generalized EKF based on the extended mechanization.

All filters used for computing the navigation solutions presented in this section were updated by carrier-phase differential GPS position and velocity measurements (at 1 Hz). Furthermore, they were aligned by estimated coarse tilt angles (i.e. roll and pitch) during the static leveling procedure (see Section 2.5.2) and assuming large uncertainty in the initial heading while running the EKF. For this aim, the quaternion-based large-angle misalignment algorithm explained in Section 2.6.1 was used for platform fine alignment. Note also that the automatic "inertial-based" non-moving period detection algorithm developed in Section 2.5.1 was activated. By this means, periods during which the filters had to be (re-)initialized (position initialization by means of GPS, velocity assumed to be zero, and coarse alignment procedure engaged) or put in zero-velocity update mode were computed automatically without any interaction from the user.

The performance of the forward EKF attitude solution is presented in the upper panel of Figure 9.12. The different synthetic IMU computers and the generalized EKF have similiar performance improvement: the roll and heading angles accuracy are improved by 90% and 27% with respect to the averaged single IMU performance, respectively, while the pitch angles are 18% less accurate. Both forms of the geometrically-constrained EKF bring nearly 76% and 13% of improvement on roll and pitch angles. Heading is improved by 23% when boresight angles are known, while no difference is observed in the case when the boresight angles are estimated. The federated filter presents the worst results among the filters acounting for redundancy. Regarding position and velocity accuracy (lower panel in Figure 9.12), the geometrically-constrained EKF with unknown calibration parameters performs best in this case (respectively 48% and 51% of improvement for position and velocity) while again, the federated filter brings the lowest advantage with nearly 28% of improvement for both, position and velocity. Comparing the smoother solutions for the same dataset reveals that the highest attitude accuracy improvement

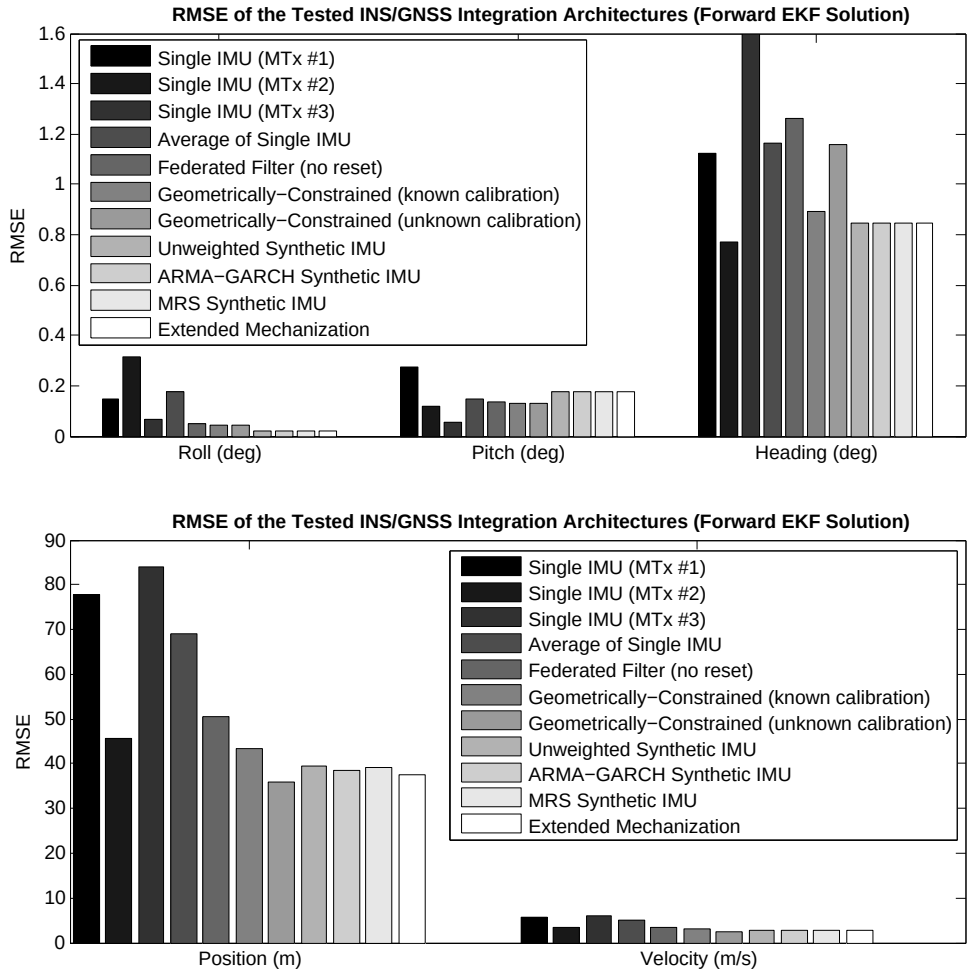

Figure 9.12: Performance of the forward EKF filter running the different redundancy architectures. The upper and lower panels show respectively the attitude and position/velocity RMSE. Note that the large position errors are due to periods (of up to 40 seconds duration) during which the system was in inertial coasting mode.

is brought by the generalized EKF (extended mechanization with known boresight angles) and the geometrically-constrained EKF with known boresight angles (upper panel in Figure 9.13) with around 60% of improvement for all three angles. All forms of synthetic IMUs have similar performance with nearly 55% of improvement for roll and heading, and 63% for pitch. Position and velocity accuracy improvement is shown in the lower panel of Figure 9.13. The generalized EKF improves the position and velocity accuracy by 73% and 40% respectively. The geometrically-constrained filter operated with unknown calibration parameters presents the least improvement in position accuracy.

