# 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)

Einhundertunderster Band Volume 101 Vehicle Dynamic Model Based Navigation for Small UAVS

Mehran Khaghani

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 978-3-908440-47-5

Redaktion des 101. Bandes: Dr. M. Khaghani, J. Müller-Gantenbein, Dr. Jan Skaloud, Prof. A. Geiger Druck: Print-Atelier ADAG, Zürich

#### VORWORT

Unbemannte Luftfahrzeuge (UAV) kommen immer mehr zum Einsatz, nicht nur bei Anwendungen in der Geomatik, Bau- oder Umwelttechnik, sondern auch bei anderen wie z.B. visuellen Inspektionen, Überwachung und Paketzustellung. Im Gegensatz zu persönlichen Drohnen benötigen professionelle UAVs redundante Sensoren und eine zuverlässige Steuerung/Navigation, die ihren sicheren Betrieb auch bei variierenden oder unerwarteten Umgebungsbedingungen ermöglichen.

Diese Arbeit konzentriert sich auf die Verbesserung der autonomen Navigation im Freien, welche auch im Zeitalter von (fast) 4 voll funktionsfähigen globalen Satellitennavigationssystemen (GNSS) wie eine Sisyphusarbeit anmutet. In der Tat ist GNSS die grundlegende Technologie für Drohnenoperationen, die auf der Grundlage einer vorgeplanten Trajektorie (Wegpunktnavigation) basiert. Sie erfreut sich zunehmender Beliebtheit, sowohl im Freizeit- als auch im professionellen Bereich. Während die steigende Anzahl der GNSS-Satelliten die Zahl der Signale und Frequenzen erhöhte, blieb die fragile Empfangsgüte eine Achillesferse bei der Garantie der Verfügbarkeit und Kontinuität der Positionierung. Aus diesen Gründen entstanden immer mehr Vorschläge für Backups, die hauptsächlich auf optischen Sensoren basieren. Diese Ansätze haben jedoch mehrere Nachteile wie bei eingeschränkter Sicht und bei schlechter Auflösung der Oberflächentextur. Zudem erhöht die Verwendung von mehreren Sensoren das Fluggewicht und die System-Komplexität.

Aus diesen Gründen - und in Analogie zur bemannten Luftfahrt - bleibt die Navigationsautonomie für die Kontrolle und die Betriebssicherheit der Drohnen von grundlegender Bedeutung. Darüber hinaus wird die Qualität und Zuverlässigkeit der autonomen Navigation wahrscheinlich die zukünftige Genehmigungserteilung für BVLOS-Operationen (beyond visual line of sight) beeinflussen. Die Arbeit von Mehran Khaghani zeigt einen neuen und vielversprechenden Weg, wie die Autonomie ohne zusätzliche Sensoren elegant und effizient sichergestellt werden kann. Die vorliegenden Forschungsergebnisse zeigen, dass eine Verbesserung von bis zu einem Faktor 100 möglich ist, ohne das Sensorsetup zu ändern. Der beschriebene Ansatz widerspricht dem Mainstream der integrierten Navigation auf Basis von kinematischen Modellen für Luftfahrzeuge, wie sie seit Jahrzehnten bekannt sind. Im neuen Ansatz wird das auf Trägheitssensoren beruhende Rückgrat der Navigation durch die Kenntnis der Aerodynamik des Fahrzeugs und der Steuerbefehle des Autopiloten ersetzt.

Die Originalität des vorgeschlagenen Ansatzes wurde zunächst im Zuge mehrerer referenzierter Veröffentlichungen in Fachzeitschriften und an Konferenzen durch Fachgremien, danach zusätzlich durch die Revision des US-Patents, erteilt am 22. Mai 2018 unter der Patentnummer US 9 978 285 evaluiert. Die vorgeschlagene Methode fand auch Zuspruch in der Drohnengemeinschaft, indem sie industrielle und staatliche Mittel für ihre Weiterentwicklung und Implementierung in verschiedenen Anwendungen mobilisierte.

**Prof. Dr. Jan Skaloud** EPFL Lausanne Projektleiter **Prof. Dr. Alain Geiger** ETH Zürich Präsident der SGK

#### PREFACE

L'utilisation de micro-véhicules aériens sans pilote (UAV) se développe non seulement en géomatique, en génie civil ou en ingénierie environnementale, mais également dans de nombreuses autres applications telles que les inspections visuelles, la surveillance et la livraison de colis. Contrairement aux drones personnels, les drones professionnels doivent disposer de capteurs redondants et d'un système de contrôle / navigation fiable pour rendre sûre leur exploitation dans des conditions environnementales variables ou imprévisibles.

Le travail ci-présent se concentre sur l'amélioration de la navigation autonome dans les espaces ouverts, qui semble être un rocher de Sisyphe, même à l'heure où (presque) 4 systèmes mondiaux de navigation par satellite (GNSS) sont pleinement opérationnels. En effet, la technologie GNSS est à la base des opérations de drones utilisant des trajectoires pré-planifiées (navigation par way-points), dont nous voyons l'essor rapide aujourd'hui - à la fois à des fins récréatives et professionnelles. Cependant, alors que le nombre des satellites GNSS augmente le nombre de signaux et de fréquences, la fragilité de leur réception reste un talon d'Achille pour garantir la disponibilité et la continuité de ce moyen de positionnement pour les drones. De plus en plus de méthodes reposant sur la détection optique sont proposées pour remédier à ce problème. Néanmoins, des limitations dans la visibilité, la résolution ou la texture ainsi que le besoin de capteurs supplémentaires qui augmentent la charge utile et la complexité du traitement, restreignent ces méthodes à un usage marginal.

Pour ces raisons - et par analogie avec les aéronefs pilotés par des humains l'autonomie de la navigation demeure primordiale pour le contrôle des drones et la sécurité opérationnelle.

De plus, la qualité et la fiabilité de la navigation autonome sont susceptibles d'influer la délivrance future des autorisations pour les opérations BVLOS (beyond visual line of sight ou au-delà de la ligne de visée). Les travaux de Mehran Khaghani démontrent une manière nouvelle et prometteuse d'assurer cette autonomie de manière élégante et efficace sans ajouter de capteurs supplémentaires à bord. En effet, cette recherche a montré qu'il était possible d'obtenir des améliorations d'un facteur allant jusqu'à 100 sans changer la configuration sensorielle de l'aéronef en question. L'approche décrite va à l'encontre des approches conventionnelles de la navigation intégrée basée sur la modélisation cinématique des véhicules aériens, telle qu'appréhendée depuis des décennies. Elle remplace le cœur du système, basé sur la détection inertielle, par celui issu de la connaissance de l'aérodynamique du véhicule et des commandes du pilote automatique.

L'originalité de l'approche proposée a d'abord été évaluée dans le cadre d'examens par des pairs dans plusieurs publications référencées dans des revues et des conférences ; en second lieu par la révision du brevet américain délivré le 22 mai 2018 sous le numéro de brevet US 9 978 285. La méthode proposée a également suscité l'intérêt dans la communauté des drones en attirant des financements industriels et gouvernementaux pour son développement et sa mise en œuvre pour différentes applications.

**Prof. Dr. Jan Skaloud** EPFL Lausanne Directeur de projet **Prof. Dr. Alain Geiger** ETH Zürich Président de la CGS

#### FOREWORD

The usage of micro unmanned aerial vehicles (UAV) is expanding not only in geomatics, civil, or environmental engineering, but also in many other applications such as visual inspections, surveillance and parcel delivery. In contrary to personal drones, professional UAVs need to have redundant sensors and reliable control/navigation that makes their operation safe in varying or unexpected environmental conditions.

This work focuses on improving autonomous navigation in open spaces, which in the era of (almost) 4 fully operational global navigation satellite systems (GNSS) seems like an Sisyphean task. Indeed, GNSS is the enabling technology for drone operations based on pre-planned trajectories (way-point navigation), boom of which we witness today – both, in recreational and professional sectors. Yet, while the abundance of GNSS satellites increased the number of signals and frequencies, the fragility of their reception remained an Achilles' heel for "guaranteeing" availability and continuity of this mean of positioning for UAVs. For these reasons there is an upswing in proposing backup plans that are mostly based on optical sensing. Nevertheless, limited visibility, resolution or surface texture and need for additional sensor(s) that increase the payload as well as processing complexity are limiting such approaches for specific rather than general usage.

For these reasons – and in analogy to manned aircraft – navigation self-sufficiency (autonomy) remains to play a primordial role for drone control and operational safety.

Furthermore, the quality and reliability of autonomous navigation is likely to influence future deliverance of authorizations for BVLOS (beyond visual line of sight) operations. The work of Mehran Khaghani shows a new and promising way, how this can be achieved elegantly and efficiently without adding additional sensors on board. Indeed, this research has shown that up to 100 times improvement is possible without changing the sensory setup on board the UAV. The described approach goes against the mainstream of integrated navigation based on kinematic modeling for aerial vehicles, as it is known for decades. It does it by replacing the heart of the system based on inertial sensing by that stemming from the knowledge of vehicle aerodynamics and autopilot commands.

The originality of the proposed approached was first challenged by several referenced publications in peer-reviewed journals and conferences; second by the revision of US patent– granted on May 22, 2018 under patent number US 9,978,285. The proposed method also echoed in drone community by attracting industrial and governmental funding for its further development and implementation in different flavors.

**Prof. Dr. Jan Skaloud** EPFL Lausanne Director of project **Prof. Dr. Alain Geiger** ETH Zürich President of SGK

## Abstract

The dominant navigation system for small civilian UAVs today is based on integration of inertial navigation system (INS) and global navigation satellite system (GNSS). This strategy works well to navigate the UAV, as long as proper reception of GNSS signal is maintained. However, when GNSS outage occurs, the INS-based navigation solution drifts very quickly, considering the limited quality of IMU(s) employed in INS for small UAVs. In beyond visual line of sight (BVLOS) flights, this poses the serious danger of losing the UAV and its eventual falling down. Limited payload capacity and cost for small UAVs, as well as the need for operating in different conditions, with limited visibility for example, make it challenging to find a solution to reach higher levels of navigation autonomy based on conventional approaches. This thesis aims to improve the accuracy of autonomous navigation for small UAVs by at least one order of magnitude. The proposed novel approach employs vehicle dynamic model (VDM) as process model within navigation system, and treats data from other sensors such as IMU, barometric altimeter, and GNSS receiver, whenever available, as observations within the system. Such improvement comes with extra effort required to determine the VDM parameters for any specific UAV. This work investigates the internal capability of the proposed system for estimating VDM parameters as part of the augmented state vector within an extended Kalman filter (EKF) as the estimator. This reduces the efforts required to setup such navigation system that is platform dependent. Multiple experimental flights using two custom made fixed-wing UAVs are presented together with Monte-Carlo simulations. The results reveal improvements of 1 to 2 orders of magnitude in navigation accuracy during GNSS outages of a few minutes' duration. Computational cost for the proposed VDM-based navigation does not exceed 3 times that of conventional INS-based systems, which establishes its applicability for online application. A global sensitivity analysis is presented, spotting the VDM parameters with higher influence on navigation performance. This provides insight for design of calibration procedures. The proposed VDM-based navigation system can be interesting for professional UAVs from at least two points of view. Firstly, it adds little to no extra hardware and cost to the UAV. Secondly and more importantly, it might be currently the only way to reach such significant improvement in navigation autonomy for small UAVs regardless of visibility conditions and electromagnetic signals reception. Possibly, such environmental condition independence for navigation system may be needed to obtain certifications from legal authorities to expand UAV applications to new types of mission.

Key words: UAV, Navigation, Vehicle Dynamic Model, GNSS Outage, Inertial, Kalman Filter

# Contents

Li	ist of figures xv				$\mathbf{x}\mathbf{v}$
$\mathbf{L}^{\mathrm{i}}$	ist of	tables	5	:	xix
$\mathbf{C}$	onve	ntions	and Notation		xxi
Ir	trod	uction	ı		1
	Con	text .			1
	Rese	earch O	Dejectives		2
	App	roach			2
	The	sis Out	line		3
Ι	Pre	elimin	aries		<b>5</b>
1	Inte	egrated	d Navigation		7
	1.1	Funda	amentals of INS-based Navigation		8
		1.1.1	Coordinate Systems		8
		1.1.2	Attitude Representation		8
		1.1.3	Frames Definition		11
		1.1.4	Inertial Navigation System (INS)		15
		1.1.5	Global Navigation Satellite System (GNSS)		17
	1.2	Estim	ation Methods		18
		1.2.1	Linear Kalman Filter (KF)		19
		1.2.2	Extended Kalman Filter (EKF)		21
		1.2.3	Unscented Kalman Filter (UKF)		22

#### Contents

	1.3	Sensor	Error Modeling	22
		1.3.1	Systematic and Stochastic Errors	22
		1.3.2	Stochastic Processes	23
		1.3.3	Stochastic Modeling	24
		1.3.4	Traditional Methods $\ldots$	25
		1.3.5	Limitations of Traditional Stochastic Modeling Methods	26
		1.3.6	Generalized Method of Wavelet Moments (GMWM) $\ . \ . \ . \ . \ .$	26
<b>2</b>	Bas	is for I	Dynamic Modeling of UAVs	27
	2.1	Classic	cal Mechanics	27
		2.1.1	Newtonian-Euler Formulation	27
		2.1.2	Lagrangian Formulation	28
		2.1.3	Hamiltonian Formulation	28
	2.2	Flight	Aerodynamics	29
II	V	DM-B	ased Navigation Framework	31
11 3	V] The	DM-B eory an	ased Navigation Framework d System Architecture	31 33
II 3	<b>V</b> ] <b>The</b> 3.1	DM-B eory an Relate	ased Navigation Framework d System Architecture d Work	<b>31</b> <b>33</b> 33
11 3	V The 3.1 3.2	DM-B eory an Relate Metho	ased Navigation Framework d System Architecture d Work	<ul> <li>31</li> <li>33</li> <li>35</li> </ul>
II 3	V] The 3.1 3.2	DM-B eory an Relate Metho 3.2.1	ased Navigation Framework         d System Architecture         d Work	<ul> <li><b>31</b></li> <li><b>33</b></li> <li>33</li> <li>35</li> <li>37</li> </ul>
II 3	VI The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga	ased Navigation Framework         d System Architecture         d Work	<ul> <li><b>31</b></li> <li><b>33</b></li> <li>33</li> <li>35</li> <li>37</li> <li>37</li> </ul>
II 3	VI The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga 3.3.1	ased Navigation Framework         d System Architecture         d Work	<ul> <li><b>31</b></li> <li><b>33</b></li> <li>35</li> <li>37</li> <li>37</li> <li>39</li> </ul>
11 3	VI The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga 3.3.1 3.3.2	ased Navigation Framework         d System Architecture         d Work	<ul> <li>31</li> <li>33</li> <li>33</li> <li>35</li> <li>37</li> <li>37</li> <li>39</li> <li>40</li> </ul>
II 3	V1 The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga 3.3.1 3.3.2 3.3.3	ased Navigation Framework         d System Architecture         d Work	<ul> <li>31</li> <li>33</li> <li>33</li> <li>35</li> <li>37</li> <li>37</li> <li>39</li> <li>40</li> <li>41</li> </ul>
II 3	V1 The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga 3.3.1 3.3.2 3.3.3 State S	ased Navigation Framework         d System Architecture         d Work	<ul> <li>31</li> <li>33</li> <li>33</li> <li>35</li> <li>37</li> <li>37</li> <li>39</li> <li>40</li> <li>41</li> <li>42</li> </ul>
II 3	V1 The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga 3.3.1 3.3.2 3.3.3 State S 3.4.1	ased Navigation Framework         d System Architecture         d Work	<ul> <li>31</li> <li>33</li> <li>33</li> <li>35</li> <li>37</li> <li>37</li> <li>39</li> <li>40</li> <li>41</li> <li>42</li> <li>43</li> </ul>
11 3	VI The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga 3.3.1 3.3.2 3.3.3 State S 3.4.1 3.4.2	ased Navigation Framework d System Architecture d Work	<ul> <li>31</li> <li>33</li> <li>33</li> <li>35</li> <li>37</li> <li>37</li> <li>39</li> <li>40</li> <li>41</li> <li>42</li> <li>43</li> <li>43</li> </ul>
11 3	V1 The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga 3.3.1 3.3.2 3.3.3 State S 3.4.1 3.4.2 3.4.3	ased Navigation Framework d System Architecture d Work dology Overview framework Theoretical Benefits of VDM in Navigation framework General Form of VDM frameters VDM as Process Model framework VDM Parameters framework Wind Velocity framework	<ul> <li>31</li> <li>33</li> <li>33</li> <li>35</li> <li>37</li> <li>37</li> <li>39</li> <li>40</li> <li>41</li> <li>42</li> <li>43</li> <li>43</li> <li>44</li> </ul>
11 3	V1 The 3.1 3.2 3.3	DM-B cory an Relate Metho 3.2.1 Naviga 3.3.1 3.3.2 3.3.3 State S 3.4.1 3.4.2 3.4.3 3.4.4	ased Navigation Framework         d System Architecture         d Work	<ul> <li>31</li> <li>33</li> <li>33</li> <li>35</li> <li>37</li> <li>39</li> <li>40</li> <li>41</li> <li>42</li> <li>43</li> <li>43</li> <li>44</li> <li>44</li> </ul>
11 3	V1 The 3.1 3.2 3.3	DM-B eory an Relate Metho 3.2.1 Naviga 3.3.1 3.3.2 3.3.3 State 9 3.4.1 3.4.2 3.4.3 3.4.4 3.4.5	ased Navigation Framework d System Architecture d Work	<ul> <li>31</li> <li>33</li> <li>33</li> <li>35</li> <li>37</li> <li>39</li> <li>40</li> <li>41</li> <li>42</li> <li>43</li> <li>43</li> <li>44</li> <li>44</li> <li>45</li> </ul>

#### Contents

	3.5	Obser	vation Models	48
		3.5.1	IMU data	48
		3.5.2	GNSS Position and Velocity	49
		3.5.3	Barometric Altitude	50
		3.5.4	Airspeed	50
<b>4</b>	Fixe	e <b>d-Wi</b> ı	ng Implementation	53
	4.1	VDM	for a Typical Fixed-Wing UAV	53
		4.1.1	Aerodynamic Forces	53
		4.1.2	Aerodynamic Moments	55
		4.1.3	Air Density Model	56
		4.1.4	Actuator Dynamics	56
	4.2	Exper	imental Setup	57
		4.2.1	Platforms	57
		4.2.2	Sensor Setup	58
		4.2.3	Actuators	60
	4.3	Filter	Setup	60
		4.3.1	Stochastic Models for Observations	61
		4.3.2	Initialization	62
		4.3.3	Model Linearization for EKF	63
II	I R	lesults	and Analyses	65
5	Mai	in Res	ults	67
	5.1	Proof	of Concept	69
		5.1.1	Navigation States	69
		5.1.2	Auxiliary States	70
		5.1.3	Discussion on Observability	74
	5.2	Evalua	ation of Wind Effects	77
		5.2.1	Navigation States	77
		5.2.2	Auxiliary States	79
	5.3	Valida	tion of Simulations	82

		5.3.1	Emulation Scenario			84
		5.3.2	Experimental Scenario			88
		5.3.3	Observability Discussions for Emulation Scenario			89
		5.3.4	Observability Discussions for Experimental Scenario			93
	5.4	Furthe	er Experimental Results			96
		5.4.1	Calibration Flight			96
		5.4.2	Evaluation Flight			99
6	Fur	ther A	nalyses		-	105
	6.1	Globa	l Sensitivity Analysis			105
		6.1.1	Theory			106
		6.1.2	Results			107
	6.2	VDM-	based Navigation without IMU			110
		6.2.1	Emulation Scenario			110
		6.2.2	Experimental Scenario	 •		113
	6.3	Comp	utational Cost	 •	 •	115
I۷	V C	Conclu	sion Remarks		1	117
7	Cor	nclusio	ns and Suggestions		-	119
	7.1	Conclu	usions Remarks			119
	7.2	Sugge	stions for Future Research	 •	 •	121
A	Ma	nual M	Iodel Linearization for a Simplified Case		-	123
$\mathbf{A}_{\mathbf{j}}$	ppen	dix				123
Bi	bliog	graphy			-	138

# List of Figures

1.1	Cartesian $(x,y,z)$ and ellipsoidal $(\phi,\lambda,h)$ coordinate systems (adapted from [32])	9
1.2	Inertia $(i)$ , Earth $(e)$ , local-level $(l)$ , and body $(b)$ frames	12
1.3	Body (b) and wind (w) frames with airspeed ( <b>V</b> ), wind velocity ( <b>w</b> ), UAV velocity ( <b>v</b> ), angle of attck ( $\alpha$ ), and sideslip angle ( $\beta$ )	14
1.4	Three types of estimation problems (adapted from $[23]$ ) $\ldots$ $\ldots$ $\ldots$	19
3.1	Simplified system architecture in [44] showing roles of IMU and VDM in PVA determination	34
3.2	Simplified system architecture in [8] showing roles of IMU and VDM in PVA determination	35
3.3	Simplified system architecture in current research showing roles of IMU and VDM in PVA determination	37
3.4	Conventional (INS-based) navigation system for UAVs	38
3.5	General scheme for the proposed (VDM-based) navigation system for UAVs $$ .	39
3.6	VDM-based navigation filter architecture $(\tilde{\mathbf{X}}_k \equiv \hat{\mathbf{X}}_{k k-1})$	42
3.7	IMU frame alignment with respect to body frame	46
3.8	Airspeed frame alignment with respect to body frame	47
4.1	Schematic of the fixed-wing UAV used in this research with propeller, aileron(s), elevator(s), and rudder	54
4.2	Custom made UAVs, TOPO plane 1 (a) and TOPO plane 2 (b), together with the IMU board (c)	57
5.1	Reference trajectory and the solution from a sample run with GNSS signals available during first $100s$ only (adapted from [38])	69
5.2	Position errors for all the 100 Monte-Carlo runs with empirical RMS and predicted confidence level (adapted from [38])	70
5.3	RMS of attitude errors from 100 Monte-Carlo runs (adapted from [38])	71

### List of Figures

5.4	Comparison between INS-based and VDM-based navigation: RMS of position and yaw errors from 100 Monte-Carlo runs (adapted from [38])	. 71
5.5	RMS of estimation error for IMU error terms from 100 Monte-Carlo runs, together with the amplitude of simulated errors (adapted from $[38]$ )	. 72
5.6	RMS of estimation error for VDM parameters from 100 Monte-Carlo runs (adapted from [38])	. 73
5.7	RMS of estimation error for wind speed from 100 Monte-Carlo runs, together with real wind speed [43] (realization values) (adapted from [38])	. 74
5.8	Ratios of uncertainties of states during GNSS presence and GNSS outage periods (navigation starts with GNSS available at $t = 0s$ , GNSS outage begins at $t = 100s$ , and navigation ends at $t = 400s$ ) (adapted from [38])	. 75
5.9	Correlation matrix at the end of navigation (adapted from $[38]$ )	. 76
5.10	Comparison between INS-based and VDM-based navigation: RMS of position and yaw errors from 50 Monte-Carlo runs (case B) (adapted from $[39]$ )	. 78
5.11	Position errors for all the 50 Monte-Carlo runs (case B) (adapted from $[39])$ .	. 79
5.12	RMS of position and attitude errors from 50 Monte-Carlo runs (case B) (adapted from [39])	. 80
5.13	RMS of estimation error for IMU error terms from 50 Monte-Carlo runs, together with the amplitude of simulated errors	. 81
5.14	RMS of estimation error for VDM parameters from 50 Monte-Carlo runs	. 82
5.15	RMS of wind estimation errors from 50 Monte-Carlo runs (case B) (adapted from [39])	. 83
5.16	Horizontal view of reference trajectory in Monte-Carlo simulations and navigation solutions from a sample run with GNSS signals available during first $100s$ only (adapted from [40])	. 84
5.17	Position error in Monte-Carlo simulation for VDM-based navigation, with maximum RMS of 38 $m$ during 3 minutes of GNSS outage. The same error for INS-based navigation was 681 $m$ (adapted from [40]).	. 85
5.18	RMS of attitude errors in 100 Monte-Carlo simulation runs for VDM-based navigation, with maximum RMS of $0.20^{\circ}$ , $0.19^{\circ}$ , and $0.96^{\circ}$ for roll, pitch, and yaw, respectively, during 3 minutes of GNSS outage. The same errors for INS-based navigation were $0.67^{\circ}$ , $0.68^{\circ}$ , and $7.28^{\circ}$ (adapted from [40])	. 86
5.19	RMS of estimation error for IMU error terms from 100 Monte-Carlo runs for VDM-based navigation, together with the amplitude of simulated errors (realization values) (adapted from [40])	. 87
5.20	RMS of estimation error for VDM parameters from 100 Monte-Carlo runs (adapted from [40])	. 88
5.21	RMS of estimation error for wind speed from 100 Monte-Carlo runs, together with real wind speed (adapted from [40])	. 89

5.22	Horizontal view of reference trajectory in the real flight and navigation solutions with GNSS signals available during first 100s only (altitude change $\simeq 50m$ ) (adapted from [40])
5.23	Position error in the real flight for VDM-based navigation, with maximum value of 53 $m$ during 3 minutes of GNSS outage. The same error for INS-based navigation was 2076 $m$ (adapted from [40])
5.24	Velocity components for VDM-based navigation with GNSS signals available during first $100s$ only (adapted from [40]) $\dots \dots \dots$
5.25	Ratios of uncertainties of states during GNSS presence and GNSS outage periods in a sample Monte-Carlo run (GNSS outage begins at $t = 100s$ , and navigation ends at $t = 280s$ ) (adapted from [40])
5.26	Correlation matrix at the end of navigation in in a sample Monte-Carlo run (adapted from [40])
5.27	Ratios of uncertainties of states during GNSS presence and GNSS outage periods in the real flight (GNSS outage begins at $t = 100s$ , and navigation ends at t = 280s) (adapted from [40])
5.28	Correlation matrix at the end of navigation in the real flight (adapted from [40]) 96
5.29	Horizontal view (left) and altitude profile (right) for the reference trajectory used in VDM parameters calibration
5.30	IMU error estimated via INS-based navigation filter in Step 2
5.31	Refinement of GNSS lever-arm via INS-based navigation filter in Step 2 $\ldots$ 100
5.32	Relative change in VDM parameters during calibration via VDM-based naviga- tion filter in Step 3: mean of absolute values for all parameters and 3 sample ones
5.33	Position error in evaluation of VDM parameters calibration in Step 4 with 3 minutes of GNSS outage at the end
5.34	Horizontal view (left) and altitude profile (right) for the reference trajectory used in evaluation flight with VDM-based navigation solution for 3 minutes of GNSS outage at the end
5.35	Position error in evaluation flight with 3 minutes of GNSS outage at the end $\therefore 102$
5.36	Velocity components for VDM-based navigation in evaluation flight with 3 min- utes of GNSS outage at the end
6.1	Roll errors for all the 100 runs of Monte-Carlo simulation with empirical RMS 111
6.2	Pitch errors for all the 100 runs of Monte-Carlo simulation with empirical RMS 112
6.3	Yaw errors for all the 100 runs of Monte-Carlo simulation with empirical RMS 112
6.4	Roll error for experimental data
6.5	Pitch error for experimental data
6.6	Yaw error for experimental data

# List of Tables

4.1	Geometric measures of the fixed-wing platform
4.2	MEMS IMU error stochastic model
4.3	Stochastic models for GNSS data errors
4.4	Airspeed error stochastic model
4.5	Summary of the map between control commands in PWM and actuator states in steady condition for TOPO plane 2
4.6	Nominal values of VDM parameters (aerodynamic coefficients) [19] 63
5.1	Summary of results section organization
5.2	Summary of Monte-Carlo simulation cases (adapted from [39])
5.3	Wind velocity statistics for case B (adapted from $[39]$ )
5.4	Summary of maximum navigation errors for cases A to D [39] 80
5.5	Maximum RMS of position and attitude estimation error in Monte-Carlo simulations with and without barometric altimeter [40]
6.1	First order $(S)$ and total effects $(S_T)$ indices in sensitivity analysis of aerody- namic forces (limited to 3 decimal digits)
6.2	Maximum position error during 3 minutes of GNSS outage with perturbations on most and least influential VDM parameters in aerodynamic forces 109
6.3	First order $(S)$ and total effects $(S_T)$ indices in sensitivity analysis of aerody- namic moments (limited to 3 decimal digits)
6.4	Maximum position error during 3 minutes of GNSS outage with perturbations on most and least influential VDM parameters in aerodynamic moments 110
6.5	Maximum RMS of attitude errors for VDM/GNSS and INS/GNSS navigation in emulation scenario, during and after the first minute for mitigation of initialization errors
6.6	VDM/GNSS and INS/GNSS attitude errors in experimental scenario 113
6.7	Computation time comparison (VDM-based 1 and 2: without and with VDM parameters estimation, respectively)

# Abbreviations and Nomenclature

## $\mathbf{Abbreviations}^1$

3D	Three Dimensional
ACF	Autocorrelation Function
BVLOS	Beyond Visual Line of Sight
$\operatorname{CFD}$	Computational Fluid Dynamics
DCM	Direct Cosine Matrix
ECEF	Earth Centered Earth Fixed
EKF	Extended Kalman Filter
ENU	East-North-Up
GMWM	Generalized Method of Wavelet Moments
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ITRF	International Terrestrial Reference Frame
VLOS	Visual Line of Sight
MAV	Micro Aerial Vehicle
MEMS	Micro-Electro-Mechanical Systems
NED	North-East-Down
PPK	Post Processed Kinematic
PSD	Power Spectral Density
PVA	Position, Velocity, and Attitude
PVT	Position, Velocity, and Time
$\operatorname{RF}$	Radio Frequency
RMS	Root Mean Square
RTK	Real Time Kinematic
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VDM	Vehicle Dynamic Model
WGS	World Geodetic System

 $<sup>^1\</sup>mathrm{Abb}$  reviations that were not frequently used may have not been listed here, but described within the text upon usage.

## **Nomenclature**<sup>1</sup>

#### Scalars

$\alpha$	angle of attack
$\beta$	sideslip angle
$\beta_{GM}$	inverse of correlation time for first order Gauss-Markov process
$\delta_a$	aileron deflection
$\delta_e$	elevator deflection
$\delta_r$	rudder deflection
$\delta_{ij}$	Kronecker delta (1 for $i = j, 0$ otherwise)
$\lambda$	ellipsoidal longitude
$\pi$	the ratio between the circumference of any circle to its diameter
ρ	air density
$\sigma$	standard deviation
au	time constant
$\phi$	ellipsoidal latitude
$\omega_{ie}$	Earth rotation rate magnitude
$\omega_x$	roll rate along $\mathbf{x}^{b}$
$ ilde{\omega}_x$	dimensionless roll rate along $\mathbf{x}^b$
$\omega_y$	pitch rate along $\mathbf{y}^b$
$\tilde{\omega}_y$	dimensionless pitch rate along $\mathbf{y}^b$
$\omega_z$	yaw rate along $\mathbf{z}^{b}$
$\tilde{\omega}_z$	dimensionless yaw rate along $\mathbf{z}^{b}$
a	semi-major axis of reference ellipsoid
b	wing span
$\bar{c}$	mean aerodynamic chord
$C_{F_T}$	thrust force coefficients
$C_{F_x}$	drag force coefficients
$C_{F_y}$	side force coefficients
$C_{F_z}$	lift force coefficients
$C_{M_x}$	roll moment coefficients
$C_{M_{y}}$	pitch moment coefficients
$C_{M_z}$	yaw moment coefficients
D	propeller diameter
e	eccentricity of reference ellipsoid
$F_T$	thrust force
$F_x^w$	drag force along $\mathbf{x}^w$
$F_{y}^{w}$	side force along $\mathbf{y}^w$
$\check{F_z^w}$	lift force along $\mathbf{x}^w$
h	altitude
J	thrust advance ratio

<sup>&</sup>lt;sup>1</sup>Some of the listed symbols may have been also used with a different meaning within the text, in which case such meaning is described upon usage. Furthermore, symbols that were not frequently used may have not been listed here, but described within the text upon usage.

m	mass
$M_x^b$	roll moment, expressed in body frame
$M_{y}^{b}$	pitch moment, expressed in body frame
$M_z^b$	yaw moment, expressed in body frame
p	pitch [angle]
$p_0$	ambient pressure at zero height
$ar{q}$	dynamic pressure
r	roll [angle]
$R_a$	gas constant for air
$R_M$	meridian radius of curvature
$R_P$	prime vertical radius of curvature
S	wing surface
t	time
Т	correlation time
$T_0$	ambient temperature at zero height
$v_x^b$	velocity component along $\mathbf{x}^b$
$v_y^b$	velocity component along $\mathbf{y}^b$
$v_z^b$	velocity component along $\mathbf{z}^{b}$
V	airspeed magnitude
$V_x^b$	airspeed component along $\mathbf{x}^{b}$
$V_y^b$	airspeed component along $\mathbf{y}^b$
$V_z^b$	airspeed component along $\mathbf{z}^{b}$
$w_N$	wind velocity component along $\mathbf{x}^l$ (north)
$w_E$	wind velocity component along $\mathbf{y}^l$ (east)
$w_D$	wind velocity component along $\mathbf{z}^l$ (down)
y	yaw [angle]

## Vectors and Matrices

$\alpha$	angular acceleration
$\Phi$	transition matrix
$oldsymbol{\omega}^a_{bc}$	angular velocity of c-frame with respect to b-frame, expressed in a-frame
$\mathbf{\Omega}^{a}_{bc}$	skew-symmetric matrix representation of $\omega_{bc}^{a}$ defined in Equation (1.1.8)
a	linear acceleration
$\mathbf{C}^a_b$	rotation matrix from b-frame to a-frame, so that for any vector $\mathbf{v}$ : $\mathbf{v}^a = \mathbf{C}_b^a \mathbf{v}^b$
$\mathbf{C}_i$	elementary rotation matrix about $i^{th}$ coordinate defined in Equation (1.1.2)
$\mathbf{f}_{b}^{a}$	specific force applied on point b expressed in a-frame
<b>F</b>	[linearized] system dynamics matrix
$\check{\mathbf{g}}$	gravitational acceleration
g	gravity acceleration (gravitational minus centrifugal)
$\mathbf{g}_{h}^{a}$	gravity acceleration applied on point b or origin of b-frame expressed in a-frame
G	noise shaping matrix
н	[linearized] observation model matrix
i	unit vector along $\mathbf{x}$ axis
Ι	identity matrix

## Abbreviations and Nomenclature

$\mathbf{I}^b$	matrix of mass moments of inertia, expressed in b-frame
j	unit vector along $\mathbf{y}$ axis
k	unit vector along $\mathbf{z}$ axis
Κ	Kalman filter gain matrix
$\mathbf{L}$	[linearized] input matrix
$\mathbf{M}^{a}$	moment vector expressed in a-frame
$\mathbf{n}_a$	measurement noise for sensor $a$
Ν	matrix of non-orthogonality factors
Р	covariance matrix
$\mathbf{q}_{b}^{a}$	quaternion representing orientation of b-frame with respect to a-frame
$\mathbf{\bar{Q}}$	process noise covariance matrix
r	position vector
$\mathbf{r}_{e}^{l}$	position vector in e-frame in elliptical coordinates
$\ddot{\mathbf{R}}$	observation covariance matrix
$\mathbf{S}$	matrix of scale factor
$\mathbf{U}$	control / deterministic input
v	UAV velocity
$\mathbf{V}$	airspeed vector
$\mathbf{W}$	wind velocity / system dynamics white noise vector
$\mathbf{x}^{a}$	first axis in Cartesian a-frame
X	state vector
$\mathbf{X}_{a}$	actuator states
$\mathbf{X}_{ap}$	actuator parameters states
$\mathbf{X}_{e}^{I}$	IMU error states
$\mathbf{X}_{e}^{B}$	barometer error states
$\mathbf{X}_{e}^{A}$	Airspeed error states
$\mathbf{X}_m^I$	IMU mounting misalignment states
$\mathbf{X}_m^G$	GNSS antenna mounting misalignment states
$\mathbf{X}_m^B$	barometer mounting misalignment states
$\mathbf{X}_m^A$	airspeed sensor mounting misalignment states
$\mathbf{X}_n$	navigation states
$\mathbf{X}_{vp}$	VDM parameters states
$\mathbf{X}_w$	wind velocity states
$\mathbf{y}^{a}$	second axis in Cartesian a-frame
$\mathbf{Z}$	measurements vector
$\mathbf{z}^{a}$	third axis in Cartesian a-frame

## Miscellaneous

	inner product
$\otimes$	quaternion product
$  \mathbf{x}  $	Euclidean norm (2-norm) of $\mathbf{x}$
$[\boldsymbol{\omega}]_q$	quaternion representation of vector $\boldsymbol{\omega}$ , defined as $[0, \boldsymbol{\omega}^T]^T$
$\nabla$	gradient operator
$\frac{\partial x}{\partial y}$	partial derivative of $x$ with respect to $y$
$\dot{x}$	time derivative of $x$

### Abbreviations and Nomenclature

$rac{Dx}{Dt} \mathbf{x}^a \mathbf{x}^a \mathbf{x}^T \mathbf{\hat{X}} \mathbf{\hat{X}}_{k l}$	material derivative of $x$ vector $\mathbf{x}$ expressed in a-frame transpose of vector or matrix $\mathbf{x}$ estimated value of $\mathbf{X}$ estimated value of $\mathbf{X}$ at step $k$ given the observations up to step $l$
$X_{CM}$ $X_{GM}$ $X_{RW}$ $X_{SF}$ $X_{WN}$	associated to center of mass associated to first order Gauss-Markov process associated to random walk associated to scale factor associated to white noise
$egin{array}{l} \mathcal{N}\left(\mu,\sigma^2 ight) \ \mathbb{E}[x] \ \mathbb{R} \ \mathbb{Z} \end{array}$	normal (Gaussian) distribution with the mean $\mu$ and variance $\sigma^2$ expected value of $x$ set of real numbers set of integers

## Introduction

### Context

 $INS^1/GNSS^2$  (INS-based navigation, in general) is the dominant navigation system for small UAVs<sup>3</sup> today [57, 9]. In such a system, INS provides autonomous PVA<sup>4</sup> solution with high frequency (typically 50 to 500 Hz for UAVs) in a dead reckoning fashion. This PVA solution gets updated in a navigation filter whenever observations from other sensors/sources are available. The most common navigation aid comes from GNSS receivers, which in loosely coupled scheme consists of PVT<sup>5</sup> solution at a lower frequency (1 to 10 Hz, normally). The integration of these data types can provide solutions with sufficient short-term and long-term accuracy for many applications. Since IMU<sup>6</sup> is an environment dependent sensor (its measurements do not depend on external infrastructures or electromagnetic signal reception, for example), INS offers "autonomous navigation". Another common aiding sensor is barometer providing altitude data autonomously.