Globally, the use of redundant sensors brought significant improvements ($\sim \sqrt{R}$) in the navigation solution. Introducing this redundancy information into the filter architecture or through the synthetic IMU computers clearly enables achieving higher accuracy compared to the federated filter. However for the analysed datasets, the gain obtained with the ARMA-

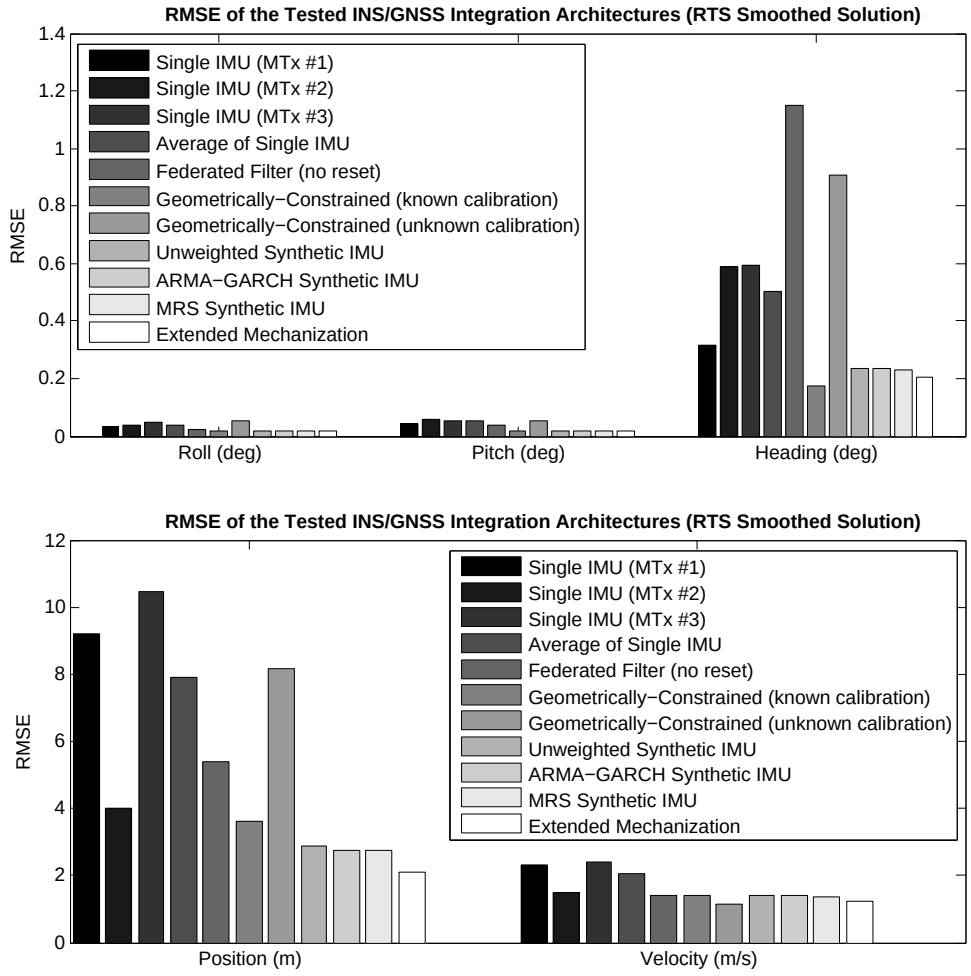

Figure 9.13: Performance of the smoother running the different redundancy architectures. The upper and lower panels show respectively the attitude and position/velocity RMSE. Note that the large position errors are due to periods (of up to 40 seconds duration) during which the system was in inertial coasting mode.

GARCH or MRS based synthetic IMU compared to the unweighted form was marginal with an improvement of 1-3% in accuracy. In the sequel, closer views on critical parts of the *campus* and *urban* datasets are provided.

9.6.2 Close Views on Inertial Coasting

During the *urban* run, the car came twice across tunnels in which the filter operated in inertial coasting mode. The first tunnel lasted approximately 40 seconds and is depicted in the upper left panel of Figure 9.14. The second tunnel (upper right panel) was located in a dense urban area and was shorter (the vehicle came through in approximately 15 seconds). The planimetric and altimetric position drift due to the integrated inertial sensor errors is only presented for the first tunnel in the lower two panels of Figure 9.14. The performance obtained by computing