During regular presence of GNSS observations, conventional INS-based navigation provides sufficient accuracy for most applications within reasonable expectations from the sensor setup. However, navigation faces serious challenges when GNSS outage happens [46, 80, 30]. This is not a rare situation and can happen due to intentional corruption of GNSS signals (jamming and spoofing), or loss of direct line of sight to the satellites, or unintentional electromagnetic interference in satellite signal reception [27]. In such cases, navigation is continued based on stand-alone INS with possible aiding from navigation aids such as barometric altimeters. The accuracy of the data provided by INS is directly determined by the quality of the IMU(s) that is used in the system. The long-term accuracy of  $3D^7$  inertial coasting based on small and low-cost IMUs available for small UAVs today is so low that after only seconds or a minute of GNSS outage, the position uncertainty is too far from being of practical use. If this "drift" is not controlled by some other means in case of long outages in BVLOS<sup>8</sup> flights, the UAV gets quickly lost in space [8], posing dangers on people, animals, or objects on ground. This is one

<sup>&</sup>lt;sup>1</sup>Inertial Navigation System

 $<sup>^2 \</sup>mathrm{Global}$  Navigation Satellite System

<sup>&</sup>lt;sup>3</sup>Unmanned Aerial Vehicle

<sup>&</sup>lt;sup>4</sup>Position, Velocity, and Attitude

<sup>&</sup>lt;sup>5</sup>Position, Velocity, and Time

<sup>&</sup>lt;sup>6</sup>Inertial Measurement Unit

<sup>&</sup>lt;sup>7</sup>Three Dimensional

<sup>&</sup>lt;sup>8</sup>Beyond Visual Line Of Sight

#### Approach

of the reasons why operation of  $MAVs^1$  is limited to  $VLOS^2$  in most developed countries (e.g., USA [1], Canada [10], and a large part of the EU [51]).

### **Research Objectives**

There are mainly two categories of approaches to address the problem of rapid drift of navigation solution for small UAVs during GNSS outages. The first is trying to improve sensor error modeling (with IMU in focus) using advanced techniques [58, 54, 55], and the second one is employing additional sensors to aid the system [82, 48, 77]. The first approach, essentially, can only improve modeling and compensation for time-correlated errors. It cannot help with time-uncorrelated errors, which in many cases is a main contributor of low-cost inertial sensors errors. In general, this approach does not provide qualitatively sufficient improvements for UAVs to overcome problems in GNSS outage conditions. Solutions related to the second approach add cost, weight, and complexity to the system, and more importantly, their performance may depend on environmental conditions that are not met all the times, which challenges the autonomy of the navigation system. A widely used (yet partial) solution of the second category is employing vision based methods that provide relative or absolute measurements to inertial navigation [80, 11, 3, 69]. Apart from adding extra weight and hardware and software complications, their correct functioning requires some prerequisites on light, visibility, and terrain texture. While very attractive in nominal scenarios, they might not work reliably, for example, at night or in foggy conditions or over ground with uniform texture (vegetation, water, snow, etc.).

The main objective of this research is to introduce a navigation algorithm that considerably (i.e., at least by one order of magnitude) mitigates the quick drift of low-cost inertial navigation during GNSS outage in airborne environment, while preserving the navigation autonomy and avoiding additional sensors and thus extra cost and weight. Ideally, the effort required to design and implement such a navigation system is kept reasonably low so it can be practically adopted for small UAVs. A suitable solution of such kind can be extremely beneficial for increasing the reliability of autonomous navigation of small UAVs.

### Approach

UAV is a physical system that can be associated with a dynamic model (called VDM<sup>3</sup> throughout this thesis) that provides some information on its behavior. This fact is totally ignored by conventional (INS-based) navigation systems, which are fundamentally based on sensor data. Even in their process models, INS-based navigation filters still rely on sensor data from IMU(s). This means that all the knowledge on dynamic behavior of the UAV, which is basically available for free from hardware point of view, is not used in navigation. In other words, the physical constraints on UAV motion are not considered, meaning that any physically impossible motion suggested by the sensors due to their errors is accepted within the navigation system.

<sup>&</sup>lt;sup>1</sup>Micro Aerial Vehicle

<sup>&</sup>lt;sup>2</sup>Visual Line Of Sight

<sup>&</sup>lt;sup>3</sup>Vehicle Dynamic Model

The primary idea of this research is to integrate the knowledge on dynamic behavior of the UAV for improving the quality of navigation (in terms of PVA determination) with emphasis on improving positioning accuracy in GNSS outage conditions. The "price to pay" is to drop the very attractive property of INS-based navigation systems from design point of view, which is platform independence. INS-based navigation systems can be used on any platform without knowing anything about it, as long as the sensor setup is suitable for the operation conditions (such as maximum speed for GNSS or acceleration for IMU). VDM-based navigation on the other hand, requires some knowledge on the specific platform navigation for which is intended.

### Thesis Outline

The thesis is structured in 4 parts, comprising 7 chapters in total.

#### Part I: Preliminaries

This part provides background material related to integrated navigation and dynamic modeling.

- **Chapter 1:** Essential aspects of integrated navigation are reviewed. A general description of INS and GNSS as the main players in integrated navigation of UAVs today is provided, along wide fundamental definitions and models used in INS-based navigation systems that are expandable to VDM-based navigation. Kalman filter as the most frequently used estimator is then briefed, followed by some notes on modeling techniques for stochastic sensor errors.
- **Chapter 2:** A brief overview of classical mechanics and flight aerodynamics is presented, as the requirements for deriving the VDM for a UAV. Advantages of different formulations of classical mechanics and their relevance to different situations in dynamic modeling are also briefly discussed.

#### Part II: VDM-Based Navigation Framework

The main contribution of the thesis is presented in this part, which is the development of the VDM-based navigation.

- Chapter 3: The theory of VDM-based navigation is developed and presented in general form for aerial applications. Navigation system architecture is detailed, followed by presentation of process models for all the augmented states and the observation models for all the sensors employed in the research, considering their errors and mounting misalignments.
- **Chapter 4**: The general VDM-based navigation introduced in Chapter 3 is applied to a specific fixed-wing platform with a single propeller in front and four (or five) control surfaces (two [coupled] ailerons, an elevator (or two coupled ones), and a rudder). The experimental setup used in the research is detailed, followed by description of navigation filter setup.

#### Part III: Results and Analyses

Simulation and experimental results are presented and analyzed in this part.

- Chapter 5: Results and analyses of applying VDM-based navigation to the specific UAV introduced in previous chapter are presented. At the first stage, proof of concept is established via Monte-Carlo simulations. Effects of wind velocity on the performance of the navigation system are also evaluated via Monte-Carlo simulations. Experimental results from flights of two platforms of the same type are then presented to backup the simulation results and reveal practical details about navigation performance.
- Chapter 6: Some further analyses of VDM-based navigation are presented. First, a global sensitivity analysis is presented to reveal the most and the least influential VDM parameters for the specific UAV used in the research, results of which can potentially assist the VDM calibration. Afterwards, a navigation scenario without IMU is presented to assess the ability of VDM-based navigation in attitude estimation in absence of IMU data, which could have practical application in case of IMU failure or malfunction. Finally, a brief discussion on computational cost of VDM-based navigation is presented, with comparison made to INS-based navigation.

#### Part IV: Conclusion Remarks

This short part, contains conclusion remarks from the conducted research in **Chapter 7**, followed by some suggestions for the future research on the subject.

## Part I

# Preliminaries

## Chapter 1

## **Integrated Navigation**

#### Overview

Integrated navigation can be viewed as an estimation problem to optimally combine data from different sources and deliver navigation information such as position and attitude. Among others, this task requires relevant modeling of stochastic errors of input data. This chapter establishes preliminary materials on essential aspects of integrated navigation. Most wordings and equations on "attitude representations", "frames definition", and "stochastic processes" are borrowed from [71]. The same is true for [23] regarding "estimation methods".

The term "navigation" is commonly understood as determination of position, velocity, and attitude (referred to as "navigation states" hereinafter) of a moving platform. Occasionally, this term is used equivalently as guidance, which usually refers to designing/planning desired navigation states along the mission, further aimed to reach by the control system using actuators to manipulate the motion of platform/vehicle. Throughout this document, the first meaning of navigation is always meant.

Navigation is performed based on data/information available from different sensors and models. A navigation system that fuses data/information from two or more sources is called an "integrated navigation" system [20, 21, 26, 63, 74]. The main motivation behind performing integrated navigation is to increase accuracy and reliability of navigation through combining usually complementary benefits of different sensors/models. Sensors/models can have differences in terms of short-term and long-term accuracy, autonomy, relativeness or absoluteness of provided data, navigation states they cover, data frequency, working conditions, resilience to environmental disturbances, and so on.

INS/GNSS is probably the most frequently used and well known integrated navigation system for outdoor applications in terrestrial, aerial, and marine applications [20, 21, 26, 63, 74]. As the navigation concept proposed in this thesis is closely related to INS-based navigation and INS/GNSS integration, fundamentals of the latter are briefly presented in this chapter, along with some estimation methods and sensor error modeling techniques. This presentation is not exhaustive, as it is meant to only provide sufficient basis for introduction of the VDM-based navigation as the main contribution of this research.

### 1.1 Fundamentals of INS-based Navigation

Based on data coming from IMU, namely specific forces and rotation rates/angular velocities, standalone or aided INS provides determination of PVA at high rates. INS is a dead reckoning system, meaning it calculates navigation states based on the solution of a previous time and the knowledge/estimation on the rate of change of this solution (velocity/acceleration) over the passed interval. While INS autonomously provides high frequency (50 to 2000 Hz, typically) PVA solution with good short term accuracy, it is prone to drift due to its dead reckoning nature that integrates not only the true signal measured by the IMU, but also the accompanying error. This solution, therefore, gets updated using other sources (navigation aids) whenever available. For outdoor applications, GNSS is usually the main and minimum navigation aid. GNSS provides absolute PVT solution with good long term accuracy at lower frequencies, provided that signals from sufficient number of satellites are well received.

#### 1.1.1 Coordinate Systems

A coordinate system provides a parameterization of position of any point in space. In navigation, two frequently used coordinate systems are Cartesian and ellipsoidal systems, both depicted in Figure 1.1.

#### Cartesian Coordinate System

In a 3D Cartesian coordinate system, position of a point p is determined by a vector  $[x, y, z]^T$  containing its signed distances (the coordinates) from an origin point O along three mutually orthogonal axes  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$ . For simplicity, sometimes the same notation is used for both the coordinates and the axes. This, however, does not normally make an ambiguity and the meaning is well understood in the context. If directions of  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  axes obey the right hand rule, the coordinate system is called "right handed", otherwise it is "left handed". Throughout this thesis, all the Cartesian coordinates are right handed.

#### Ellipsoidal Coordinate System

An ellipsoidal coordinate system, is set up in relation to Cartesian coordinates and is based on an ellipsoid of revolution with semi-major axis a along  $\mathbf{x}$  and  $\mathbf{y}$  and semi-minor axis balong  $\mathbf{z}$ . Ellipsoidal coordinates of a point p are expressed as  $[\phi, \lambda, h]^T$  representing latitude, longitude, and altitude (height), respectively. Although the definitions related to this frame should be clear from Figure 1.1, more formal definitions and transformations to/from Cartesian coordinates can be found in [32].

#### 1.1.2 Attitude Representation

Among several representations of attitude of a reference frame<sup>1</sup> (or an object) with respect to another one, three are frequently used in navigation, namely rotation matrix (or direction

 $<sup>^{1}</sup>$ A reference frame is a specific realization of a coordinate system. More details and definitions of the ones used in this thesis are presented in Section 1.1.3



Figure 1.1 – Cartesian (x, y, z) and ellipsoidal  $(\phi, \lambda, h)$  coordinate systems (adapted from [32])

cosine matrix), Euler angles, and quaternion. These three representations are reviewed in this section, with most wordings and equations borrowed from [71].

Vectors can be expressed in different reference frames, which are indicated as a superscript. For example,  $\mathbf{x}^a$  represents vector  $\mathbf{x}$  expressed in a-frame.

#### **Rotation Matrix**

A rotation matrix is a matrix transforming a vector expressed in a Cartesian frame b to the same vector expressed in another Cartesian frame a, and throughout this document is denoted as  $\mathbf{C}_{h}^{a}$ , where

$$\mathbf{x}^a = \mathbf{C}^a_b \mathbf{x}^b. \tag{1.1.1}$$

The matrix  $\mathbf{C}_{b}^{a}$  is an orthonormal matrix, therefore  $(\mathbf{C}_{b}^{a})^{T} = (\mathbf{C}_{b}^{a})^{-1} = \mathbf{C}_{a}^{b}$  and det  $(\mathbf{C}_{b}^{a}) = 1$ . The columns of  $\mathbf{C}_{b}^{a}$  matrix represent the unit vectors of b-frame projected along a-frame axes, or cosines of the three angles between each unit vector of b-frame and the three axes of a-frame. In other words, the element of the  $i^{th}$  row and  $j^{th}$  column of  $\mathbf{C}_{b}^{a}$  is the cosine of the angle between the  $i^{th}$  axis of a-frame and the  $j^{th}$  axis of b-frame. This is why  $\mathbf{C}_{b}^{a}$  is also called a DCM<sup>1</sup>.

 $<sup>^1\</sup>mathrm{Direct}$  Cosine Matrix

#### **Euler Angles**

The rotation from b-frame to a-frame can be performed by three sequential elementary rotation matrices, denoted as  $\mathbf{C}_i(\cdot): i = 1, 2, 3$ , associated to three Euler angles often called roll (r), pitch (p), and yaw (y) in navigation, and performed about associated axis of the rotated frame. The composed rotation is then given as following.

$$\mathbf{C}_{b}^{a} = \mathbf{C}_{1}(r) \ \mathbf{C}_{2}(p) \ \mathbf{C}_{3}(y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos r & \sin r \\ 0 & -\sin r & \cos r \end{bmatrix} \begin{bmatrix} \cos p & 0 & -\sin p \\ 0 & 1 & 0 \\ \sin p & 0 & \cos p \end{bmatrix} \begin{bmatrix} \cos y & \sin y & 0 \\ -\sin y & \cos y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
(1.1.2)

Similarly, the inverse transformation can be obtained as

$$\mathbf{C}_{a}^{b} = (\mathbf{C}_{b}^{a})^{T} = \mathbf{C}_{3}^{T}(y) \, \mathbf{C}_{2}^{T}(p) \, \mathbf{C}_{1}^{T}(r).$$

$$(1.1.3)$$

#### Quaternions

Instead of describing rotations between two frames using the Euler angles, *quaternions* can be used with some advantages (see [2], for example). 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  $\mathbf{u} = [u_x, u_y, u_z]^T$  [75]. 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 as

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{||\mathbf{u}||}{2}\right) \\ \left(\frac{u_x}{||\mathbf{u}||}\right) \sin\left(\frac{||\mathbf{u}||}{2}\right) \\ \left(\frac{u_y}{||\mathbf{u}||}\right) \sin\left(\frac{||\mathbf{u}||}{2}\right) \\ \left(\frac{u_z}{||\mathbf{u}||}\right) \sin\left(\frac{||\mathbf{u}||}{2}\right) \end{bmatrix}, \qquad (1.1.4)$$

where  $||\mathbf{u}|| = \sqrt{u_x^2 + u_y^2 + u_z^2}$ .

The operations of addition, subtraction, and multiplication by a scalar are done in the same manner as in vector algebra. The quaternion multiplication of  $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$  with another quaternion  $\mathbf{p} = [p_0, p_1, p_2, p_3]^T$ , denoted by the symbol  $\otimes$ , is defined as following.

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3 \\ q_0 p_1 + q_1 p_0 + q_2 p_3 - q_3 p_2 \\ q_0 p_2 + q_2 p_0 - q_1 p_3 + q_3 p_1 \\ q_0 p_3 + q_3 p_0 + q_1 p_2 - q_2 p_1 \end{bmatrix}$$
(1.1.5)

Extensive details about quaternion algebra can be found in [2, 75]. The relationship between the rotation matrix  $\mathbf{C}_b^a$  and its corresponding attitude quaternion  $\mathbf{q}_b^a$  is

$$\mathbf{C}_{b}^{a} = \mathbf{f}(\mathbf{q}_{b}^{a}) = \begin{bmatrix} q_{0}^{2} + q_{1}^{2} - q_{2}^{2} - q_{3}^{2} & 2(q_{1}q_{2} - q_{0}q_{3}) & 2(q_{1}q_{3} + q_{0}q_{2}) \\ 2(q_{1}q_{2} + q_{0}q_{3}) & q_{0}^{2} - q_{1}^{2} + q_{2}^{2} - q_{3}^{2} & 2(q_{2}q_{3} - q_{0}q_{1}) \\ 2(q_{1}q_{3} - q_{0}q_{2}) & 2(q_{2}q_{3} + q_{0}q_{1}) & q_{0}^{2} - q_{1}^{2} - q_{2}^{2} + q_{3}^{2} \end{bmatrix}.$$
 (1.1.6)

#### **Rotation of Reference Frames**

Reference frames may rotate arbitrarily with respect to each other. In order to study motions of reference frames experiencing such relative rotations, it is necessary to calculate derivatives of attitude representations.

If the vector  $\boldsymbol{\omega}_{ab}^{b} = [\omega_{1}, \omega_{2}, \omega_{3}]^{T}$  represents the angular velocity (rotation rate) of b-frame with respect to a-frame and expressed in b-frame, time-derivative of rotation matrix  $\mathbf{C}_{b}^{a}$  is given as [21]

$$\dot{\mathbf{C}}^a_b = \mathbf{C}^a_b \mathbf{\Omega}^b_{ab},\tag{1.1.7}$$

where  $\Omega_{ab}^{b}$  is the skew-symmetric matrix representation of associated  $\boldsymbol{\omega}_{ab}^{b} = [\omega_1, \omega_2, \omega_3]^T$  vector as

$$\mathbf{\Omega}_{ab}^{b} = \begin{bmatrix} \boldsymbol{\omega}_{ab}^{b} \times \end{bmatrix} = \begin{bmatrix} 0 & -\omega_{3} & \omega_{2} \\ \omega_{3} & 0 & -\omega_{1} \\ -\omega_{2} & \omega_{1} & 0 \end{bmatrix}.$$
(1.1.8)

The time-derivative of Euler angles can be shown to be calculated as following [75].

$$\dot{r} = \omega_1 + (\omega_2 \sin r + \omega_3 \cos r) \tan p$$
  

$$\dot{p} = \omega_2 \cos r - \omega_3 \sin r$$
  

$$\dot{y} = (\omega_2 \sin r + \omega_3 \cos r) \sec p$$
(1.1.9)

Note that the well known singularity of Euler angles can be seen in Equation (1.1.9) when  $p = \pm \pi/2$ .

Finally, the time-derivative of quaternion  $\mathbf{q}_b^a$  is expressed as [75]

$$\dot{\mathbf{q}}_{b}^{a} = \frac{1}{2} \mathbf{q}_{b}^{a} \otimes \left[\boldsymbol{\omega}_{ab}^{b}\right]_{q}, \qquad (1.1.10)$$

with 
$$\left[\boldsymbol{\omega}_{ab}^{b}\right]_{q} = [0, \omega_{1}, \omega_{2}, \omega_{3}]^{T}$$
 being the quaternion equivalent of  $\boldsymbol{\omega}_{ab}^{b} = [\omega_{1}, \omega_{2}, \omega_{3}]^{T}$ .

At points, it may be necessary to transform an angular velocity vector or its skew-symmetric matrix representation from one reference frame to another. In case of an angular velocity vector, this can be done in the same way it is done for any vector using rotation matrices as expressed in Equation (1.1.1).

$$\boldsymbol{\omega}_{ba}^c = \mathbf{C}_d^c \boldsymbol{\omega}_{ba}^d. \tag{1.1.11}$$

For the skew-symmetric matrix  $\Omega_{ba}^d$ , such transformation can be done as following.

$$\mathbf{\Omega}_{ba}^c = \mathbf{C}_d^c \; \mathbf{\Omega}_{ba}^d \; \mathbf{C}_c^d \tag{1.1.12}$$

#### 1.1.3 Frames Definition

A reference frame is a specific realization of a coordinate system, with a well defined origin and axes. In INS-based navigation, five reference frames are frequently used. These reference frames and the wind-frame, later used in VDM-based navigation, are briefly defined as following, with most wordings borrowed from [71]. More details on definition of these frames can be found in [32, 71].

#### Inertial Frame (i-frame)

Inertial frame is a non-accelerating and non-rotating reference frame that is at rest or subject to a uniform translational motion. In such a frame, the laws of Newtonian mechanics are valid. Considering the limited resolution and accuracy of most sensors today and the tolerance for the resulting errors, an inertial frame is approximated as a celestial frame with origin at the center of mass of the Earth, such that the  $\mathbf{x}^i$  axis points towards the vernal equinox and is, thus, the intersection line between the equatorial and the ecliptic plane, the  $\mathbf{z}^i$  axis points towards the mean celestial pole, and the  $\mathbf{y}^i$  axis completes the 3D right-handed Cartesian system [71]. Figure 1.2 depicts the i-frame, as well as e-frame, l-frame, and b-frame, defined in the following.