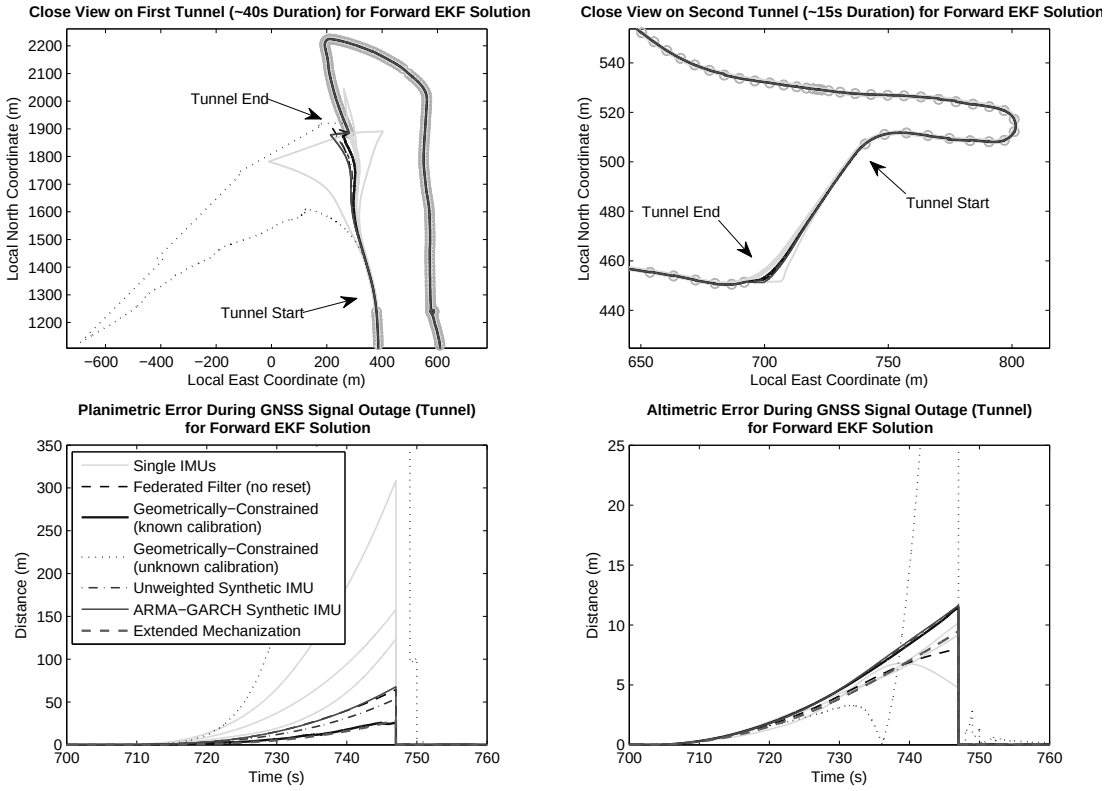

Figure 9.14: Close views on *urban* trajectory parts during which the vehicule came across tunnels. The upper panels present planimetric navigation solutions obtained during inertial coasting mode over a 40 seconds (left panel) and a 15 seconds (right panel) long period. The lower panels show the planimetric (left panel) and altimetric (right panel) errors during the GNSS-signal outage of 40 seconds duration in the tunnel.

each single IMU individually (in this case three *MTx* devices) is represented by the light gray curves. The final planimetric error at the end of the tunnel varies a lot among these three solutions: more than 300 meters error for the worst and nearly 120 meters for the best case. Note that the car came across the tunnel approximately 10 minutes after start. Therefore, some states accounting for sensor biases might not have been observed correctly at this stage. The remaining curves in the panels are issued from the filters accounting for redundancy. The worst performance is clearly accomplished by the geometrically-constrained EKF filter in which the boresight angles had to be estimated (dotted black curve). Due to the decreasing estimated filter precision and the accumulated errors in each individual filter block, errors in the estimated inter-IMU boresight angles increase and impact largely the final position accuracy. This confirms the previously mentioned remark stating that inter-IMU calibration shall only be performed in optimal conditions when these states are observable. The federated filter performs similarly to the ARMA-GARCH and the MRS-based synthetic IMU (note that the MRS performs almost exactly the same and is not depicted for clarity reasons) with a final planimetric error of around 60 meters. The unweighted synthetic IMU solution presents less drift than the two other synthetic IMU forms (around 50 meters). Although better in this case, several (not presented) analyses in which artifical GNSS gaps were introduced revealed that

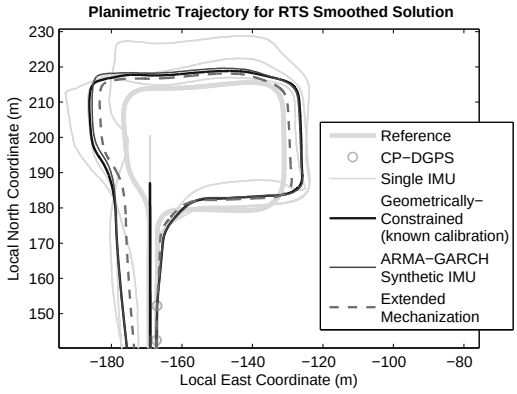

Figure 9.15: Close view on *campus* trajectory parts during which the vehicle came across the underground parking garage.

globally, the unweighted synthetic IMU performed similarly to the two other more sophisticated forms. However, their accuracy dramatically improved compared to the classical single IMU solutions. The minimum position drift is obtained with the geometrically-constrained EKF filter with known calibration parameters and the generalized EKF running extended mechanization with only 20 meters planimetric error at the end of the tunnel. In this case, redundancy at state level performed best.

During the *campus* run, the car experienced a period without GNSS data while passing through an underground parking garage. A closer view on this trajectory portion is drawn in Figure 9.15. This scenario is demanding on the performance of the inertial navigation system since many sharp turns and velocity changes occured during this GNSS-free 25 seconds long period. The planimetric positions issued from smoothed solutions are drawn on this figure with the same coding as was used in Figure 9.14. The same conclusion as for the *urban* dataset can be drawn in this case: the geometrically-constrained EKF together with the generalized EKF performed best while the different forms of synthetic IMU computers had no difference between them. Again, the solution provided by the geometrically-constrained EKF operated with unknown boresight angles dramatically deviates from the reference trajectory (not shown in the figure).

Finally, we analyse for the *urban* run the navigation performance with respect to the number of IMUs increasing from two units up to five. For this purpose, we used IMUs from the second DIMU system. Remember that some *MTx* IMUs within this system were not calibrated and that the *MTx* #4 experienced large level of noise (see Section 9.4.3). Although we consider here only the results obtained with the generalized EKF (extended mechanization), both panels of Figure 9.16 enable to draw conclusions that are also representative for the other redundancy architectures. We can observe some improvement in the attitude accuracy when using more than two sensors (left panel). However, no significant improvement in the attitude accuracy can be observed when using three, four or five devices. A closer view on the 40-seconds long tunnel is depicted in the right panel. Again, the position solution obtained by integrating each individual IMU are drawn as light gray curves (only three out of five are shown for clarity). Using more than two sensors significantly improves the accuracy of inertial navigation. The solutions obtained with three and four sensors are similar. This can be explained by the high noise level featuring *MTx* #4 which may not bring any gain in the solution acccuracy, compared