Figure 1.2 – Inertia (i), Earth (e), local-level (l), and body (b) frames

#### Earth Frame (e-frame)

Earth frame frame with origin at the center of mass of the Earth, such that the  $\mathbf{x}^e$  axis points towards the Greenwich meridian, the  $\mathbf{z}^e$  axis points towards the mean direction of the rotation axis of the Earth, and the  $\mathbf{y}^e$  axis completes the 3D right-handed Cartesian system. This frame is therefore an ECEF<sup>1</sup> frame. Examples of important realizations are the ITRF<sup>2</sup> and the WGS<sup>3</sup>-84 being the reference frame of GPS<sup>4</sup>. Any point in this frame can be expressed either in Cartesian coordinates  $[x^e, y^e, z^e]^T$ , or ellipsoidal coordinates  $[\phi, \lambda, h]^T$  [71].

<sup>&</sup>lt;sup>1</sup>Earth Centered Earth Fixed

<sup>&</sup>lt;sup>2</sup>International Terrestrial Reference Frame

<sup>&</sup>lt;sup>3</sup>World Geodetic System

<sup>&</sup>lt;sup>4</sup>Global Positioning System
# Local-Level Frame (l-frame)

Local-level frame is a local geodetic frame with arbitrary origin, for example a point on the Earth surface, such that the  $\mathbf{x}^l$  axis points to geographical north, the  $\mathbf{y}^l$  axis points to geographical east, and the  $\mathbf{z}^l$  axis points down along ellipsoidal normal. The NED<sup>1</sup> frame is right-handed. To be strictly correct, this frame should be called ellipsoidal tangential frame rather than local-level frame. However, for the purpose of navigation, the difference between astronomical and ellipsoidal local-level frames can be neglected [71]. Therefore, throughout the rest of this thesis, the local-level frame or local frame refers to the ellipsoidal NED frame.

#### Body Frame (b-frame)

The body frame is a frame attached to the body of the object/vehicle. The usual convention is choosing the forward or longitudinal direction of the vehicle for  $\mathbf{x}^b$  axis, while  $\mathbf{y}^b$  axis points towards right and  $\mathbf{z}^b$  axis is downward. The rotation matrix  $\mathbf{C}_l^b$  to transform vectors from local-level frame to body frame (as  $\mathbf{v}^b = \mathbf{C}_l^b \mathbf{v}^l$ ) is defined as follows, with Euler angles as attitude parameters and  $\mathbf{C}_i(\cdot)$  elementary rotation matrices defined in Equation (1.1.2).

$$\mathbf{C}_{l}^{b} = \mathbf{C}_{1}(r) \,\mathbf{C}_{2}(p) \,\mathbf{C}_{3}(y) \tag{1.1.13}$$

The parameterization of attitude by Euler angles (roll r, pitch p, and yaw y) exhibits a singularity at  $p = \pi/2$ , reason for which alternative ways of attitude representation such as quaternions are provided in Section 1.1.2.

Note that the axis around which roll angle is measured is the same as the vehicle roll axis  $(\mathbf{x}^b)$ , which is not the case for the other two Euler angles. The axis of pitch angle is the vehicle pitch axis  $(\mathbf{y}^b)$  rotated back by roll angle. Similarly, the axis of yaw angle is the vehicle yaw axis  $(\mathbf{z}^b)$  rotated back by roll angle and then pitch angle. This arises some confusions or criticisms with the widely accepted terminology, as the "rate of change of roll/pitch/yaw angle" is not generally equal to the "roll/pitch/yaw rate". One can investigate Equation (1.1.9) to see that the two are equal (for arbitrary values of  $\omega_i$ 's) only when roll and pitch angles are zero.

#### Wind Frame (w-frame)

Wind frame is not typically used in INS-based navigation, unless air data is also integrated within the system. However, since this frame will be used in the proposed navigation system, its definition is presented here.

The airflow around a vehicle (UAV) which is due to the composition of its velocity  $\mathbf{v}$  and wind velocity  $\mathbf{w}$ , is described by the airspeed vector  $\mathbf{V}$ . The wind frame has its origin coinciding with that of the body frame and its first axis in the direction of  $\mathbf{V}$ , and is defined by two angles with respect to body frame. These angles are called the angle of attack  $\alpha$  and the sideslip angle  $\beta$ , and are depicted in Figure 1.3. The following equation relates UAV's velocity  $\mathbf{v}$  and wind velocity  $\mathbf{w}$  via airspeed  $\mathbf{V}$  (see Figure 1.3).

$$\mathbf{v} = \mathbf{V} + \mathbf{w} \tag{1.1.14}$$

<sup>&</sup>lt;sup>1</sup>North-East-Down



Figure 1.3 – Body (b) and wind (w) frames with airspeed (**V**), wind velocity (**w**), UAV velocity (**v**), angle of attck ( $\alpha$ ), and sideslip angle ( $\beta$ )

This reads in body frame as

$$\mathbf{v}^{b} = \mathbf{V}^{b} + \mathbf{C}_{l}^{b} \mathbf{w}^{l} \quad \Leftrightarrow \quad \mathbf{V}^{b} = \mathbf{v}^{b} - \mathbf{C}_{l}^{b} \mathbf{w}^{l} \Leftrightarrow \begin{bmatrix} V_{x}^{b} \\ V_{y}^{b} \\ V_{z}^{b} \end{bmatrix} = \begin{bmatrix} v_{x}^{b} \\ v_{y}^{b} \\ v_{z}^{b} \end{bmatrix} - \mathbf{C}_{l}^{b} \begin{bmatrix} w_{x}^{l} \\ w_{y}^{l} \\ w_{z}^{l} \end{bmatrix}.$$
(1.1.15)

The rotation matrix to transform vectors from body frame to wind frame is defined as

$$\mathbf{C}_{b}^{w} = \mathbf{C}_{3}(\beta) \, \mathbf{C}_{2}^{T}(\alpha) = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}.$$
(1.1.16)

The airspeed magnitude V, the angle of attack  $\alpha$ , the sideslip angle  $\beta$ , and the dynamic pressure  $\bar{q}$  are defined as follows, where  $\rho$  denotes air density. The dynamic pressure will be used in expressions for aerodynamic forces and moments in Sections 4.1.1 and 4.1.2.

$$V = \sqrt{V_x^{b^2} + V_y^{b^2} + V_z^{b^2}}, \quad \bar{q} = \frac{\rho V^2}{2}, \quad \alpha = \arctan\left(\frac{V_z^b}{V_x^b}\right), \quad \beta = \arcsin\left(\frac{V_y^b}{V}\right) \quad (1.1.17)$$

#### Sensor Frame

Sensor frame is a rigidly attached to a sensor with its origin at the observations reference point, and its axes along principal directions of the sensor, if such directions are defined. For example, these principal axes for an IMU will be in the direction of single accelerometers/gyroscopes inside the sensor (assuming perfect orthogonality), along which the accelerations and rotation rates are expressed by the sensor.

As in general there can be non-orthogonality among the principal directions of a sensor that prevents defining a Cartesian frame based on those directions, the principal axes in definition of the sensor frame throughout this thesis are considered to be the ones after taking the non-orthogonality into account. As discussed in Section 3.5.1, raw data from sensors (IMU) are compensated for pre-calibrated non-orthogonality [and scale factors], and then they are assumed to be perfectly orthogonal and forming the sensor frame. As the pre-calibration is imperfect in general, there will be some residual errors, which are absorbed by the additive errors considered in observations. This modification of definition is made to avoid defining a separate reference frame for the sensor either before or after taking non-orthogonality into account, which would have equivalent result in navigation.

In case of some sensors, there is no principal direction, as far as navigation system is concerned. In case of a GNSS receiver, for example, the observations are only related to the phase center of the antenna, regardless of its orientation. In these cases, only the origin of the sensor frame matters and the direction of its axes are arbitrary or irrelevant.

Sensor frames for the sensors used in this thesis are presented in Section 3.4.5.

# 1.1.4 Inertial Navigation System (INS)

INS is composed of an IMU providing inertial measurements and a computer running a navigation algorithm to calculate navigation solution based on such measurements. In UAV applications, the strapdown mechanization is used, in which the inertial sensors are rigidly attached (strapped down) to the platform. This is as opposed to the gimbaled mechanization, in which the sensors are mounted on a platform that is mechanically isolated from rotational motion of the platform thanks to gimbals. In strapdown mechanization, all the inertial measurements are resolved in IMU sensor frame (after compensation for non-orthogonality), that is normally in a fixed position and orientation relation to the body frame. Unless the body frame coincides with the sensor frame, the misalignment between the two frames (sensor and body) needs to be determined and compensated for. The navigation equations are conventionally derived assuming that measurements are available in body frame (after applying necessary compensations). As INS is principally integrating the linear accelerations and rotation rates to obtain PVA solution, the measurements need to be expressed in navigation frame –the frame in which the navigation solution is to be determined.

Navigation equations can be derived and then solved in any reference frame, as long as everything is clearly defined and the equations are treated correctly. Applying the pertinent transformations, the results will be equivalent regardless of the choice of reference frames. Therefore, one needs to check which reference frames provide the most helpful representation of the navigation solution for the intended application. In this research, a set of frames have been chosen that are usually adopted in aerial navigation and are also beneficial for VDM-based navigation. Position is stated with respect to e-frame and is expressed in ellipsoidal coordinate system, as

$$\mathbf{r}_e^l = [\phi, \lambda, h]^T \,. \tag{1.1.18}$$

Velocity is also stated (measured) with respect to e-frame  $(\mathbf{v}^e)$ , but expressed in l-frame  $(\mathbf{v}^l_e)$  for more comprehensive representation. The relation between  $\mathbf{v}^l_e$  and  $\mathbf{v}^e$  is established via the following rotation.

$$\mathbf{v}_e^l = [v_N, v_E, v_D]^T = \mathbf{C}_e^l \mathbf{v}^e \tag{1.1.19}$$

#### **Integrated Navigation**

Finally, attitude is expressed with respect to l-frame and is represented by the quaternion  $\mathbf{q}_b^l$  for continuity and computation efficiency benefits. For results representation, though, Euler angles are used, as they provide a more comprehensible and intuitive representation of attitude. The INS navigation state vector is then defined as

$$\mathbf{x}^{l} = \begin{bmatrix} \mathbf{r}_{e}^{l} \\ \mathbf{v}_{e}^{l} \\ \mathbf{q}_{b}^{l} \end{bmatrix}$$
(1.1.20)

In order to obtain navigation equations (time-derivative of  $\mathbf{x}^{l}$ ), Newton's second law of motion for a rigid body in i-frame is first considered.

$$\ddot{\mathbf{r}}^i = \mathbf{C}^i_b \mathbf{f}^b + \breve{\mathbf{g}}^i(\mathbf{r}). \tag{1.1.21}$$

Substituting  $\mathbf{r}^i$  by  $\mathbf{r}^e = \mathbf{C}_i^e \mathbf{r}^i$  and considering Equation (1.1.7) when taking time derivative of  $\mathbf{C}_i^e$  gives Newton's second law of motion in e-frame.

$$\ddot{\mathbf{r}}^e = \mathbf{C}_b^e \mathbf{f}^b + \breve{\mathbf{g}}^e - 2\mathbf{\Omega}_{ie}^e \mathbf{v}^e - \mathbf{\Omega}_{ie}^e \mathbf{\Omega}_{ie}^e \mathbf{r}^e \tag{1.1.22}$$

Note that in deriving Equation (1.1.22), the angular velocity of Earth with respect to inertial frame ( $\Omega_{ie}^{e}$ ) is assumed to be a constant, which is quite acceptable for navigation purposes. As both the gravitational acceleration  $\mathbf{\breve{g}}^{e}$  and the centrifugal acceleration  $\Omega_{ie}^{e}\Omega_{ie}^{e}\mathbf{r}^{e}$  are only a function of position and the main part of both is in the same direction (along  $\mathbf{x}_{D}$  in l-frame), they are normally combined to form the local gravity vector  $\mathbf{g}^{e}$  as

$$\mathbf{g}^e = \breve{\mathbf{g}}^e - \mathbf{\Omega}^e_{ie} \mathbf{\Omega}^e_{ie} \mathbf{r}^e. \tag{1.1.23}$$

Denoting  $\dot{\mathbf{r}}^e$  by  $\mathbf{v}^e$  and using Equation (1.1.23) in Equation (1.1.22), velocity dynamics in e-frame can be written in a more compact form as

$$\dot{\mathbf{v}}^e = \mathbf{C}_b^e \mathbf{f}^b + \mathbf{g}^e - 2\mathbf{\Omega}_{ie}^e \mathbf{v}^e. \tag{1.1.24}$$

Rewriting Equation (1.1.24) for  $\mathbf{v}_e^l$  as defined in Equation (1.1.19) and considering Equation (1.1.7) when taking time derivative of  $\mathbf{C}_e^l$  gives velocity dynamics equation in l-frame as

$$\dot{\mathbf{v}}_{e}^{l} = \mathbf{C}_{b}^{l}\mathbf{f}^{b} + \mathbf{g}^{l} - \left(\mathbf{\Omega}_{el}^{l} + 2\mathbf{\Omega}_{ie}^{l}\right)\mathbf{v}_{e}^{l},\tag{1.1.25}$$

where the angular velocity vectors  $\boldsymbol{\omega}_{ie}^{l}$  (angular velocity of Earth expressed in local frame) and  $\boldsymbol{\omega}_{el}^{l}$  (local frame transport rate) are defined as follows, with  $\boldsymbol{\omega}_{ie} = 7.292115 \times 10^{-5} rad/s$ being magnitude of angular velocity of the Earth [59]. The rotation matrix  $\mathbf{C}_{b}^{l}$  is calculated from the quaternion  $\mathbf{q}_{b}^{l}$  using Equation (1.1.6).

$$\boldsymbol{\omega}_{ie}^{l} = \begin{bmatrix} \omega_{ie} \cos \phi & 0 & -\omega_{ie} \sin \phi \end{bmatrix}^{T}$$
(1.1.26)

$$\boldsymbol{\omega}_{el}^{l} = \begin{bmatrix} \dot{\lambda}\cos\phi & -\dot{\phi} & \dot{\lambda}\sin\phi \end{bmatrix}^{T}$$
(1.1.27)

A gravity model for the ellipsoidal local frame can be used to evaluate the gravity vector  $\mathbf{g}^{l}$  in local frame, for example as proposed by WGS-84 model [59] and used in this research.

Time derivative of position vector  $\mathbf{r}_{e}^{l}$  can be expressed as

$$\dot{\mathbf{r}}_e^l = \mathbf{D}^{-1} \mathbf{v}_e^l, \tag{1.1.28}$$

where the matrix  $\mathbf{D}^{-1}$  matrix is defined as

$$\mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{R_M + h} & 0 & 0\\ 0 & \frac{1}{(R_P + h)cos\phi} & 0\\ 0 & 0 & -1 \end{bmatrix},$$
(1.1.29)

to convert velocity from Cartesian to ellipsoidal coordinates. The quantities  $R_M = \frac{a(1-e^2)}{(1-e^2\sin^2(\varphi))^{3/2}}$  and  $R_P = \frac{a}{(1-e^2\sin^2(\varphi))^{1/2}}$  represent meridian radius of curvature and prime vertical radius of curvature, respectively, with *a* being the semi-major axis of the reference ellipsoid and *e* being its eccentricity [59].

According to Equation (1.1.10), time derivative of  $\mathbf{q}_b^l$  as the attitude representative can be expressed as following.

$$\dot{\mathbf{q}}_{b}^{l} = \frac{1}{2} \mathbf{q}_{b}^{l} \otimes \left[\boldsymbol{\omega}_{lb}^{b}\right]_{q} = \frac{1}{2} \mathbf{q}_{b}^{l} \otimes \left[\boldsymbol{\omega}_{ib}^{b} - \left(\mathbf{C}_{b}^{l}\right)^{T} \left(\boldsymbol{\omega}_{ie}^{l} + \boldsymbol{\omega}_{el}^{l}\right)\right]_{q}$$
(1.1.30)

Finally, Equations (1.1.28), (1.1.25), and (1.1.30) can be collected to form the time derivative of INS navigation state vector<sup>1</sup>  $\mathbf{x}^{l}$  as following.

$$\dot{\mathbf{x}}^{l} = \begin{bmatrix} \dot{\mathbf{r}}_{e}^{l} \\ \dot{\mathbf{v}}_{e}^{l} \\ \dot{\mathbf{q}}_{b}^{l} \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-1}\mathbf{v}_{e}^{l} \\ \mathbf{C}_{b}^{l}\mathbf{f}^{b} - \left(2\mathbf{\Omega}_{ie}^{l} + \mathbf{\Omega}_{el}^{l}\right)\mathbf{v}_{e}^{l} + \mathbf{g}^{l} \\ \frac{1}{2}\mathbf{q}_{b}^{l} \otimes \left[\boldsymbol{\omega}_{ib}^{b} - \left(\mathbf{C}_{b}^{l}\right)^{T}\left(\boldsymbol{\omega}_{ie}^{l} + \boldsymbol{\omega}_{el}^{l}\right)\right]_{q} \end{bmatrix}$$
(1.1.31)

This is the final form of INS navigation equations in l-frame adopted in this research. As mentioned before, INS navigation equations can take different –yet equivalent– forms according to choices made on coordinate systems, reference frames, and attitude representations.

# 1.1.5 Global Navigation Satellite System (GNSS)

GNSS is usually the minimum –and the most important– aiding system in INS-based navigation for outdoor applications. GNSS provides range related observations per satellite of the constellation, usually converted to higher level measurements on absolute PVT solution for terrestrial and aerial –and even space– applications, based on triangulation using the observed pseudo-distances to satellites. Currently, there are four fully or partially operational constellations [5], namely the American GPS<sup>2</sup>, the Russian GLONASS<sup>3</sup>, the European global

<sup>&</sup>lt;sup>1</sup>Rigorously defined, states are independent, while the four components of the quaternion  $\mathbf{q}_b^l$  are not, as there is the unity constraint ( $||\mathbf{q}_b^l|| = 1$ ). A rigorous way to resolve this issue is replacing the four quaternion states by three error states for attitude [71]. However, for the sake of implementation simplicity, especially the automated linearization as described in Section 4.3.3, the four quaternion states are kept in the state vector in this research, with the unity of quaternions being preserved externally.

<sup>&</sup>lt;sup>2</sup>Global Positioning System

 $<sup>^3{\</sup>rm GLObal}$ NAvigation Satellite System (read as "GLObalnaya NAvigatsionnaya Sputnikovaya Sistema" in Russian)

satellite-based navigation system (Galileo), and the Chinese BeiDou-2 navigation satellite system. There are also three regional constellations, namely the Chinese BeiDou-1 navigation satellite system, the Indian NAVIC<sup>1</sup>, and the Japanese QZSS<sup>2</sup>.

Although the principle seems simple and straightforward, algorithmic implementation can be very complex due to the many systematic and stochastic effects that have to be accounted for, such as relativistic effects, timing errors on satellite and receiver clocks, atmospheric effects on signal propagation, multi-path phenomena in signal reception, and low signal power. Today, there are still ongoing research activities on hardware and software aspects to get the most out of available signals. Furthermore, the raw observations from GNSS receivers can be processed in many different ways to get the position/velocity data, from single point positioning to multi-constellation multi-frequency carrier phase differential methods [5]. Positioning accuracy can vary from several meters to a few centimeters for kinematic applications.

The level of integration with INS can vary from loosely to tightly coupled schemes, depending on the internal loops between the two systems. The simplest and most common integration scheme is loosely coupling, in which each system provide their independent navigation solutions (PVA for INS and PVT for GNSS) that are then fused together to estimate the final solution. This is the coupling scheme used in this research. More details on estimation methods is provided in Section 1.2.