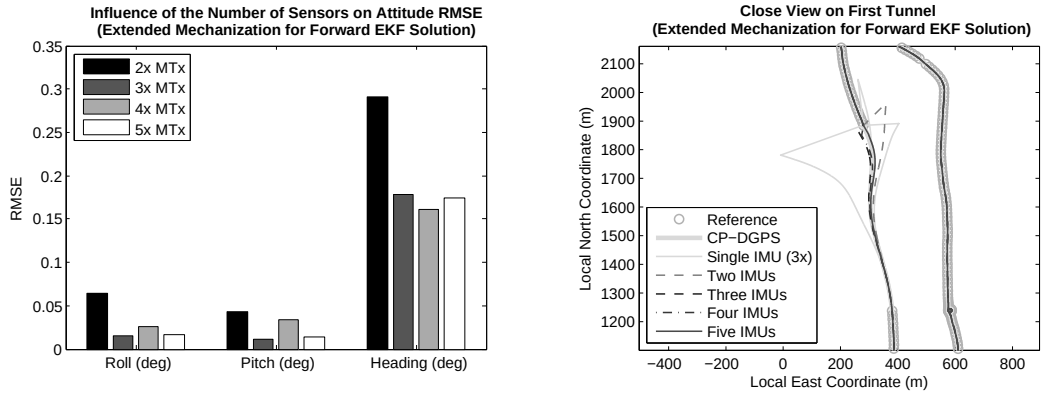

Figure 9.16: Influence of the number of sensors in the forward EKF solution of the *urban* run. The left panel shows the attitude RMSE and the right panel provides a close view on position solutions during the 40 seconds long tunnel.

to the three-IMU based filter. However, the accumulated error is significantly bounded when using five sensors since the obtained trajectory is very close to the reference.

Chapter 10

Conclusions and Recommendations

10.1 Contributions and Conclusions

This research aimed in proposing and investigating novel modeling and estimation approaches for integrated inertial navigation. Particular attention was paied to the development of theoretical aspects and its validation by means of simulation studies and real applications. The developed concepts were described using an algorithmic notation that is suitable for implementation. This section highlights the major findings.

10.1.1 Theoretical/Conceptual Contributions

The following main theoretical contributions and conclusions can be drawn from this research.

Adapted EM algorithm for inertial sensor error modeling. We described a modified likeli-hood-based estimation technique based on the well known EM algorithm. The modifications aimed at estimating error models traditionally used within the navigation community for which conventional techniques such as the autocorrelation, the PSD or also the Allan variance methods fail. In particular, the often encountered problem of selecting the inverse correlation time and the variance of driving noise for a first-order Gauss-Markov process employed with INS/GNSS integration has been treated. We analysed the algorithm results with a critical and practical point of view on the basis of simulations with typical error signals. These simulations showed that the EM algorithm performs better than the Allan variance and offers the possibility to estimate first-order Gauss-Markov processes mixed with other types of noises. An empirical scenario was presented to support the former findings. There, the positive effect of using the more sophisticated EM-based error modeling on a filtered trajectory was demonstrated. At the same time, the conducted tests revealed limits of this approach that are related to convergence and stability issues. Suggestions were given to circumvent or mitigate these problems when the complexity of the error structure is "reasonable". The work also highlighted the fact that, similarly to the Allan variance, this approach may not be able to estimate reasonably well the parameters of more complex composite stochastic processes.

A new estimation framework for composite stochastic processes. We developed a new estimation method for the parameters of composite Gaussian processes, called the generalized method of wavelet moments. In this thesis, we considered processes which are mostly used in navigation. The proposed estimator results as the optimization of a criterion based on standardized distance between the sample and the model-based wavelet variances. Indeed, wavelet variances provide a decomposition of the process variance through different scales and hence contain information about different features of the stochastic model. We investigated the asymptotic properties of the estimator by making adequate links to papers (written in the framework of this thesis) which are dedicated to purely theoretical aspects. By means of simulation studies, we demonstrated the capability of the GMWM approach to estimate parameters of very complex models such as a sum of first-order Gauss-Markov processes for which other estimation methods fail. This method opened the door to original investigations of more complex models that capture the behavior of inertial sensor errors. The GMWM estimator was hence applied on signals issued from low-cost MEMS-based inertial measurement units, using a sum of Gauss-Markov processes as stochastic models. The benefits of using such models was highlighted by analysing the quality of the free-inertial trajectory during the (artificial) absence of GNSS data. During these epochs, the inertial navigation operated in coasting mode while GNSS-supported trajectory acted as a reference. As the overall performance of inertial navigation is strongly dependent on the errors corrupting its observations, the benefits of using the more appropriate error models (with respect to simpler ones obtained via classical Allan variance identification technique) were demonstrated by a significant improvement in the trajectory accuracy.

A study of sensor error behavior in dynamical conditions. An experimental setup was developed and implemented with the aim of constructing inertial sensor error signals under dynamic conditions. We realized and implemented data acquisition with a hardware-based real-time synchronization scheme capable to perform relative time alignment between different operating IMUs and absolute timing with respect to GPS time-frame. Moreover, algorithms for estimating the inter-IMU spatial alignment in an EKF were proposed and validated through simulation. This enabled to construct error signals of MEMS-based IMUs with respect to a navigation-grade system used as a reference. This method allowed to empirically prove the dependency of the sensor noise level on dynamics.

Inertial redundancy at observation level. The use of multiple inertial sensors operating on the same platform provides a way of dynamically estimating the noise level, to improve navigation accuracy and to detect defective sensors. In this work, we studied new algorithms accounting for redundancy at observation level, the so-called synthetic IMU computers. Three types of computers were analysed: the unweighted, ARMA-GARCH, and MRS based computers. The first computes synthetic sensors by considering equal weights for each individual device, while the two others account for varying noise levels with respect to time and among individual sensors. Experiments with real data showed that all three computers significantly reduced the level of sensor noise and improved navigation accuracy with respect to single-IMU navigation filters (30% of improvement in heading accuracy, 45% in velocity accuracy, 42% in position accuracy). However, the

advantage of using ARMA-GARCH or MRS based computers was only marginal in the experimental dataset at hand.

Inertial redundancy at state level. A geometrically-constrained EKF was developed with the possibility to deal with unknown inter-IMU geometry (i.e. misalignment and lever-arms), and its performance was investigated with respect to MEMS-based inertial sensors. The filter runs in parallel several single-IMU based filters which are regularly updated by geometrical constraints. We developed and implemented the related algorithms for this purpose. The performance of this filter was analysed in real-case experiments and confronted to results obtained with other state-based redundancy schemes such as the generalized EKF running the extended mechanization, as well as the developed synthetic IMU computers. Compared to the latters, overall performance analyses revealed the superior performance of methods accounting for redundancy at state level with accuracy gains of 27% in attitude, 50% in velocity and 48% in position. Note that the results with respect to the generalized EKF and the synthetic IMU method were comparable only in a case in which both approaches could benefit from the known inter-IMU geometry (e.g. previously estimated by the geometrically-constrained EKF).

10.1.2 Engineering Contributions

Several software modules were developed in the framework of this thesis. The most important ones among them are described in the sequel.

Navigation data processing software. A loosely-coupled extended Kalman filter/s-moother was developed and integrated with a redundant INS computer. The filter accommodates different observations as absolute position and velocity measurements, zero-velocity measurements, as well as magnetic observations, non-holonomic constraints or odometer speed measurements. A completely automated processing scheme was designed in which static periods are accurately detected and processing strategies set to define the filter behavior such as filter reset (static initialization) or in-motion coarse and fine alignment (de)activation. Moreover, a prediction-correction based inertial integrator was implemented that enables proper handling of data streams which are not aligned in time (which is the case in integrated navigation) as well as processing of irregularly sampled inertial data. Finally, the filter supports large uncertainty in the knowledge of initial platform tilt and heading by relying on INS error models accounting for such conditions. The switching between the large-tilt/large-heading error model, the small-tilt/large-heading error model, and the fine alignment model is completely automated.

Inertial sensor calibration software. A C++ based software implementing the newly proposed GMWM estimation framework was developed. The software accepts several industrial input formats (e.g. *Applanix*, *XSens*, *ASCII*) and enables fast and interactive building of composite stochastic models. The most popular error models like Gaussian white noise, quantization noise, first-order Gauss-Markov processes, random walk, random rate ramp, and more generally AR(1) processes are available. The algorithms were implemented in an efficient way such that long signals are supported.

Distributed IMU datalogger. A real-time C++ based DIMU datalogger based on *XSens MTi-G* and *MTx* units has been developed. The software enables accurate absolute and relative time synchronization by means of hardware-based synchronization of all IMUs with GPS time. The software module was made available to international research partners who adapted its functionality for DIMU acquisition on a UAV-helicopter.

10.2 Recommendations

During the development of this work, new research challenges have emerged that would make further investigations worthwile.

Investigations on composite stochastic processes for inertial sensors. The GMWM estimator developed in the framework of this research opens the door to the estimation of complex error models which are impossible to treat with the conventional methods. Although more complex error models can be explored for modeling stochastic errors in gyroscopes or accelerometers, this approach is general and has wide applications within or outside the navigation community (e.g. oscillators, atomic clocks, etc.).

Rigorous inertial sensor error model selection. Judging wether a stochastic error model is more accurate than another one is a nontrivial task when performed at the level of a navigation filter. Most of the time, the quality of a sensor error model, i.e. the augmented states in the EKF, is evaluated in inertial coasting mode. However, the error bounding during GNSS-free periods depends on the past information acquired by the filter. This is committed to the history of the encountered dynamics, the quality of the observations, as well as to the particular realization of stochastic processes. Overall, this makes the decision process tricky when claiming that a filter design is *in general* better than another design. In this respect, rigorous approaches for model selection/evaluation need to be developed.

Combining redundancy at observation and state level. In this thesis, the algorithmic approaches for inertial redundancy were clearly discriminated between observation levels and state levels. New algorithms combining inertial redundancy at observation *and* at state levels might bring even more substantial improvement to integrated inertial navigation with redundant systems.

Bibliography

H. Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

D. Allan. Statistics of Atomic Frequency Standards. In *Proceedings of the IEEE*, volume 54, pages 221–230, 1966.

D.J. Allerton and H. Jia. A Review of Multisensor Fusion Methodologies for Aircraft Navigation Systems. *Journal of Navigation*, 58(3):405–417, 2005.

S.L. Altmann. *Rotation, Quaternions, and Double Groups.* Clarendon Press, Oxford, 1986.

Anonymous. World Magnetic Model of the United States National Geospatial-Intelligency Agency. Technical report, United States National Geospatial-Intelligency Agency. URL `http://www.ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml`.

J.B. Bancroft. Multiple IMU Integration for Vehicular Navigation. In *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, September 2009.

J.B. Bancroft. *Multiple Inertial Measurement Unit Integration for Pedestrian Navigation.* PhD thesis, Department of Geomatics Engineering, The University of Calgary, Calgary, AB, Canada, 2010.

R.A. Baugh. Frequency Modulation Analysis with the Hadamard Variance. In *25th Annual Symposium on Frequency Control. 1971*, pages 222–225. IEEE, 1971.

J. Benichou and M.H. Gail. A Delta Method for Implicitly Defined Random Variables. *The American Statistician*, 43:41–44, 1989.

D.O. Benson. A Comparison of Two Approaches to Pure-inertial and Doppler-inertial Error Analysis. *IEEE Transactions on Aerospace and Electronic Systems*, (4):447–455, 1975.

T. Bollerslev. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time Series Analysis: Forecasting and Control*, volume 734. Wiley, 2011.

R.P. Brent. *Algorithms for Minimization without Derivatives.* Dover Publications, 2002.

R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*, volume 3. John Wiley, New York, second edition, 1997.

A.E. Bryson. *Applied Linear Optimal Control: Examples and Algorithms*. Cambridge University Press, 2002.

P.E. Caines. *Linear Stochastic Systems*. John Wiley & Sons, Inc. New York, NY, USA, 1987.

G. Casella and R.L. Berger. *Statistical Inference*. Duxbury Press, 2002. ISBN 0534243126.

I. Colomina, M. Giménez, J.J. Rosales, and M. Wis. What Can Skew-Redundant IMU Configurations Contribute to Photogrammetry? In *Proceedings of ISPRS Workshop*, pages 22–23. Working Group 1.5, September 2003.