# **1.2** Estimation Methods

Integrated navigation is essentially an estimation problem to find navigation states (and perhaps other augmented states) based on fusion of data/information available from sensors and models. This section provides brief explanation on some estimation methods commonly used in navigation. Most wordings and formulations are borrowed from [23] with adjustments in notation.

Estimation is defined in [23] as the "process of extracting information from data –data which can be used to infer the desired information and may contain error." This definition may get further clarified and detailed via definition of an optimal estimator as "a computational algorithm that processes measurements to deduce a minimum error –in accordance with some stated criterion of optimality– estimate of the state of a system by utilizing: knowledge of system and measurement dynamics, assumed statistics of system noise and measurement errors, and initial condition information" [23]. From temporal point of view, there are three types of estimation problems for dynamic systems, namely prediction, filtering, and smoothing, as schematically depicted in Figure 1.4.

When the time at which the estimation is required is outside the time span of available measurement/observation<sup>3</sup> data, the estimation problem is referred to as prediction. In this case, the state of the system is to be estimated for a future time with respect to the last available measurement. When the time at which the estimation is required coincides with the last point of the time span of available measurement data, the estimation problem is referred

<sup>&</sup>lt;sup>1</sup>NAVigation with Indian Constellation

<sup>&</sup>lt;sup>2</sup>Quasi-Zenith Satellite System

<sup>&</sup>lt;sup>3</sup>Throughout this thesis, "measurement" and "observation" are used interchangeably.



Figure 1.4 – Three types of estimation problems (adapted from [23])

to as filtering. In this case, the state of the system is to be estimated for the current time with respect to the last available measurement. When the time at which the estimation is required is inside the time span of available measurement data, the estimation problem is referred to as smoothing. In this case, the state of the system is to be estimated for a time with available measurements before and after that [23].

Even today after more than half a century since it was first introduced [35], Kalman filter is probably the most well known and frequently used filtering algorithm in many fields including aerial navigation. In the rest of this section, the original (linear) Kalman filter and some extended versions of it for nonlinear systems are briefly introduced.

# 1.2.1 Linear Kalman Filter (KF)

Kalman filter was originally developed to solve filtering problem in linear systems. It is the optimal estimator in such systems, minimizing a weighted scalar sum of diagonal elements of the covariance matrix of the estimation error after a measurement is applied, or equivalently, the length of the estimation error vector [23].

There are several formulations, depending on whether the system dynamics/process model<sup>1</sup> and observation equation are available/expressed in continuous or discrete form. In a common situation in engineering problems, the system dynamic is expressed in continuous form, while the measurements are available in discrete domain, and estimation is also done discretely with k denoting the current step (index) in time.

$$\dot{\mathbf{X}} = \mathbf{F}(t)\mathbf{X} + \mathbf{L}(t)\mathbf{U} + \mathbf{G}(t)\mathbf{w}$$
(1.2.1)

$$\mathbf{Z}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{n}_k \tag{1.2.2}$$

The states are denoted by  $\mathbf{X}$ , the observations by  $\mathbf{Z}$ , the deterministic input by  $\mathbf{U}$ . System dynamics noise or process noise denoted by  $\mathbf{w}$  and observation noise denoted by  $\mathbf{n}$  are both

<sup>&</sup>lt;sup>1</sup>Throughout this thesis, "system dynamics" and "process model" are used interchangeably.

supposed to follow a Gaussian distribution. The matrices  $\mathbf{F}$ ,  $\mathbf{L}$ ,  $\mathbf{G}$ , and  $\mathbf{H}$  are usually called system dynamics matrix, input matrix, process noise shaping matrix, and observation matrix.

Kalman filter has two separate phases. In prediction phase, the system dynamics is used to obtain the estimation of the state vector at current step (k) based on all the information available up to the previous step (k-1) denoted as  $\hat{\mathbf{X}}_{k|k-1}$  and the associated covariance matrix  $\hat{\mathbf{P}}_{k|k-1}$  at current step. In discrete domain, the prediction phase can be formulated as following.

$$\hat{\mathbf{X}}_{k|k-1} = \mathbf{\Phi}(t_k, t_{k-1})\hat{\mathbf{X}}_{k-1|k-1} + \int_{t_{k-1}}^{t_k} \mathbf{\Phi}(t_k, \tau) \mathbf{L}(\tau) \mathbf{U}(\tau) d\tau$$
(1.2.3)

$$\hat{\mathbf{P}}_{k|k-1} = \mathbf{\Phi}(t_k, t_{k-1})\hat{\mathbf{P}}_{k-1|k-1}\mathbf{\Phi}^T(t_k, t_{k-1}) + \mathbf{Q}_{k,k-1}$$
(1.2.4)

Here,  $\mathbf{\Phi}(t_2, t_1)$  is the transition matrix, calculated as following.

$$\mathbf{\Phi}(t_2, t_1) = e^{\int_{t_1}^{t_2} \mathbf{F}(t)dt}$$
(1.2.5)

The discrete covariance matrix  $\mathbf{Q}_{k,k-1}$  accounts for system dynamic uncertainty, calculated as

$$\mathbf{Q}_{k,k-1} = \int_{t_{k-1}}^{t_k} \mathbf{\Phi}_{t_k,\tau} \mathbf{G}(\tau) \mathbf{Q}(\tau) \mathbf{G}^T(\tau) \mathbf{\Phi}_{t_k,\tau}^T d\tau, \qquad (1.2.6)$$

where  $\mathbf{Q}(t)$  is the spectral density matrix for system dynamic noise vector  $\mathbf{w}$ . When  $t_k - t_{k-1}$  is sufficiently small (depending on the desired precision), a first order approximation can be used for  $\mathbf{Q}_{k,k-1}$ .

$$\mathbf{Q}_{k,k-1} \approx \mathbf{G}(t_k) \mathbf{Q}(t_k) \mathbf{G}^T(t_k) \left(t_{k-1} - t_k\right)$$
(1.2.7)

In the update phase, if available, the measurements  $\mathbf{Z}_k$  and their associated covariance matrix  $\mathbf{R}_K$  are used to obtain the estimation of the state vector at current step (k) based on all the information available up to the current step (k) denoted as  $\hat{\mathbf{X}}_{k|k}$  and the associated covariance matrix  $\hat{\mathbf{P}}_{k|k}$  at the current step. In order to do so, the gain matrix needs to be calculated first, as

$$\mathbf{K}_{k} = \hat{\mathbf{P}}_{k|k-1} \mathbf{H}_{k}^{T} \left( \mathbf{H}_{k} \hat{\mathbf{P}}_{k|k-1} \mathbf{H}_{k}^{T} + \mathbf{R}_{K} \right)^{-1}, \qquad (1.2.8)$$

where  $\mathbf{R}_{K}$  is the covariance matrix of measurement noise vector  $\mathbf{r}_{k}$ . Then the update phase can be performed as,

$$\hat{\mathbf{X}}_{k|k} = \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k \left( \mathbf{Z}_k - \mathbf{H}_k \hat{\mathbf{X}}_{k|k-1} \right)$$
(1.2.9)

$$\hat{\mathbf{P}}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \,\hat{\mathbf{P}}_{k|k-1} \tag{1.2.10}$$

where  $\mathbf{I}$  denotes the identity matrix [23].

# 1.2.2 Extended Kalman Filter (EKF)

In many cases, such as navigation, the system dynamic and the observation equations are nonlinear, as following.

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}, \mathbf{U}, t) + \mathbf{G}(t)\mathbf{w}$$
(1.2.11)

$$\mathbf{Z}_k = \mathbf{h}_k(\mathbf{X}_k) + \mathbf{n}_k \tag{1.2.12}$$

EKF is one of the extensions of Kalman filter to deal with such nonlinearities. In this method,  $\mathbf{f}(\mathbf{X}, \mathbf{U})$  and  $\mathbf{h}_k(\mathbf{X}_k)$  functions are linearized as following.

$$\mathbf{f}(\mathbf{X}, \mathbf{U}, t) \approx \mathbf{f}(\hat{\mathbf{X}}, \mathbf{U}, t) + \left. \frac{\partial \mathbf{f}(\mathbf{X}, \mathbf{U}, t)}{\partial \mathbf{X}} \right|_{\mathbf{X} = \hat{\mathbf{X}}, \ \mathbf{U} = \mathbf{U}(t)} \left( \mathbf{X} - \hat{\mathbf{X}} \right)$$
(1.2.13)

$$\mathbf{h}_{k}(\mathbf{X}_{k}) \approx \mathbf{h}_{k}(\hat{\mathbf{X}}) + \left. \frac{\partial \mathbf{h}(\mathbf{X})}{\partial \mathbf{X}} \right|_{\mathbf{X} = \hat{\mathbf{X}}} \left( \mathbf{X} - \hat{\mathbf{X}} \right)$$
(1.2.14)

The nonlinear part of the original system dynamics can be expressed as following.

$$\hat{\mathbf{X}} = \mathbf{f}(\hat{\mathbf{X}}, \mathbf{U}, t) \tag{1.2.15}$$

Now, two new variables  $\Delta \mathbf{X}$  and  $\Delta \mathbf{Z}$  are introduced, and  $\mathbf{F}$  and  $\mathbf{H}$  matrices are defined as following.

$$\Delta \mathbf{X} = \mathbf{X} - \hat{\mathbf{X}} \tag{1.2.16}$$

$$\Delta \mathbf{Z} = \mathbf{Z} - \mathbf{h}(\hat{\mathbf{X}}) \tag{1.2.17}$$

$$\mathbf{F}(t) \triangleq \left. \frac{\partial \mathbf{f}(\mathbf{X}, \mathbf{U}, t)}{\partial \mathbf{X}} \right|_{\mathbf{X} = \hat{\mathbf{X}}, \ \mathbf{U} = \mathbf{U}(t)}$$
(1.2.18)

$$\mathbf{H} \triangleq \left. \frac{\partial \mathbf{h}(\mathbf{X})}{\partial \mathbf{X}} \right|_{\mathbf{X} = \hat{\mathbf{X}}} \tag{1.2.19}$$

Then, the remaining part of the system dynamics and the observation model form a new linear system as following.

$$\Delta \dot{\mathbf{X}} = \mathbf{F}(t)\Delta \mathbf{X} + \mathbf{G}(t)\mathbf{w} \tag{1.2.20}$$

$$\Delta \mathbf{Z}_k = \mathbf{H}_k(\Delta \mathbf{X}_k) + \mathbf{n}_k \tag{1.2.21}$$

In prediction phase, the nonlinear part (Equation (1.2.15)) is solved in time to obtain  $\hat{\mathbf{X}}_{k|k-1}$ . Then the update phase of a linear Kalman filter is applied to the linear system of Equation (1.2.20) and Equation (1.2.21) to calculate  $\Delta \hat{\mathbf{X}}_{k|k}$  assuming  $\Delta \hat{\mathbf{X}}_{k|k-1} = \mathbf{0}$ . The updated state vector can now be calculated as following.

$$\hat{\mathbf{X}}_{k|k} = \hat{\mathbf{X}}_{k|k-1} + \Delta \hat{\mathbf{X}}_{k|k} \tag{1.2.22}$$

The propagation of covariance matrix is performed in the same manner as in linear Kalman filter.

# 1.2.3 Unscented Kalman Filter (UKF)

In EKF, covariance matrix is propagated via an exact method, but applied to a linearized version of system dynamic, which is accurate only to first order. In case of high nonlinearities, this may cause high errors and even unstability. UKF on the other hand, features a deterministic sampling approach to generate a set of points that can capture the mean of the states, as well as the covariance matrix to the second order. These points are then propagated using the original [nonlinear] system dynamics, and again collocated to estimate new mean and covariance. Therefore, the way to represent covariance information (using the spread of a set of points) is not exact. However, the propagation of such representation occurs via nonlinear version of system dynamics with no approximations. This can be seen as opposed to EKF, in which the way to represent covariance information is exact, whereas its propagation is performed via the linearized model accepting an approximation.

In many cases, UKF has been reported to produce better results, when the system is highly nonlinear in the time scale of propagation [78]. However, better performance of UKF over EKF is not always the case [62]. When the duration of propagation at each step (or the sampling frequency) is short enough and no extraordinary nonlinearity is expected in the system, EKF may be the preferred choice due to lower computational cost. This is probably the reason why EKF is still the dominant filtering method in navigation. It is also the preferred choice for VDM-based navigation in this research. More details on definition and formulation on UKF, as well as ready to implement explanations can be found in [70].

# 1.3 Sensor Error Modeling

Measurements from real sensors always have errors, at least when it comes to continuous quantities. In this section, a very brief overview of common types of errors encountered in INS-based navigation and some modeling/estimation techniques is presented.

# **1.3.1** Systematic and Stochastic Errors

The errors in a sensor data can be divided into two categories of systematic and stochastic errors. Systematic errors are the ones with predictable behavior and known cause, and are often constant or changing very slowly. An example is nonorthogonality in IMU data, caused by physical deviation of sensors triad from mutual orthogonality to some extent. Systematic errors can usually be measured directly or estimated in a calibration procedure. In such cases, the errors can be directly compensated. However, this calibration may have not been done for any sensor used in a setup, or there may be some residual to such errors. In such cases, these errors can be modeled and be a part of augmented states in the navigation filter to be estimated/refined.

Stochastic (or random) errors on the other hand, are the ones with unknown causes or very complicated models, which are often randomly changing over time. These can be further categorized into time-correlated and time-uncorrelated errors. Time-correlated errors such as random biases can be part of augmented states in the navigation filter and estimated each time the navigation is performed. Time-uncorrelated errors, often assumed to be white Gaussian noise, cannot be estimated by nature and are only introduced to the filter as observation uncertainty. In any case, performance of filtering depends to a great extent on relevant modeling of stochastic errors.

## **1.3.2** Stochastic Processes

Four types of stochastic processes that are frequently used in navigation are white noise, random bias, random walk, and first order Gauss-Markov process. These are introduced in this section, with most wordings and equations borrowed from [71]. More details and formal definitions can be found in [71].

## White Noise

Continuous white noise, denoted by W(t), is a stationary process with equal intensity at different frequencies, resulting in a constant PSD<sup>1</sup>. Although a white noise process can have any probability distribution, the Gaussian distribution is often assumed for time-uncorrelated errors, making it a white Gaussian noise.

The equivalent discrete-time white noise process  $\{W_k : k \in \mathbb{Z}\}$  is defined as the consecutive averages of W(t) over small periods of time  $(\Delta t_k = t_k - t_{k-1})$ . A discrete white Gaussian (normally distributed) noise process  $W_k$  with zero mean and variance  $\sigma_{WN}^2$  is denoted by  $W_k \sim \mathcal{N}(0, \sigma_{WN}^2)$ .

# Random Constant (Bias)

A random constant or a random bias, as the name suggests, is a process with constant yet randomly determined value over time. Therefore, the governing equation in continuous domain has the form

$$\dot{X}(t) = 0,$$
 (1.3.1)

and the following form in discrete domain.

 $X_{k+1} = X_k \tag{1.3.2}$ 

The initial condition for this process is often assumed to be follow a Gaussian (normal) distribution with variance  $\sigma_{WN}^2$ .

# Random Walk

A random walk process is the output of integrating a Gaussian white noise in time, with the following governing equation in continuous domain.

$$X(t) = W(t) \quad \text{for} \quad t \in \mathbb{R}.$$
(1.3.3)

<sup>&</sup>lt;sup>1</sup>Power Spectral Density

#### **Integrated Navigation**

Integrating this equation yields

$$X(t) = \int_{t_0}^t W(s)ds$$
 (1.3.4)

with initial condition  $X(t_0)$ , often assumed to be zero. The random walk process is not stationary.

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

$$X_{k+1} = X_k + W_k, (1.3.5)$$

with  $W_k \sim \mathcal{N}(0, \sigma_{RW}^2)$ .

## First Order Gauss-Markov Process

A first order Gauss-Markov process has a first order delay model driven by Gaussian white noise W(t). This is equivalent to a first order autoregressive model. The governing equation in continuous time is as following.

$$X(t) = -\beta_{GM}X(t) + W(t) \quad \text{for} \quad t \in \mathbb{R},$$
(1.3.6)

where  $q = 2\sigma_{GM}^2 \beta_{GM}$  with  $\beta_{GM} \in \mathbb{R}^+$  assumed constant in time.

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

$$X_{k+1} = e^{-\beta_{GM}\Delta t} X_k + W_k, (1.3.7)$$

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

$$q_k = \sigma_{GM}^2 \left[ 1 - e^{-2\beta_{GM}\Delta t} \right]. \tag{1.3.8}$$

#### 1.3.3 Stochastic Modeling

Consider the sequence  $\{y_k : k = 1, ..., N\}$  representing the observed one-dimensional error signal of a sensor. This sequence can be seen as a realization of an univariate Gaussian time series  $\{Y_k : k \in \mathbb{Z}\}$  to which the conditional distribution  $F(Y_k|Y_{k-1}, Y_{k-2}, ..., \theta)$  with parameters  $\theta$  is associated.

Stochastic modeling consists of two main tasks. The first one called model building refers to determining the types of processes observed within a signal to be modeled, denoted by  $F(\cdot)$ . The second task called model estimation refers of determining the parameters of those models, denoted by  $\theta$ .

In this section, some methods for stochastic modeling are briefly presented. First, some of frequently used traditional methods will be presented, and then a recent and very powerful method called GMWM<sup>1</sup> [29] will be presented, as the method used in this research for stochastic modeling of sensor errors.

<sup>&</sup>lt;sup>1</sup>Generalized Method of Wavelet Moments

# 1.3.4 Traditional Methods

Three of the most frequently used traditional methods for stochastic modeling are mentioned here.

#### Signal Autocorrelation Method

Computing the ACF<sup>1</sup> of the observed signal  $\{y_k : k = 1, ..., N\}$  may reveal the presence or the absence of [time-]correlated noise. This modeling technique assumes the noise to be composed of an uncorrelated and a correlated part. The procedure, described with a detailed example in [81], consists of low-pass filtering to remove the uncorrelated part, and then fitting a model to the ACF of the remaining correlated filter output.

#### Allan Variance Method

In 1998, the IEEE<sup>2</sup> standard put forward this technique as a noise identification method to determine the characteristics of the underlying random processes that perturb data [72].

Let  $\overline{Y}_k(\tau)$  be the sample average of  $\tau$  consecutive observations.

$$\bar{Y}_k(\tau) = \frac{1}{\tau} \sum_{j=0}^{\tau-1} Y_{k-j}$$
(1.3.9)

The Allan variance at scale  $\tau$ , noted as  $\sigma_{\bar{Y}}^2(\tau)$ , 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 non-overlapping  $\bar{Y}(\tau)$ :

$$\sigma_{\bar{Y}}^2(\tau) = \frac{1}{2} \mathbb{E}\left[ \left( \bar{Y}_k(\tau) - \bar{Y}_{k-\tau}(\tau) \right)^2 \right].$$
(1.3.10)

Several estimators of the Allan variance, noted  $\hat{\sigma}_{\bar{y}}^2(\tau)$ , have been proposed. One of the most efficient ones has been proposed in [25] and can be computed from a realization  $\{y_k : k = 1, \ldots, N\}$  using

$$\hat{\sigma}_{\bar{y}}^2(\tau) = \frac{1}{2(N-2\tau+1)} \sum_{k=2\tau}^N \left( \bar{y}_k(\tau) - \bar{y}_{k-\tau}(\tau) \right)^2.$$
(1.3.11)

In general, only basic processes are considered with the Allan variance, such as 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 [28].

<sup>&</sup>lt;sup>1</sup>AutoCorrelation Function

<sup>&</sup>lt;sup>2</sup>Institute of Electrical and Electronics Engineers

# Power Spectral Density (PSD) Method

The PSD analysis is a convenient method for analyzing and characterizing signals, and for stochastic modeling. It is especially well suited for analyzing periodic or aperiodic signals. 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 log-log PSD plots.