I. Colomina, M. Giménez, J.J. Rosales, M. Wis, A. Gómez, and P. Miguelsanz. Redundant IMUs for Precise Trajectory Determination. In *XXth ISPRS Congress Proceedings*, Istanbul, Turkey, 2004a.

I. Colomina, M. Giménez, and M. Wis. On the Use of Redundant Inertial Data for Geodetic Applications. In *Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2004)*, pages 801–808, Long Beach, CA, USA, September 2004b. ION.

NIMA WGS84 Update Committee. Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems - 3rd Edition. Technical report, National Imagery and Mapping Agency, 1997.

I. Daubechies. *Ten Lectures on Wavelets*, volume 61. SIAM, 1992.

A.P. Dempster, N.M. Laird, and D.B. Rubing. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

K.E. Drexler. *Nanosystems- Molecular Machinery, Manufacturing, and Computation*. John Wiley & Sons, New York, 1992.

J. Durbin and S.J. Koopman. *Time Series Analysis by State Space Methods*, volume 38 of *24*. Oxford University Press, 2012.

G. Einicke, G. Falco, J. Malos, D. Reid, and D. Hainsworth. Parameter Estimation for Mine Navigation. In *IGNSS Symposium 2009*, Australia, December 2009.

N. El-Sheimy, H. Hou, and X. Niu. Analysis and Modeling of Inertial Sensors using Allan Variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149, January 2008. ISSN 0018-9456.

J. Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill New York, NY, USA:, 2008.

W.T. Fong, S.K. Ong, and A.Y.C. Nee. Methods for In-field User Calibration of an Inertial Measurement Unit without External Equipment. *Measurement Science and Technology*, 19: 085202, 2008. doi: 10.1088/0957-0233/19/8/085202.

P.H. Franses and D. Van Dijk. *Nonlinear Time Series Models in Empirical Finance*. Cambridge University Press, 2000.

A.R. Gallant and G. Tauchen. Which Moments to Match? *Econometric Theory*, 12(4):657–681, October 1996. ISSN 0266-4666.

A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.

M.G. Genton and E. Ronchetti. Robust Indirect Inference. *Journal of the American Statistical Association*, 98(461):67–76, March 2003. ISSN 0162-1459.

T. Giannakopoulos. *Study and Application of Acoustic Information for the Detection of Harmful Content, and Fusion with Visual Information*. PhD thesis, National and Kapodistrian University of Athens, Greece, July 2009.

Z. Gosiewski and A. Ortyl. Strapdown Inertial Navigation System. Part 2 - Error Models. *Journal of Theoretical and Applied Mechanics*, (4):937–962, 1998.

C. Gourieroux, A. Monfort, and E. Renault. Indirect Inference. *Journal of Applied Econometrics*, 8:85–118, May 1993. URL `http://ideas.repec.org/a/jae/japmet/v8y1993isps85-118.html`.

C.W.J. Granger and M.J. Moris. Time Series Modelling and Interpretation. *Journal of the Royal Statistical Society, series A*, 139:246–257, 1976.

C.A. Greenhall. Recipes for Degrees of Freedom of Frequency Stability Estimators. In *IEEE Transactions on Instrumentation and Measurements*, volume 40, pages 994–999, 1991.

C.A. Greenhall. Spectral Ambiguity of Allan Variance. *IEEE Transactions on Instrumentation and Measurement*, 47(3):623–627, June 1998.

C.A. Greenhall, D.A. Howe, and D.B. Percival. Total Variance, an Estimator of Long-Term Frequency Stability. In *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, volume 46, pages 1183–1191, September 1999.

S. Guerrier. Improving Accuracy with Multiple Sensors: Study of Redundant MEMS-IMU/GPS Configurations. In *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, pages 3114–3121, Savannnah, GA, USA, September 22-25 2009.

S. Guerrier, Y. Stebler, J. Skaloud, and M.P. Victoria-Feser. Comparison and Optimal Tuning of Wavelet Variance based Estimators of Composite Stochastic Processes. *(Under preparation)*.

S. Guerrier, A. Waegli, J. Skaloud, and M.P. Victoria-Feser. Fault Detection and Isolation in Multiple MEMS-IMUs Configurations. *IEEE Transactions on Aerospace and Electronic Systems*, 48:2015–2031, 2012. ISSN 0018-9251. doi: 10.1109/TAES.2012.6237576.

S. Guerrier, Y. Stebler, J. Skaloud, and M.P. Victoria-Feser. Wavelet-Variance-based Estimation for Composite Stochastic Processes. *Journal of the American Statistical Association (To appear)*, 2013.

N. Gupta and R. Mehra. Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations. *IEEE Transactions on Automatic Control*, 1974. ISSN 0018-9286.

J.D. Hamilton. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica: Journal of the Econometric Society*, pages 357–384, 1989.

J.D. Hamilton. *Time Series Analysis*, volume 10. Cambridge University Press, 1994.

S. Han and J. Wang. Quantization and Colored Noises Error Modeling for Inertial Sensors for GPS/INS Integration. *Sensors Journal, IEEE*, (99):1–1.

L.P. Hansen. Large Sample Properties of Generalized Method of Moments Estimators. *Econometrica: Journal of the Econometric Society*, 50(4):1029–1054, July 1982. ISSN 0012-9682.

A.C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, Cambridge, UK, 1991. ISBN 0521405734.

B. Hofmann-Wellenhof, K. Legat, and M. Wieser. *Navigation - Principles of Positioning and Guidance*. Springer, Wien, 5th edition, 2003.

E.E. Holmes. Derivation of the EM Algorithm for Constrained and Unconstrained Multivariate Autoregressive State-Space (MARSS) Models. Technical report, Northwest Fisheries Science Center, NOAA Fisheries 2725 Montlake Blvd E., Seattle, WA 98112, December 2010.

H. Hou. Modeling Inertial Sensors Errors using Allan Variance. Master's thesis, Geomatics Engineering, University of Calgary, AB, Canada, 2004.

D.A. Howe. An Extension of the Allan Variance with Increased Confidence at Long Term. In *Proceedings of the 1995 IEEE International Frequency Control Symposium*, pages 321–329, 1995.

D.A. Howe, R.L. Beard, C.A. Greenhall, F. Vernotte, W.J. Riley, and T.K. Peppler. Enhancements to GPS Operations and Clock Evaluations using a Total Hadamard Deviation. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 52(8):1253–1261, 2005.

D. Huang, H. Leung, and N. El-Sheimy. Expectation Maximization based GPS/INS Integration for Land-vehicle Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 43 (3):1168–1177, 2007. ISSN 0018-9251.

iMAR GmbH. iIMU-FSAS [-E] Technical Documentation, March 2009. URL `http://www.imar-navigation.de/`.

C. Jekeli. *Inertial Navigation Systems with Geodetic Applications*. Walter de Gruyter, 2000. doi: 3110159031.

C.J. Kim. Dynamic Linear Models with Markov-Switching. *Journal of Econometrics*, 60(1-2): 1–22, 1994.

C. Kittel. *Elementary Statistical Physics*. John Wiley & Sons, 1958.

X. Kong. *Inertial Navigation System Algorithms for Low Cost IMU*. PhD thesis, Department of Mechanical and Mechatronic Engineering, Graduate School of Engineering, University of Sydney, Australia, 2000.

X. Kong, E.M. Nebot, and H. Durrant-Whyte. Development of a Nonlinear Psi-angle Model for Large Misalignment Errors and its Application in INS Alignment and Calibration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1430–1435. IEEE, 1999.

Litton Guidance and Control Systems Division. Product Description of the LN-200 Family. Technical Report Document No. 20896I, Litton Systems, Inc., 1996.

S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

M.A. Martínez-Beneito, D. Conesa, A. López-Quílez, and A. López-Maside. Bayesian Markov Switching Models for the Early Detection of Influenza Epidemics. *Statistics in Medicine*, 27 (22):4455–4468, 2008.

R.E. McCulloch and R.S. Tsay. Bayesian Inference and Prediction for Mean and Variance Shifts in Autoregressive Time Series. *Journal of the American Statistical Association*, pages 968–978, 1993.

G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 1997.

S. Nassar. *Improving the Inertial Navigation System (INS) Error Model for INS and INS/DGPS Applications.* PhD thesis, The University of Calgary, AB, Canada, November 2003.

S. Nassar, K.P. Schwarz, and N. El-Sheimy. Modeling Inertial Sensor Errors Using Autoregressive (AR) Models. *Journal of the Institute of Navigation*, 51(4):259–268, 2004.

A.T. Nelson. *Nonlinear Estimation and Modeling of Noisy Time-series by Dual Kalman Filtering Methods.* PhD thesis, Oregon Graduate Institute of Science and Technology, OR, USA, September 2000.

M. Park and Y. Gao. Error Analysis and Stochastic Modeling of Low-cost MEMS Accelerometer. *Journal of Intelligent and Robotic Systems*, 46(1):27–41, 2006.

A.J. Pejsa. Optimum Skewed Redundant Inertial Navigators. *AIAA Journal*, 12:899–902, 1974.

D.B. Percival. Total Variance as an Exact Analysis of the Sample Variance. In *Proceedings of the 29th Annual Precise Time and Time Interval (PTTI) Systems and Applications Meeting*, pages 97–105, Long Beach, CA, USA, 1997.

D.B. Percival and P. Guttorp. Long-Memory Processes, the Allan Variance and Wavelets. *Wavelets in Geophysics*, 4:325–344, 1994.

D.B. Percival and A.T. Walden. *Wavelet Methods for Time Series Analysis.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, USA, 2000.

J. Perry and J. Childs. Timing On the Fly: Synchronization for Direct Georeferencing on Small UAVs. *Inside GNSS*, 4(6):34–40, November/December 2009.

C.R. Rao. *Linear Statistical Inference and Its Applications (2nd ed.).* John Wiley, New York, 1973.

J.J. Rosales and I. Colomina. A Flexible Approach for the Numerical Solution of the INS Mechanization Equations. In *Proceedings of the 6th Geomatic week*, 2005.

P. Savage. Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms. *Journal of Guidance, Control, and Dynamics*, 21(2):208–221, March-April 1998a.

P.G. Savage. Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms. *Journal of Guidance Control and Dynamics*, 21(1):19–28, January-February 1998b. ISSN 0731-5090.

P. Schaer, J. Skaloud, Y. Stebler, P. Tomé, and R. Stengele. Airborne LiDAR In-flight Accuracy Estimation. *GPS World*, 20(8):37–41, 2009.

B.M. Scherzinger. Inertial Navigator Error Models for Large Heading Uncertainty. *Position Location and Navigation Symposium, 1996 IEEE/ION*, page 477–484, 1996.

G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

A. Serroukh, A.T. Walden, and D.B. Percival. Statistical Properties and Uses of the Wavelet Variance Estimator for the Scale Analysis of Time Series. *Journal of the American Statistical Association*, 95(449):184–196, 2000.

J. Shao. *Mathematical Statistics*. Springer Texts in Statistics. Springer, 2003. ISBN 9780387953823. URL `http://books.google.ch/books?id=cyqTPotl7QcC`.

E. Shin. *Estimation Techniques for Low-cost Inertial Navigation*. PhD thesis, The University of Calgary, AB, Canada, May 2005.

E.H. Shin and N. El-Sheimy. Accuracy Improvement of Low-cost INS/GPS for Land Applications. Master's thesis, University of Calgary, Department of Geomatics Engineering, AB, Canada, December 2001.

R.H. Shumway and D.S. Stoffer. An Approach to Time Series Smoothing and Forecasting using the EM Algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

R.H. Shumway and D.S. Stoffer. Dynamic Linear Models with Switching. *Journal of the American Statistical Association*, pages 763–769, 1991.

R.H. Shumway and D.S. Stoffer. *Time Series Analysis and its Applications*. Springer Verlag, 2000. ISBN 0387989501.