# 1.3.5 Limitations of Traditional Stochastic Modeling Methods

## Autocorrelation Method

When several processes are superposed, the ACF does not always enable to clearly separate them. The estimation of the model parameters becomes challenging or even impossible in such cases. As an example, the estimation task for a signal consisting the sum of a white noise and a first order Gauss-Markov process is non-trivial, despite its apparent simplicity. 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.

# **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 [28]. The Allan variance (and PSD) method works reasonably well only for processes which are clearly identifiable and separable in the spectral domain. However, it does not allow to directly read out the parameters of a Gauss-Markov process as large values of  $\beta$  make this process similar to Gaussian white noise, while small values of  $\beta$  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.

# 1.3.6 Generalized Method of Wavelet Moments (GMWM)

GMWM is a recently introduced method for stochastic modeling [29], with many advantages over traditional methods, from numerical stability and computational inefficiency to statistical consistency. GMWM makes use of the wavelet variance estimate the parameter vector  $\boldsymbol{\theta}$  of the complex models  $F(\cdot)$  that underlie the observed error. A software was developed recently in statistical tool R as a calibration platform based on GMWM [12]. Using GMWM estimator properties, this platform provides some very useful tools for sensor error modeling. For example, it provides confidence intervals for the parameters, and goodness of fit test to determine if the selected model with estimated parameters fit well the observed data. It also provides a criterion to assess the prediction error of the estimated model. Furthermore, it allows to rank models of different structures according to an index that penalizes model complexity and thus over-fitting. In this research, this platform is used for stochastic modeling of IMU errors. Also, an online version of the tool is available under ggmwm.smac-group.com.

# Chapter 2

# Basis for Dynamic Modeling of UAVs

# Overview

Motion dynamics for a UAV can be described by a model obtained based on the principles of classical mechanics for rigid bodies and flight aerodynamics to predict forces and moments acting on a UAV during its flight. This chapter provides a brief presentation of the bases for dynamic modeling of UAVs employed in this research.

# 2.1 Classical Mechanics

Classical mechanics is one of the oldest branches of science, studying motion of objects involving not very small sizes and not very high velocities, which lie in the realms of quantum mechanics and relativistic mechanics, respectively. The three main formulations for classical mechanics, leading to equivalent results, are mentioned in the following.

# 2.1.1 Newtonian-Euler Formulation

Famous Newton's laws of motion were formulated for point masses, to which only translational motions were associated. For a rigid body –as a continuous distribution of mass with fixed distances between any two points in it– there can also be rotational motions. Based on Newton's laws of motion, Euler formulated equation's of motion for rigid bodies. For a rigid body with mass m under resultant force  $\Sigma \mathbf{F}$ , denoting linear/translational acceleration of its center of mass by  $\mathbf{a}_{CM}$ , the first Newton-Euler equation, or Newton's second law for translational motions, is formulated as following.

$$\Sigma \mathbf{F} = m \mathbf{a}_{CM} \tag{2.1.1}$$

The second Newton-Euler equation, or Newton's second law for rotational motions, is formulated as

$$\Sigma \mathbf{M}_{CM} = \mathbf{I}_{CM} \boldsymbol{\alpha} + \boldsymbol{\omega} \times \mathbf{I}_{CM} \boldsymbol{\omega}, \qquad (2.1.2)$$

where  $\Sigma \mathbf{M}_{CM}$  is the resultant moments applied to the rigid body about its center of mass,  $\boldsymbol{\omega}$  is its angular velocity, and  $\boldsymbol{\alpha}$  denotes its angular acceleration. The moment of inertia matrix for the rigid body around its center of mass is denoted by  $\mathbf{I}_{CM}$ , whose elements are defined as [73]

$$I_{ij} \triangleq \int_{V} \rho(\mathbf{r}) \left( \delta_{ij} \sum_{k=1}^{3} (x_k^2) - x_i x_j \right) dV, \qquad (2.1.3)$$

with V being the volume of the rigid body,  $\rho(\mathbf{r})$  being mass density as a function of position of position vector  $\mathbf{r}$  –from center of mass to any point–,  $x_i$  being the  $i^{th}$  element of position vector, and  $\delta_{ij}$  denoting Kronecker delta valued 1 for i = j and 0 otherwise.

#### 2.1.2 Lagrangian Formulation

Newton-Euler formulation requires knowledge of all forces, including the ones related to constraints applied on the body and also the conservative forces, i.e., the ones derived from a potential filed whose work on the body in displacement from one point to the other does not depend on the path. Lagrangian formulation, on the other hand, allows to obtain equations of motion without explicitly expressing these forces.

Another very useful property of this formulation is working with generalized coordinates. Generalized coordinates are the minimum number of parameters required to uniquely define the configuration of the body. If there are holonomic constraints applied to the motion, generalized coordinates implicitly contain those constraints and make the analysis easier due to reduction in number of coordinates/dimensions (by the numbers of the constraints) and eliminating the need to apply those constraints separately. Even in absence of holonomic constraints, specific motions may be expressed more conveniently in coordinates other than Cartesian, which can be handled using generalized coordinates in Lagrangian formulation usually in an easier manner compared to Newton-Euler formulation.

If the generalized coordinates are denoted by  $q_i$ , Lagrangian (L) of the system (which may consist of more than one body) is defined as the difference between the kinetic energy (T) and the potential energy (U) of the system.

$$L \triangleq T - V \tag{2.1.4}$$

The equations of motion for a system with N generalized coordinates is then states as [24]

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \quad \text{(for } i = 1, 2, \dots, N\text{)},$$
(2.1.5)

where  $Q_i$  represents the generalized force not arising from a potential field.

# 2.1.3 Hamiltonian Formulation

Hamiltonian formulation is another formulation of classical mechanics. In this approach, for a system with N degrees of freedom, the states of are chosen to be N generalized coordinates  $q_i$  as in Lagrangian formulation, and N generalized momenta  $p_i$ , defined as

$$p_i \triangleq \frac{\partial L}{\partial q_i}.\tag{2.1.6}$$

The Hamiltonian of the system is defined as

$$H(q, p, t) \triangleq \sum_{i=1}^{N} (p_i \dot{q}_i) - L(q, \dot{q}, t),$$
(2.1.7)

which often, but not always, corresponds to the total energy of the system T + V. The Hamiltonian equations of motion are represented as following [6].

$$\dot{q}_i = \frac{\partial H}{\partial p_i} \tag{2.1.8}$$

$$\dot{p}_i = -\frac{\partial H}{\partial q_i} + Q_i \tag{2.1.9}$$

While the equations of motion in Newton-Euler and Lagrangian formulations are represented by N second order differential equations for a system with N degrees of freedom, Hamiltonian formulation results directly in 2N first order equations for the 2N states of the system.

In specific problems, Hamiltonian formulation has some implementation and representation advantages over Lagrangian formulation. However, "the Hamiltonian methods are not particularly superior to Lagrangian techniques for the direct solution of mechanical problems. Rather, the usefulness of the Hamiltonian viewpoint lies in providing a framework for theoretical extensions in many areas of physics. Within classical mechanics it forms the basis for further developments, such as Hamilton-Jacobi theory, perturbation approaches and chaos. Outside classical mechanics, the Hamiltonian formulation provides much of the language with which present-day statistical mechanics and quantum mechanics is constructed" [24].

# 2.2 Flight Aerodynamics

Aerodynamic forces and moments are the ones applied on a body immersed in a gas (usually air) due to interactions between the body and the surrounding gas in relative motion with respect to one another. Modeling these forces and moments is very challenging in general. Theoretical approaches to model them, analyze the air (gas) flow around the body, allowing to determine forces and moments applied on it in the boundaries with the body. According to Newton's third law of motion, the forces and moments applied on the body will be the equal and oppositely directed reactions to those.

The most general governing equation for fluid dynamics often considered in engineering applications is the Navier-Stocks equation. This equation represents application of classical mechanics to a fluid flow field, assuming a linear constitutive relation between stress tensor and rate of strain tensor and constant viscosity parameters  $\eta$  and  $\zeta$  for the fluid [45].

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \eta \nabla^2 \mathbf{v} + \left(\zeta + \frac{1}{3}\eta\right) \nabla (\nabla \cdot \mathbf{v})$$
(2.2.1)

In this equation,  $\rho$ , p, and **v** denote density, pressure, and velocity fields, respectively. The "material derivative"  $\frac{D}{Dt}$  is defined as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla, \qquad (2.2.2)$$

with  $\nabla$  being the gradient operator, and  $\nabla^2$  being the Laplace operator defined as  $\nabla^2 = \nabla \cdot \nabla$ . In Cartesian coordinates,  $\nabla$  is defined as

$$\nabla = \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z},\tag{2.2.3}$$

with i, j, and k being the unit vectors in the directions of x, y, and z axes, respectively.

Analytical solution to Navier-Stocks equation is possible only in very specific situations. In fact, even very fundamental aspects such as existence and uniqueness of solution are still subjects of open questions. This has been recognized important enough by CMI<sup>1</sup> to state it in the year 2000 as one of the seven "Millennium Problems" with a one-million-dollar prize, only one of which<sup>2</sup> solved to date. "The challenge is to make substantial progress toward a mathematical theory which will unlock the secrets hidden in the Navier-Stokes equations<sup>3</sup>"[33].

In most cases, this equation, or a simplified version of it (ignoring viscosity or two dimensional modeling, for example) is solved via the methods and tools provided by  $CFD^4$  [31]. Experimental derivation of aerodynamic forces and moments is normally performed via wind tunnel tests, performing of which can be very expensive in terms of time and cost, especially for larger bodies.

The model for aerodynamic forces and moments used in the dynamic model of the aircraft is usually represented by a set of equation in polynomial form, relating those forces and moments to dynamic states (such as velocity) and control commands (such as aileron deflection) via some coefficients. These coefficients are specific to the aircraft and usually given via extensive lookup tables obtained by CFD analyses or wind tunnel tests or a combination of both. Depending on the fidelity, accuracy, and simplicity requirements on the final dynamic model of the aircraft, these coefficients may be considered as constants over operational conditions (excluding stall conditions, for example). Extensive details on flight aerodynamics can be found in many references, a commonly referenced one of which being [64].

<sup>&</sup>lt;sup>1</sup>Clay Mathematics Institute

 $<sup>^2 {\</sup>rm Poincar\acute{e}\ conjecture}$ 

 $<sup>^3\</sup>mathrm{Restricting}$  the problem to incompressible fluids filling all of  $\mathbb{R}^n$ 

<sup>&</sup>lt;sup>4</sup>Computational Fluid Dynamics

# Part II

# **VDM-Based Navigation Framework**

# Chapter 3

# **Theory and System Architecture**

## Overview

The main contribution of this research, development of VDM-based navigation for aerial applications, is described in this chapter in a general form for aerial applications. The description is preceded by a review of related work in the literature on some forms of using VDM within navigation system while establishing the novelties of current research. General form of the VDM for UAVs and the architecture of the navigation system are detailed, together with the process models of augmented states, as well as observation models considering imperfections in measurements and mounting of sensors.

A considerable amount of wordings and equations throughout this chapter are borrowed form peer reviewed published papers [36, 37, 38, 39, 41] and an under review paper [40].

In this chapter, the proposed VDM-based navigation algorithm is described in detail.

# 3.1 Related Work

There has been research activities on some forms of integration of VDM into navigation system in order to improve the navigation accuracy of aerial vehicles, especially in GNSS outage conditions. Most of these proposed solutions employ INS as main process model within navigation filter [8, 79, 17], while using VDM output either in prediction phase or in update phase within the filter (see Figures 3.1 and 3.2, for example). Such approach is totally IMU dependent and is prone to divergence in case of IMU malfunctions. On the other hand, high frequency content of IMU error coming from sensor noise or platform vibrations is integrated within the navigation system, which increases the drift rate for the navigation solution. Although both INS and VDM have been utilized at the same level in multi process model architectures by some authors [44] (see Figure 3.1), navigation solution at the end has been delivered based on filtered INS output and problems of total reliance on IMU and integration of high frequency error content persist. In many cases, presence of wind is discarded [8, 79, 17, 15], in-flight estimation of VDM parameters in not provided [8, 79, 17, 15, 14], or VDM is integrated into navigation only partially [15, 16, 53]. Some researchers also consider IMUs of higher accuracy [44], which is impractical for small UAVs in terms of size, weight, and cost. A sample collection of mentioned studies are briefly reviewed in the following.

One of the earliest studies on integrating VDM in navigation system of UAVs dates back to 1999, when Koifman and Bar-Itzhack [44] studied the possibility of aiding INS by means of aircraft dynamics for a fixed-wing platform. They suggested a navigation filter with two simultaneous process models, VDM and INS, and employed observations from an unspecified source on horizontal position, horizontal velocity, and attitude of the aircraft within the navigation system. They also used an altimeter to dampen the vertical channel error in GNSS outage conditions. Figure 3.1 depicts a simplified version of navigation system architecture in their research to show the roles of IMU and VDM in determination of PVA solution. The study was aimed at aiding low-grade INS. However, the proposed error statistics suggested a tactical-grade INS, a system for which the quality of sensors is higher than those normally available for small UAVs. Based on simulation results, they concluded that VDM is capable of aiding INS under specific conditions. Appropriate maneuvering of the aircraft was found to be necessary to estimate wind velocity and the errors in dynamic model. They reported that if not estimated, errors of up to 10 percent in dynamic model coefficients led to large accumulation of navigation errors that make the aiding pointless.



Figure 3.1 – Simplified system architecture in [44] showing roles of IMU and VDM in PVA determination

Bryson and Sukkarieh [8] employed VDM to enhance the observability of sensor errors in INS within the navigation system inside a fixed-wing UAV. They compared two configurations, with INS computing the navigation solution in both, where a Kalman filter utilized VDM output to estimate the error in INS. Figure 3.2 depicts a simplified version of navigation system architecture in their research to show the roles of IMU and VDM in determination of PVA solution. In configuration 1, which was found to have a better performance, VDM provided the navigation filter with velocity and attitude data to estimate sensor errors in the IMU, as well as velocity and attitude error in INS solution. Configuration 2 utilized acceleration and rotation rate data from VDM to estimate only the sensor errors in the IMU. Wind was ignored, and there was no mechanism for VDM parameter estimation.

Crocoll et al. [15] proposed and tested a navigation system for a quadcopter UAV that was aided by a translational dynamics model. The authors unified the VDM and INS before Kalman filter prediction step to reduce computational cost and improve stability. Their model ignored rotational dynamics, and no considerable improvement in heading angle estimation was observed. The effect of wind was also not considered, and a parameter identification step was required to identify model parameters, which could not be re-calibrated in-flight. This identification was carried out using measurements from a laser range finder. The in-flight



Figure 3.2 – Simplified system architecture in [8] showing roles of IMU and VDM in PVA determination

parameter estimation [16] and wind estimation [52, 53] were integrated into their system in further developments, which proved the benefits of both in final navigation accuracy. However, only the translational dynamic model was considered in these studies, as well. The reported results revealed improvements over unaided navigation system, but probably not enough to properly guide a UAV in case of GNSS outages of several minutes' duration.

Sendobry reviewed and classified many studies on using VDM in navigation available in the literature, and proposed a new approach of own [68]. This approach, along with the one proposed in [14], may be considered as the closest ones to the one proposed here, in the sense that inertial measurements are treated as observations and not as a process model. In-flight parameter estimation was available within the filter, but no estimation on the wind velocity was considered, and no results on navigation autonomy during GNSS outage was provided. The algorithm performance was investigated through simulations for a quadcopter UAV.

In his doctoral dissertation, Lennon Cork [14] proposed a VDM-based navigation for a full-size aircraft (F16). The dynamic model was borrowed from a publicly available NASA technical report [56], derived from wind-tunnel tests covering a broad range of flight situations, in form of large lookup tables for aerodynamic coefficients. Constructing such a model for a small UAV is overly complex and impractical if not practically unfeasible (outside large companies). In-flight VDM parameter estimation was not proposed, which can become problematic for small UAVs if the structure undergoes some modifications (body or propeller repairs altering aerodynamics, payload change, etc.). Wind estimation was also impossible in absence (or malfunction) of airspeed sensors. Navigation performance of proposed system was investigated via Monte-Carlo simulations, revealing 4 to 8 times improvement in positioning accuracy during GNSS outages of up to 3 minutes.

# 3.2 Methodology Overview

In this research, a navigation algorithm is proposed that integrates VDM within navigation system. The main idea behind this algorithm is to benefit from available information on vehicle dynamic modeling and control input within the navigation system to increase redundancy and accuracy in PVA determination. Since VDM is a formulation of physical laws on UAV motion, integration of it can reject parts of impossible motion suggested by sensors (including IMU) due to their errors, as well as initialization errors. As will be seen in Chapter 5, it significantly mitigates the drift of autonomous navigation during GNSS outages of a few minutes.

Since VDM is only a mathematical model and relies on no extra sensors, integrating it within navigation system adds no additional cost and complexity in terms of hardware, which can be an interesting aspect for small UAVs. VDM needs to be fed with the control input to the UAV, though, which is principally available in the control/autopilot system. Correct time tagging of this data in relation to IMU and other observations is required. Wind velocity is another input that VDM needs, which can be estimated within the proposed navigation system even in absence of airspeed sensor. In case airspeed sensor is available, the internal estimation capability of wind estimation adds redundancy to the system that may be beneficial in case the sensor malfunctions, and better performance is also expected. Detailed evaluation of wind effects on performance of proposed navigation system can be found in Section 5.2.

A key feature in the proposed solution is VDM acting as the main process model within the navigation filter, where its solution is updated with raw IMU observations and if available, observations from GNSS receiver and other sensors (see Figure 3.3). Such architecture avoids the complications of multi-process-model filters [8, 44] and thus leads to simpler filter implementation, smaller state vector, and lower computational cost. It is also preferred over the architectures in which INS is the main process model that gets updated by VDM, due to the following reasons. In case of IMU failure, the proposed architecture can simply stop using all or some of inertial observations<sup>1</sup>, while the architecture with INS as the main process model will fail. Of course, if the controller or the processing unit fails, VDM no longer works either, but in such a case the whole system may be lost irrespective of its navigation quality.

On the other hand, the high frequency measurement noise in IMU data contribute to further drift of the navigation solution when integrated within the navigation filter, as analytically shown in [67]. The mechanical vibrations on the platform also affect the IMU measurements, but not the VDM output. If these vibrations were of considerably lower frequencies than sampling rate of the IMU, they could be properly reconstructed, and thus the navigation accuracy would not be mitigated. However, these vibrations are mainly related to propellers spinning, and typically can have frequencies of hundreds of Hz. Depending on the sampling frequency of IMUs (usually between 50 to 500 Hz), parts of these vibrations may become aliased noise that further deteriorates navigation performance. Theoretically, this can be avoided by choosing IMUs with high enough sampling frequency. However, this may not be affordable in all cases. Therefore, treating the IMU data as observations and avoiding integration of them as a process model is expected to decrease the error growth in autonomous navigation.

The structure of VDM depends on the host platform type (fixed-wing, copter, etc.) and its control actuators, which is generally described well in the literature [13, 19, 22, 7]. The model parameters, however, depend on specific platform at hand. These parameters can be either identified/pre-calibrated, or estimated in-flight. The ability of proposed navigation system for in-flight parameter estimation minimizes the required effort in design and operation.

<sup>&</sup>lt;sup>1</sup>provided that proper fault detection and exclusion is implemented, which goes beyond the scope of current research.



Figure 3.3 – Simplified system architecture in current research showing roles of IMU and VDM in PVA determination

# 3.2.1 Theoretical Benefits of VDM in Navigation

As a mathematical motion model of the platform (UAV), VDM provides information on linear and rotational accelerations, very similar to an IMU. This is an independent additional source of information to what is available in conventional INS-based navigation. Therefore, utilizing VDM in navigation system increases redundancy, and if tuned well enough, the accuracy of navigation.

VDM is constructed based on physical laws of motion, with its parameters being platformdependent. Therefore, even in presence of remaining structural and parametric uncertainties, it suggests motions that are physically possible and exact for a hypothetical platform that is similar to the real UAV. The degree of this similarity depends on the level of modeling errors. INS is platform-independent, but affected by remaining systematic errors, as well as high levels of stochastic errors with complex structure in case of MEMS IMUs. Hence, INS may suggest motions that are not physically possible for any platform similar to the one at hand. Therefore, fusion of VDM data with other available data, removes parts of impossible motions suggested by IMU and reduces navigation error, as a result.

When GNSS updates are available regularly, the growth of navigation errors due to accumulation of IMU errors is controlled. However, in long GNSS outages, bringing another source of information and removing parts of IMU error can considerably limit growth of navigation error. This is what VDM-based navigation does by applying physical constraints on dynamic behavior of the platform, even though with some uncertainty.

The role of VDM in UAV navigation may be compared to the role of non-holonomic constraints in terrestrial vehicle navigation. Although these constraints are very simplistic, when combined with IMU data, they substantially decrease uncertainty of inertial coasting by applying physical constraints on motion, even though partially. In [18], as one example of many references on this matter, applying non-holonomic constraints and odometry via a wheel encoder has reported to reduce position error in horizontal channel from 808 meters to only 11 meter in a GNSS free period of 2 minutes, meaning an improvement of over 70 times.

# 3.3 Navigation System Architecture

The sensors used for navigation of a UAV fall under two categories, "autonomous sensors" and "environment dependent sensors". IMUs and pressure sensors are examples of autonomous

sensors that provide data regardless of environmental features and conditions such as texture, visibility, or signal reception conditions. On the other hand, there are environment dependent sensors such as RF<sup>1</sup> ranging sensors (GNSS receivers for example) and vision based sensors. In conventional navigation system for a UAV, data from all these sensors are fused in a navigation system with INS being the process model to provide the navigation solution, as depicted schematically in Figure 3.4. Time stamping of sensor data to a common reference (usually GPS time) may be handled inside sensors or externally.



- S I: Autonomous sensors (IMU, pressure sensor, etc.)
- S II: Environment dependent sensors (RF-ranging (e.g., GNSS), vision, ultrasound, optical flow, etc.)
- **INS:** Inertial Navigation System

Figure 3.4 – Conventional (INS-based) navigation system for UAVs

As explained in Section 3.2, the main idea behind the proposed navigation algorithm in this research is integrating information on dynamic model of UAV in a particular way within the navigation system. This is achieved by using VDM as process model in navigation filter (Extended Kalman filter, for example) and feeding the navigation filter with control input to UAV, as schematically depicted in Figure 3.5. This figure, shows the general implementation scheme for the proposed navigation system.

Depending on specific sensors available on the platform, there are some hardware requirements for the proposed navigation method to work. The autopilot/control unit output to actuators needs to be accessible by navigation system (since this information is needed by VDM) and time-stamped with respect to other sensory data to a common reference. An alternative would be having sensors directly measuring actuator states such as propeller speed. For autonomous navigation, for example, usually an IMU is a minimum necessary sensor setup. With the proposed method, there is nevertheless the possibility for the system to work (determine PVA) without the IMU, but with at least a single GNSS receiver or another setup of sensors providing position fixes. Since an IMU is almost always used within a navigation system for unmanned vehicles, such configuration can be most beneficial as a backup solution in case of IMU failure to increase reliability of the whole system.

Control input and data from all sensors must be properly referenced to a common time base in order to be used within the navigation system. This requires a hardware/software time stamper to perform data tagging with respect to a common time reference. Depending on configuration of sensors, a partial time stamping may be performed in other manners. For

<sup>&</sup>lt;sup>1</sup>Radio Frequency



S II: Environment dependent sensors (RF-ranging (e.g., GNSS), vision, ultrasound, optical flow, etc.)VDM: Vehicle Dynamic Model

Figure 3.5 – General scheme for the proposed (VDM-based) navigation system for UAVs

example, in case both an IMU and a GNSS receiver are available, there may be a connection between them in which the GNSS time frame is used to tag IMU data. A time stamping unit may also be available within the controller. In any case, all the data flow to the navigation system needs to be referenced to a common time base.

# 3.3.1 General Form of VDM

Vehicle dynamic model (VDM) is a mathematical model describing dynamics of motion of the platform. The VDM here is derived based on Newtonian formulation of classical mechanics, and is presented in state space form with the following state vector<sup>1</sup>.

$$\mathbf{X}_{n} = \left[\mathbf{r}_{e}^{l}{}^{T}, \mathbf{v}_{e}^{l}{}^{T}, \mathbf{q}_{b}^{l}{}^{T}, \boldsymbol{\omega}_{ib}^{b}{}^{T}\right]^{T}$$
(3.3.1)

The position vector  $\mathbf{r}_e^l = [\phi, \lambda, h]^T$  represents position of platform (represented by position of body frame origin) in earth frame and in ellipsoidal coordinates. The velocity vector  $\mathbf{v}_e^l = [v_N, v_E, v_D]^T$  denotes velocity of platform as observed in earth frame and expressed in

 $<sup>^{1}</sup>$ See footnote 1 on page 17 for a discussion on dependence of quaternion components in the state vector.

#### Theory and System Architecture

local (NED) frame. The quaternion  $\mathbf{q}_b^l = [q_0, q_1, q_2, q_3]^T$  defines orientation of body frame (platform) with respect to local frame. Finally, the angular velocity vector  $\boldsymbol{\omega}_{ib}^b = [\omega_x, \omega_y, \omega_z]^T$  represents angular velocity (rotation rate) of body frame with respect to inertial frame, expressed in body frame.

The dynamic model is derived based on Newton-Euler formulation of classical mechanics as presented in Section 2.1.1 and can be expressed as [38]

$$\dot{\mathbf{X}}_{n} = \begin{bmatrix} \dot{\mathbf{r}}_{e}^{l} \\ \dot{\mathbf{v}}_{e}^{l} \\ \dot{\mathbf{q}}_{b}^{l} \\ \dot{\boldsymbol{\omega}}_{ib}^{b} \end{bmatrix} = \begin{bmatrix} \mathbf{D}^{-1}\mathbf{v}_{e}^{l} \\ \mathbf{C}_{b}^{l}\mathbf{f}^{b} - \left(2\mathbf{\Omega}_{ie}^{l} + \mathbf{\Omega}_{el}^{l}\right)\mathbf{v}_{e}^{l} + \mathbf{g}^{l} \\ \frac{1}{2}\mathbf{q}_{b}^{l}\otimes\left[\omega_{ib}^{b} - \left(\mathbf{C}_{b}^{l}\right)^{T}\left(\omega_{ie}^{l} + \omega_{el}^{l}\right)\right]_{q} \\ \left(\mathbf{I}^{b}\right)^{-1}\left[\mathbf{M}^{b} - \mathbf{\Omega}_{ib}^{b}\left(\mathbf{I}^{b}\omega_{ib}^{b}\right)\right] \end{bmatrix}, \qquad (3.3.2)$$

where  $\mathbf{I}^{b}$  denotes the matrix of moments of inertia of the UAV, and  $\mathbf{M}^{b}$  represents the [resultant] vector of moments applied on the UAV, both expressed in b-frame. The rest of the symbols were introduced/defined in Section 1.1.4.

The presented model (Equation (3.3.2)) is so far a general form of dynamic model for any rigid body in arbitrary motion in 3D space affected by Earth gravity, external specific force  $\mathbf{f}$ , and external moment  $\mathbf{M}$ , based on classical mechanics. Completing the model for a specific type of vehicle, a fixed-wing UAV for example, requires proper models for  $\mathbf{f}$  and  $\mathbf{M}$ , an aerodynamic model in this case, which can be a function of dynamic states, control inputs, and physical parameters of the platform. Those physical parameters can be different per specific UAV.

# 3.3.2 Importance of Modeling Earth Effects

Although a "non-rotating flat Earth" model is widely assumed for UAV applications in robotics community, a "rotating ellipsoidal Earth" model [59] is used in this research. It is a known fact that for large platforms, even with rather short flight ranges and moderate speeds, considering a non-rotating flat Earth would result in large navigation errors. However, some simplified calculations reveal that employing a non-rotating flat Earth model may influence error accumulation to a non-trivial extent, even for small UAVs.

Ignoring Earth rotation introduces errors mainly due to discarding Coriolis acceleration. For a UAV flying at 20 m/s, the discarded acceleration can be as big as  $2.92 \times 10^{-3} m/s^2$  (=  $2 \times 7.29 \times 10^{-5} \times 20$ ). In 3 minutes of inertial coasting, this leads to a position error of 47 m (=  $1/2 \times 2.92 \times 10^{-3} \times 180^2$ ).

Assuming a flat Earth also introduces errors due to discarding changes in the direction of gravitational force. Over a range of 1 km, considering the mean radius of 6371 km for Earth, this change in direction is  $1/6371 \ rad$ . The average committed error in acceleration will be  $7.70 \times 10^{-4} \ m/s^2$  (=  $1/2 \times 1/6371 \times 9.81$ ). In 3 minutes of inertial coasting, this can lead to a position error of 12 m (=  $1/2 \times 7.70 \times 10^{-4} \times 180^2$ ), setting the combined error to 49 m (=  $\sqrt{47^2 + 12^2}$ ).

Error of such magnitude seems to be negligible compared to the errors that typical MEMS<sup>1</sup> IMUs produce in the same situation (in order of kilometers). However, as will be seen in

<sup>&</sup>lt;sup>1</sup>Micro-Electro-Mechanical Systems

Chapter 5, this is almost on par with the ranges of errors in VDM-based navigation, and thus quite important to avoid. Also, as those errors grow linearly with speed and flights range and quadratically with time, modeling Earth effects becomes more important for larger platforms and in beyond visual line of sight (BVLOS) operations.

# 3.3.3 VDM as Process Model

The proposed navigation system utilizes VDM as the main process model within the navigation filter. The filter provides estimation of navigation states and other necessary augmented states as will be discussed in Section 3.4. An extended Kalman filter (EKF) [23] is chosen to estimate the states (**X**) and associated covariance matrix (**P**). Other types of filters/estimators such as unscented Kalman filter (UKF) could also be used. As mentioned in Section 1.2.2, while prediction of state vector **X** is provided by the generally nonlinear process model, EKF utilizes a linearized version of the process model (via  $\mathbf{F} = \partial \dot{\mathbf{X}} / \partial \mathbf{X}$ , with  $\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}, \mathbf{U}, t)$  and **X** being the state vector, **U** being the vector of deterministic inputs, and t denoting time) and a linearized version of observation model (via  $\mathbf{H} = \partial \mathbf{Z} / \partial \mathbf{X}$ , with  $\mathbf{Z} = \mathbf{h}(\mathbf{X}, t)$  being the observation vector) to provide prediction of covariance matrix **P** and to update both state vector and covariance matrix. More details on Kalman filters can be found in [23] and many other sources available on the subject. Description of different states and their process model will be presented in Section 3.4.

As depicted in Figure 3.6, VDM provides the navigation solution  $(\mathbf{X}_n)$  that is updated as a part of the augmented states vector  $(\mathbf{X})$  by the filter based on available observations. Hence, IMU data are treated as observations, just the way data from other sensors such as GNSS receiver, barometer, and airspeed sensor are when available. It is important to note that IMU observations are related to system states via the VDM, as discussed in Section 3.5.1. Any other available sensor such as optic flow sensor or magnetometer can also be integrated within the navigation system as an additional observation source.

VDM is fed with the control input  $(\mathbf{U})$  acting on the UAV, which is commanded by the autopilot and therefore available. To be more exact, VDM needs to be fed with actuator states. If internal dynamics of the actuators is ignored, actuator states will be equivalent to the control input  $(\mathbf{U})$ . Otherwise, the control input is fed to associated models for actuator dynamics, which provide actuator states to be fed to the VDM. In such a case, the unknown parameters of actuator dynamics model may also be considered as a part of the augmented states to be estimated within the filter.

Another needed input for VDM is the wind velocity  $(\mathbf{X}_w)$ , which can be estimated within the navigation system with or without the aid of airspeed sensors. Of course, if airspeed sensors are available, better performance is expected thanks to the direct measurements and the introduced redundancy.

Finally, the parameters of VDM, which are basically the physical properties of the UAV, are required. Pre-calibration of these parameters as fixed values is an option. However, to increase flexibility and accuracy of the proposed approach while minimizing design effort, in-flight estimation/refinement of VDM parameters  $(\mathbf{X}_{vp})$  is implemented.

The state vector can be further augmented to include actuator parameters, sensor errors, and sensor mounting misalignment. Depending on the needs and available observations and



Figure 3.6 – VDM-based navigation filter architecture ( $\tilde{\mathbf{X}}_k \equiv \hat{\mathbf{X}}_{k|k-1}$ )

maneuvers to provide the required observability, these can be estimated or only compensated for. State space augmentation is detailed in Section 3.4.

# **3.4** State Space Augmentation

The navigation filter may be used to estimate some parameters or variables in addition to the navigation states, and this is done via augmenting the state vector. In the following, all the states that may be included in the augmented state vector (either for the purpose of calibration or navigation) are introduced with their process model. The process model is dynamic ( $\dot{\mathbf{X}} \neq 0$ ) in general, but it can be static ( $\dot{\mathbf{X}} = 0$ ) if a constant variable (a parameter) is being estimated within the filter.

# **3.4.1** Actuator Dynamics

The internal dynamic of any actuator state  $a_i$  (such as propeller speed, and deflections of aileron, elevator, and rudder for a fixed-wing UAV) is modeled considering first-order delay, scaling, and shifting of the command as

$$\dot{a}_i = \frac{1}{\tau_i} \left( b_i u_i + c_i - a_i \right), \tag{3.4.1}$$

where  $a_i$  denotes the actuator state and  $u_i$  is the desired (commanded) value for  $a_i$ . Three parameters are recognized for this model, being  $\tau_i$  as a first order delay,  $b_i$  as a scale factor, and  $c_i$  as a shift. All these parameters can be also included in the augmented state vector. If this is the case, they are modeled as constants (for example,  $\dot{\tau}_i = 0$ ). To sum up, if there are M actuators on the platform, the state space is augmented by an actuator state vector as

$$\mathbf{X}_a = \left[a_1, a_2, \dots, a_M\right]^T,\tag{3.4.2}$$

and an actuator parameter state vector as

$$\mathbf{X}_{ap} = [\tau_1, b_1, c_1, \tau_2, b_2, c_2, \dots, \tau_M, b_M, c_M]^T.$$
(3.4.3)

The process model for  $\mathbf{X}_a$  is simply a collocation of (3.4.1), and the process model for  $\mathbf{X}_{ap}$  is as following.

$$\dot{\mathbf{X}}_{ap} = [0]_{3M \times 1} \tag{3.4.4}$$

# 3.4.2 VDM Parameters

Parameters used in dynamic model of the UAV reflect physical properties of the platform, which are normally constant. However, their constant values may not necessarily be known initially (e.g., for a new UAV) with sufficient certitude, hence the possibility of estimating them within the filter is desirable, at least for the purpose of calibration. A static model is used for them within the filter. The VDM parameters vector  $(\mathbf{X}_{vp})$  contains mainly the aerodynamic coefficients used in the UAV dynamic modeling. Mass and moments of inertia of the UAV are not included in this vector, since they appear as scaling factors in the equations of motion and therefore they are completely correlated with the already included coefficients of aerodynamic forces and moments. Also, geometric measures of the UAV such as propeller diameter can be measured physically (or using 3D models of the platform) with sufficient accuracy. In such a case, they are not part of the augmented states vector. If there are Naerodynamic coefficients  $(p_i)$  in the VDM estimation of which within the filter is intended, the VDM parameters vector will be

$$\mathbf{X}_{vp} = [p_1, p_2, \dots, p_N]^T, \qquad (3.4.5)$$

with the process model as

$$\dot{\mathbf{X}}_{vp} = [0]_{N \times 1} \tag{3.4.6}$$

# 3.4.3 Wind Velocity

The wind velocity is stated as a vector in local-level frame consisting of the three components in north, east, and down directions.

$$\mathbf{X}_w = [w_n, w_e, w_d]^T \tag{3.4.7}$$

Although in usual cases wind velocity is expected to change rather slowly and smoothly from one moment to another, in general there is no information on such transition. Therefore, no deterministic part is considered in the process model for wind velocity, and only a stochastic model is considered for wind. If a white noise is assumed to rule the transition in time, this leads to a random walk process model. Constant part of the wind is also covered by the random walk process if a non-zero initial uncertainty is attributed to it within the filter.

# 3.4.4 Sensor Errors

Time correlated errors of a sensors may be modeled and estimated within the filter stochastically. Frequently used models are bias (random constant), random walk, and autoregressive models such as first order Gauss-Markov. The time uncorrelated error for each sensor is considered as a Gaussian white noise that is taken into account as observation uncertainty within the filter. The usage of sensor error sates (as well as sensor mounting misalignment states introduced in Section 3.4.5) in observation models is detailed in Section 3.5, while the purpose of this section is presenting these states generally with their process models.

In case a random walk process is considered for the error of a sensor, there will be no added value for modeling separately a bias, as the bias can be estimated by the random walk provided that a non-zero initial uncertainty is set for it. In other words, a random walk process with non-zero initial uncertainty is equivalent to the sum of a bias and a random walk process with zero initial uncertainty.

# IMU

For the IMU in this research, a random walk  $(\mathbf{e}_{RW}^{I})$  and a first order Gauss-Markov  $(\mathbf{e}_{GM}^{I})$  error vector are considered as additive errors, each consisting three elements for accelerometers (denoted by superscripts  $^{ai}$ ) and three elements for gyroscopes (denoted by superscripts  $^{gi}$ ). This model has been found sufficient for the low-cost IMU in consideration here, but can be extended as needed. Non-orthogonality and scale factors were pre-determined and compensated during filtering<sup>1</sup>.

$$\mathbf{e}_{RW}^{I} = \left[e_{RW}^{a1}, e_{RW}^{a2}, e_{RW}^{a3}, e_{RW}^{g1}, e_{RW}^{g2}, e_{RW}^{g3}\right]^{T}$$
(3.4.8)

<sup>&</sup>lt;sup>1</sup>In practice, not all the different error types are well observable, and therefore, some error types (e.g., scale factor) may be partially absorbed by other error types (e.g., additive errors). The unabsorbed parts will remain as sources of error, though. One may only do their best to estimate them and then accept the residuals and possibly modify error statistics within the filter to better reflect the real situation.

$$\mathbf{e}_{GM}^{I} = \left[e_{GM}^{a1}, e_{GM}^{a2}, e_{GM}^{a3}, e_{GM}^{g1}, e_{GM}^{g2}, e_{GM}^{g3}\right]^{T}$$
(3.4.9)

The IMU error states vector concatenates these two as

$$\mathbf{X}_{e}^{I} = \left[ \left( \mathbf{e}_{RW}^{I} \right)^{T}, \left( \mathbf{e}_{GM}^{I} \right)^{T} \right]^{T}.$$
(3.4.10)

# Barometer

For the barometer, an additive error modeled as a random walk  $(e_{RW}^B)$  is considered, as well as a scale factor  $(e_{SF}^B)$  that is also modeled by a random walk. Random walk is chosen over bias to account for unpredicted changes in environmental conditions (temperature and pressure). The barometer error states vector contains the additive random walk and the scale factor as

$$\mathbf{X}_{e}^{B} = \begin{bmatrix} e_{RW}^{B}, e_{SF}^{B} \end{bmatrix}^{T}.$$
(3.4.11)

#### Airspeed Sensor

The considered error states for the airspeed sensor are very similar to the barometer, an additive error as a random walk  $(e_{RW}^A)$ , as well as a scale factor  $(e_{SF}^A)$  that is also modeled by a random walk. Hence, the airspeed error states vector is defined as

$$\mathbf{X}_{e}^{A} = \left[e_{RW}^{A}, e_{SF}^{A}\right]^{T}.$$
(3.4.12)

# 3.4.5 Sensor Mounting Misalignments

A sensor frame is associated to any sensor mounted on the platform, with the origin at the effective point of the sensor, at which the measurements are made, and the axes oriented in the directions in which the measurements are made, for the ones with a direction dependent measurement. In general, the sensor frames are not aligned perfectly with the body frame. The misalignment between a sensor frame and the body frame is defined via translational (i.e., lever-arm) and rotational (i.e., boresight) components. Lever-arm is a position vector from body frame origin to sensor frame origin, expressed in body frame. Boresight is a rotation (in any representation form) expressing the attitude of sensor frame with respect to body frame. While lever-arm needs to be determined for every sensor, boresight is defined only for some sensors. For example, measurements from barometer and GNSS receivers do not have any orientation in body frame. The additional states accounting for the sensor mounting misalignments are detailed in the following. The usage of these states in observation models is detailed in Section 3.5, though.