J. Skaloud, P. Schaer, Y. Stebler, and P. Tomé. Real-time Registration of Airborne Laser Data with Sub-decimeter Accuracy. *ISPRS Journal of Photogrammetry and Remote Sensing*, pages 208–217, March 2009. doi: 10.1016/j.isprsjprs.2009.12.003.

A. Smith, P.A. Naik, and C.L. Tsai. Markov-Switching Model Selection using Kullback-Leibler Divergence. *Journal of Econometrics*, 134(2):553–577, 2006.

A.A. Smith. Estimating Nonlinear Time-series Models using Simulated Vector Autoregressions. *Journal of Applied Econometrics*, 8(S1):S63–S84, 1993.

IEEE Std 1293-1998. IEEE Standard Specification Format Guide and Test Procedure for Linear, Single-Axis, Non-gyroscopic Accelerometers. Technical report, IEEE, Piscataway, NJ, USA, 1998.

IEEE Std 952-1997. IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros. Technical report, IEEE, 1998.

Y. Stebler, S. Guerrier, J. Skaloud, and M.P. Victoria-Feser. Constrained Expectation-Maximization Algorithm for Stochastic Inertial Error Modeling: Study of Feasibility. *Measurement Science and Technology*, 22:085204, August 2011a. doi: 10.1088/0957-0233/22/8/085204.

Y. Stebler, S. Guerrier, J. Skaloud, and M.P. Victoria-Feser. Improving Modeling of MEMS-IMUs Operating in GNSS-denied Conditions. In *Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2011)*, September 2011b.

Y. Stebler, S. Guerrier, J. Skaloud, and M.P. Victoria-Feser. A Framework for Inertial Sensor Calibration using Complex Stochastic Error Models. In *Position, Location and Navigation Symposium, 2012 IEEE/ION, Session A5*, 2012.

J.M. Strus, M. Kirkpatrick, and J.W. Sinko. Development of a High Accuracy Pointing System for Maneuvering Platform. In *Proceedings of the 20th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2007)*, Fort Worth, TX, USA, September 2007.

M.A. Sturza. Navigation System Integrity Monitoring using Redundant Measurements. *Navigation*, 35(4):1988–89, 1988a.

M.A. Sturza. Skewed Axis Inertial Sensor Geometry for Optimal Performance. In *8th AIAA/IEEE Digital Avionics Systems Conference*, volume 1, pages 128–135, 1988b.

T. Terasvirta. The Invertibility of Sums of Discrete MA and ARMA Processes. *Scandinavian Journal of Statistics*, 4:165–170, 1977.

The Institute of Electrical and Electronics Engineers Inc. IEEE 1554 Recommended Practice for Inertial Sensor Test Equipment, Instrumentation, Data Acquisition and Analysis. Technical report, 2005.

S.T. Thornton and J.B. Marion. *Classical Dynamics of Particles and Systems*. Brooks/Cole, 2004. ISBN 0534408966.

D.H. Titterton and J.L. Weston. *Strapdown Inertial Navigation Technology*. Peter Peregrinus Ltd, 1997.

P. Tomé. *Integration of Inertial and Satellite Navigation Systems for Aircraft Attitude Determination*. Phd thesis, University of Oporto, Portugal, January 2002.

P. Tomé and O. Yalak. Improvement of Orientation Estimation in Pedestrian Navigation by Compensation of Magnetic Disturbances. *Journal of the Institute of Navigation*, 55(3): 179–190, 2008. URL `http://www.ion.org/`.

H. Tong and K.S. Lim. Threshold Autoregression, Limit Cycles and Cyclical Data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 245–292, 1980.

A. Waegli and J. Skaloud. Assessment of GPS/MEMS-IMU Integration Performance in Ski Racing. In *Proceedings of ENC-GNSS 2007 (TimeNav'07), Geneva, Switzerland*, 2007.

A. Waegli, S. Guerrier, and J. Skaloud. Redundant MEMS-IMU Integrated with GPS for Performance Assessment in Sports. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1260–1268, 2008.

A. Waegli, J. Skaloud, S. Guerrier, M.E. Parés, and I. Colomina. Noise Reduction and Estimation in Multiple Micro-electro-mechanical Inertial Systems. *Measurement Science and Technology*, 21:065201 (11 p.), 2010. ISSN 0957-0233. doi: 10.1088/0957-0233/21/6/065201.

Adrian Waegli. *Trajectory Determination and Analysis in Sports by Satellite and Inertial Navigation*. PhD thesis, Lausanne, Switzerland, 2009. URL `http://library.epfl.ch/theses/?nr=4288,http://library.epfl.ch/theses/?nr=4288`.

G. Wahba. A Least Squares Estimate of Spacecraft Attitude. *SIAM Review*, 7(3):409, 1965.

M. Wis and I. Colomina. Dynamic Dependency of Inertial Sensor Errors and its Application to INS/GNSS Navigation. In *Proceedings of the NAVITEC 2010 Congress*, Noordvijk, Netherlands, 8-10 December 2010.

L.S.Y. Wu, J.S. Pai, and J.R.M. Hosking. An Algorithm for Estimating Parameters of State-space Models. *Statistics & Probability Letters*, 28(2):99–106, 1996. ISSN 0167-7152.

Z. Xing. *Over-bounding Integrated INS/GNSS Output Errors*. PhD thesis, University of Minnesota, USA, October 2010.

Z. Xing and D. Gebre-Egziabher. Modeling and Bounding Low Cost Inertial Sensor Errors. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1122–1132, 2008. doi: 10.1109/PLANS.2008.4569999.

XSens Technologies B.V. MTi-G User Manual and Technical Documentation. Technical report, 2009a.

XSens Technologies B.V. MTi and MTx User Manual and Technical Documentation. Technical report, 2009b.

N.F. Zhang. Allan Variance of Time Series Models for Measurement Data. *Metrologia*, 45(5): 549, 2008.

Index

Index