If the platform and the mountings of rigidly attached sensors are not physically modified, the lever-arms and boresights are stable in time. Since their values may be difficult to measure directly, especially for boresights, they are modeled as random constants to be refined in-flight during calibration flights. Once they are estimated with satisfactory certainty, they can be removed from augmented states and be used only for compensation<sup>1</sup>.

# $\mathbf{IMU}$

The relative alignment of IMU and body frame is depicted in Figure 3.7. The lever-arm is expressed as the position vector  $\mathbf{r}_{bI}^b = [L_1^I, L_2^I, L_3^I]^T$ , and boresight is represented by the vector  $\boldsymbol{\Theta}_{bI}^b = [B_1^I, B_2^I, B_3^I]^T$  containing the three Euler angles parameterizing the rotation matrix  $\mathbf{C}_b^I$ . This rotation matrix is calculated as

$$\mathbf{C}_{b}^{I} = \mathbf{C}_{1}(B_{1})\mathbf{C}_{2}(B_{2})\mathbf{C}_{3}(B_{3}), \qquad (3.4.13)$$

with elementary rotation functions  $\mathbf{C}_i(\cdot)$  defined in Equation (1.1.2). The associated state vector is then as following.

$$\mathbf{X}_{m}^{I} = \left[ \left( \mathbf{r}_{bI}^{b} \right)^{T}, \left( \boldsymbol{\Theta}_{bI}^{b} \right)^{T} \right]^{T}$$
(3.4.14)



Figure 3.7 – IMU frame alignment with respect to body frame

As discussed in sensor frame definition in Section 1.1.3, the internal non-orthogonality of the sensor is modeled separately, and the introduced misalignment here refers to the misalignment between the IMU sensor frame after accounting for non-orthogonality and the body frame.

## **GNSS** Antenna

To express the misalignment between the GNSS antenna phase center and the body frame, only the lever-arm  $(\mathbf{r}_{bG}^b)$  is required, making the associated state vector look like the following.

$$\mathbf{X}_m^G = \mathbf{r}_{bG}^b = \begin{bmatrix} L_1^G, L_2^G, L_3^G \end{bmatrix}^T$$
(3.4.15)

<sup>&</sup>lt;sup>1</sup>In practice, sensor mounting misalignments, especially in case of boresight, can be very weekly observable. Therefore, their fine estimation is frequently discarded and some modification (in conservative direction) is made in sensor error statistics within the filter to partially absorb their effects.

# Barometer

Similar to GNSS antenna, only a lever-arm  $(\mathbf{r}_{bB}^b)$  is required to express the misalignment between the barometer frame and the body frame, resulting in the following state vector.

$$\mathbf{X}_{m}^{B} = \mathbf{r}_{bB}^{b} = \begin{bmatrix} L_{1}^{B}, L_{2}^{B}, L_{3}^{B} \end{bmatrix}^{T}$$
(3.4.16)

## **Airspeed Sensor**

For airspeed sensor, relative alignment with respect to body frame is depicted in Figure 3.8. In this case, in addition to the position vector  $\mathbf{r}_{bA}^b = [L_1^A, L_2^A, L_3^A]^T$  for lever-arm, the vector  $\mathbf{\Theta}_{bA}^b = [B_1^A, B_2^A]^T$  containing the two boresight angles is also required to express the direction of the unit vector  $\hat{\mathbf{u}}_A^b$  along which the airspeed measurement is made. As the rotation of the airspeed sensor around its effective axis  $(\hat{\mathbf{u}}_A^b)$  does not affect its measurements, a third angle is not required for expressing the boresight. The unit vector  $\hat{\mathbf{u}}_A^b$  is calculated as

$$\hat{\mathbf{u}}_{A}^{b} = \begin{bmatrix} \cos B_{2}^{A} \cos B_{1}^{A} \\ \cos B_{2}^{A} \sin B_{1}^{A} \\ \sin B_{2}^{A} \end{bmatrix}.$$
(3.4.17)

The state vector for airspeed sensor mounting misalignment is as following.

$$\mathbf{X}_{m}^{A} = \left[ \left( \mathbf{r}_{bA}^{b} \right)^{T}, \left( \boldsymbol{\Theta}_{bA}^{b} \right)^{T} \right]^{T}$$
(3.4.18)



Figure 3.8 – Airspeed frame alignment with respect to body frame

# 3.4.6 State Vector in Most General Case

In the most general case here, the augmented state vector is composed of navigation states  $(\mathbf{X}_{a})$ , actuator states  $(\mathbf{X}_{a})$ , actuator parameters  $(\mathbf{X}_{ap})$ , VDM parameters  $(\mathbf{X}_{vp})$ , wind velocity  $(\mathbf{X}_{w})$ , IMU error  $(\mathbf{X}_{e}^{I})$ , barometer error  $(\mathbf{X}_{e}^{B})$ , Airspeed error  $(\mathbf{X}_{e}^{A})$ , IMU mounting misalignment

 $(\mathbf{X}_m^I)$ , GNSS antenna mounting misalignment  $(\mathbf{X}_m^G)$ , barometer mounting misalignment  $(\mathbf{X}_m^B)$ , and airspeed sensor mounting misalignment  $(\mathbf{X}_m^A)$ .

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_n^T, \mathbf{X}_a^T, \mathbf{X}_{ap}^T, \mathbf{X}_{vp}^T, \mathbf{X}_w^T, \mathbf{X}_e^{T}, \mathbf{X}_e^{B^T}, \mathbf{X}_e^{A^T}, \mathbf{X}_m^{A^T}, \mathbf{X}_m^{G^T}, \mathbf{X}_m^{B^T}, \mathbf{X}_m^{A^T} \end{bmatrix}^T$$
(3.4.19)

However, not all these states are necessarily present in the augmented states vector in every situation. For example, sensor mounting misalignment or the VDM and actuator parameters may have been already calibrated to an acceptable level for excluding them from the estimation. Also, keeping all of these states at the same time will most probably not result in best performance outside a calibration scenario, due to the lack of enough observability. For example, if the altitude change is small, the additive error and the scale factor for the barometer may not decorrelate and therefore they are estimated poorly. Apart from the choice on augmented states, observability depends on the available observations and the maneuvers that the UAV performs. The designer needs to consider all these facts when deciding on the states to include.

# 3.5 Observation Models

Observations from onboard sensors need to be clearly related to system states via observation models in order to be used in the filter. This section provides the observation models for all the employed sensors, considering the modeled errors and the mounting misalignments.

# 3.5.1 IMU data

An ideal (error-free) IMU consists of an orthogonal triad of accelerometers, measuring specific force applied to the IMU casing expressed in IMU frame ( $\mathbf{f}_{I}^{I}$ ), and an orthogonal triad of gyroscopes measuring rotation rate (angular velocity) of the casing with respect to inertial frame expressed in IMU frame ( $\boldsymbol{\omega}_{iI}^{I}$ ). In VDM navigation equations (Equation (3.3.2)), specific force applied to body expressed in body frame ( $\mathbf{f}_{b}^{b}$ ) and rotation rate of body frame with respect to inertial frame expressed in body frame ( $\boldsymbol{\omega}_{ib}^{b}$ ) are required. The IMU is assumed to be rigidly attached to the body, with misalignment ( $\mathbf{X}_{m}^{I}$ ) as expressed in Equation (3.4.14) and additive error ( $\mathbf{X}_{e}^{I}$ ) as expressed in Equation (3.4.10).

To obtain the relation between  $\mathbf{f}_{I}^{I}$  and  $\mathbf{f}_{b}^{b}$ , let's start with the relation between position vector of IMU and body frame origins, expressed in inertial frame.

$$\mathbf{r}_{I}^{i} = \mathbf{r}_{b}^{i} + \mathbf{C}_{b}^{i} \mathbf{r}_{b} I^{b} \tag{3.5.1}$$

Differentiating this equation twice with respect to time and applying the rigid attachment assumption (meaning constant  $\mathbf{r}_b I^b$ ) and using Equation (1.1.7) leads to

$$\ddot{\mathbf{r}}_{I}^{i} = \ddot{\mathbf{r}}_{b}^{i} + \mathbf{C}_{b}^{i} \dot{\mathbf{\Omega}}_{ib}^{b} \mathbf{r}_{bI}^{b} + \mathbf{C}_{b}^{i} \mathbf{\Omega}_{ib}^{b} \mathbf{\Omega}_{ib}^{b} \mathbf{r}_{bI}^{b}.$$
(3.5.2)

On the other hand, according to Equation (1.1.21),  $\ddot{\mathbf{r}}_{I}^{i}$  and  $\ddot{\mathbf{r}}_{b}^{i}$  can be written as following.

$$\ddot{\mathbf{r}}_{I}^{i} = \mathbf{f}_{I}^{i} + \mathbf{g}_{I}^{i} = \mathbf{C}_{b}^{i}\mathbf{C}_{I}^{b}\mathbf{f}_{I}^{I} + \mathbf{g}_{I}^{i}$$

$$(3.5.3)$$
$$\ddot{\mathbf{r}}_b^i = \mathbf{f}_b^i + \mathbf{g}_b^i = \mathbf{C}_b^i \mathbf{f}_b^b + \mathbf{g}_b^i \tag{3.5.4}$$

Substitution of Equations (3.5.3) and (3.5.4) in Equation (3.5.2) and multiplying both sides by  $\mathbf{C}_{i}^{b}$  yields

$$\mathbf{C}_{I}^{b}\mathbf{f}_{I}^{I} + \mathbf{C}_{i}^{b}\mathbf{g}_{I}^{i} = \mathbf{f}_{b}^{b} + \mathbf{C}_{i}^{b}\mathbf{g}_{b}^{i} + \dot{\mathbf{\Omega}}_{ib}^{b}\mathbf{r}_{bI}^{b} + \mathbf{\Omega}_{ib}^{b}\mathbf{\Omega}_{ib}^{b}\mathbf{r}_{bI}^{b}.$$
(3.5.5)

Discarding the difference between  $\mathbf{g}_{I}^{i}$  and  $\mathbf{g}_{b}^{i}$ , and denoting  $\mathbf{f}_{I}^{I}$  as  $\mathbf{f}^{I}$  and  $\mathbf{f}_{b}^{b}$  as  $\mathbf{f}^{b}$  to simplify notation, Equation (3.5.5) can be solved for  $\mathbf{f}^{I}$  as following.

$$\mathbf{f}^{I} = \mathbf{C}_{b}^{I} \left( \mathbf{f}^{b} + \left( \dot{\mathbf{\Omega}}_{ib}^{b} + \mathbf{\Omega}_{ib}^{b} \mathbf{\Omega}_{ib}^{b} \right) \mathbf{r}_{bI}^{b} \right)$$
(3.5.6)

Considering  $\boldsymbol{\omega}_{iI}^{I} = \boldsymbol{\omega}_{ib}^{I} + \boldsymbol{\omega}_{bI}^{I}$  and applying the rigid attachment assumption (meaning constant  $\mathbf{C}_{b}^{I}$  and hence, zero  $\boldsymbol{\omega}_{bI}^{I}$ ), one can have

$$\boldsymbol{\omega}_{iI}^{I} = \mathbf{C}_{b}^{I} \boldsymbol{\omega}_{ib}^{b}. \tag{3.5.7}$$

Finally, putting  $\mathbf{f}^{I}$  from Equation (3.5.6) and  $\boldsymbol{\omega}_{iI}^{I}$  from Equation (3.5.7) together in  $\mathbf{Z}_{I}$  vector, corrupted by non-orthogonality, scale factors, additive errors as in Equation (3.4.10), and  $\mathbf{n}_{I}$  measurement noise, IMU observation model is obtained as

$$\mathbf{Z}_{I} = (\mathbf{I} + \mathbf{N} + \mathbf{S}) \begin{bmatrix} \mathbf{C}_{b}^{I} \left( \mathbf{f}^{b} + \left( \dot{\mathbf{\Omega}}_{ib}^{b} + \mathbf{\Omega}_{ib}^{b} \mathbf{\Omega}_{ib}^{b} \right) \mathbf{r}_{bI}^{b} \\ \mathbf{C}_{b}^{I} \boldsymbol{\omega}_{ib}^{b} \end{bmatrix} + \mathbf{X}_{e}^{I} + \mathbf{n}_{I}, \qquad (3.5.8)$$

where  $\mathbf{I}$  is the identity matrix. Both the skew-symmetric matrix of non-orthogonality factors  $(\mathbf{N})$ , and the diagonal matrix of scale factor  $(\mathbf{S})$  are pre-determined and compensated for within the navigation filter.

#### 3.5.2 GNSS Position and Velocity

In loosely coupled integration, GNSS observations consist of position of GNSS antenna in Earth frame ( $\mathbf{r}_G^e$ ) and velocity of GNSS antenna with respect to Earth frame expressed in local frame ( $\mathbf{C}_e^l \mathbf{v}_G^e$ ). Considering GNSS antenna mounting misalignment as expressed in Equation (3.4.15),  $\mathbf{r}_G^e$  can be related to system states as

$$\mathbf{r}_{G}^{e} = \mathbf{r}_{b}^{e} + \mathbf{r}_{bG}^{e} = \mathbf{r}_{b}^{e} + \mathbf{C}_{b}^{e} \mathbf{r}_{bG}^{b} = \mathbf{r}_{b}^{e} + \mathbf{C}_{l}^{e} \mathbf{C}_{b}^{l} \mathbf{r}_{bG}^{b}.$$
(3.5.9)

Differentiating Equation (3.5.9) with respect to time and applying the rigid attachment assumption (meaning constant  $\mathbf{r}_{bG}^b$ ) yields

$$\dot{\mathbf{r}}_{G}^{e} = \dot{\mathbf{r}}_{b}^{e} + \mathbf{C}_{b}^{e} \mathbf{\Omega}_{eb}^{b} \mathbf{r}_{bG}^{b}.$$
(3.5.10)

Defining  $\mathbf{v}_{ea}^{l} \triangleq \mathbf{C}_{e}^{l} \mathbf{v}_{a}^{e} = \mathbf{C}_{e}^{l} \dot{\mathbf{r}}_{a}^{e}$  for any arbitrary point *a*, Equation (3.5.10) can be rewritten as

$$\mathbf{v}_{eG}^{l} = \mathbf{v}_{eb}^{l} + \mathbf{C}_{e}^{l}\mathbf{C}_{b}^{e}\boldsymbol{\Omega}_{eb}^{b}\mathbf{r}_{bG}^{b} = \mathbf{v}_{eb}^{l} + \mathbf{C}_{b}^{l}\boldsymbol{\Omega}_{eb}^{b}\mathbf{r}_{bG}^{b}.$$
(3.5.11)

Considering  $\Omega_{eb}^b = \Omega_{ib}^b - \Omega_{ie}^b$  and  $\Omega_{eb}^b = \mathbf{C}_l^b \Omega_{eb}^l \mathbf{C}_b^l$ , and denoting  $\mathbf{v}_{eb}^l$  as  $\mathbf{v}_e^l$  to simplify notation, Equation (3.5.11) takes the following form.

$$\mathbf{v}_{eG}^{l} = \mathbf{v}_{e}^{l} + \left(\mathbf{C}_{b}^{l}\mathbf{\Omega}_{ib}^{b} - \mathbf{\Omega}_{ie}^{l}\mathbf{C}_{b}^{l}\right)\mathbf{r}_{bG}^{b}$$
(3.5.12)

Finally, putting  $\mathbf{r}_{G}^{e}$  from Equation (3.5.9) and  $\mathbf{v}_{eG}^{l}$  from Equation (3.5.12) together in  $\mathbf{Z}_{G}$  vector, corrupted by  $\mathbf{n}_{G}$  measurement noise, GNSS observation model is obtained as following.

$$\mathbf{Z}_{G} = \begin{bmatrix} \mathbf{r}_{b}^{e} + \mathbf{C}_{l}^{e} \mathbf{C}_{b}^{l} \mathbf{r}_{bG}^{b} \\ \mathbf{v}_{e}^{l} + \left( \mathbf{C}_{b}^{l} \mathbf{\Omega}_{ib}^{b} - \mathbf{\Omega}_{ie}^{l} \mathbf{C}_{b}^{l} \right) \mathbf{r}_{bG}^{b} \end{bmatrix} + \mathbf{n}_{G}$$
(3.5.13)

#### 3.5.3 Barometric Altitude

A barometer measures static air pressure. Assuming a hydrostatic model for the atmosphere, this pressure is then related to the altitude of the barometer inlet point  $h_B$ , knowing the atmospheric states (temperature and pressure) at some initial point for each mission. Considering barometer misalignment as expressed in Equation (3.4.16),  $h_B$  can be related to system states as

$$h_B = h_b + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{C}_b^l r_{bB}^b.$$
(3.5.14)

Corrupting  $h_B$  with barometer error as expressed in Equation (3.4.11) and  $n_B$  measurement noise, and denoting  $h_b$  as h to simplify notation, barometer observation model is obtained as following.

$$Z_B = \left(1 + e_{SF}^B\right) \left(h + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{C}_b^l r_{bB}^b \right) + e_{RW}^B + n_B$$
(3.5.15)

Another option to employ the barometric data is including the parameters of the hydrostatic model for the atmosphere in the navigation (or a separate) filter and keep estimating them as long as GNSS position observations are present. This approach was explored in [47].

#### 3.5.4 Airspeed

A single-axis airspeed sensor measures dynamic pressure (difference between total/stagnation pressure and static pressure) along its axis, which after some internal calibration and transformation gives the airspeed along its axis at inlet point  $V_{\hat{\mathbf{u}}_A}$ , which can be expressed as following.

$$V_{\hat{\mathbf{u}}_{A}} = \mathbf{V}_{eA}^{b} \cdot \hat{\mathbf{u}}_{A}^{b} = \left(\mathbf{V}_{eA}^{b}\right)^{T} \hat{\mathbf{u}}_{A}^{b} = \left(\mathbf{V}_{eA}^{l}\right)^{T} \hat{\mathbf{u}}_{A}^{l} = \left(\mathbf{v}_{eA}^{l} - \mathbf{w}_{eA}^{l}\right)^{T} \left(\mathbf{C}_{b}^{l} \hat{\mathbf{u}}_{A}^{b}\right)$$
(3.5.16)

In exactly the same way  $\mathbf{v}_{eG}^{l}$  was expressed in terms of system states in Equation (3.5.12),  $\mathbf{v}_{eA}^{l}$  can be expressed as

$$\mathbf{v}_{eA}^{l} = \mathbf{v}_{e}^{l} + \left(\mathbf{C}_{b}^{l}\boldsymbol{\Omega}_{ib}^{b} - \boldsymbol{\Omega}_{ie}^{l}\mathbf{C}_{b}^{l}\right)\mathbf{r}_{bA}^{b}.$$
(3.5.17)

Assuming  $\mathbf{w}_{eA}^{l}$  as representative of wind velocity  $\mathbf{w}_{eb}^{l}$  acting on the platform and substituting Equation (3.5.17) in Equation (3.5.16) yields

$$V_{\hat{\mathbf{u}}_{A}} = \left(\mathbf{v}_{e}^{l} + \left(\mathbf{C}_{b}^{l}\boldsymbol{\Omega}_{ib}^{b} - \boldsymbol{\Omega}_{ie}^{l}\mathbf{C}_{b}^{l}\right)\mathbf{r}_{bA}^{b} - \mathbf{w}_{eb}^{l}\right)^{T}\left(\mathbf{C}_{b}^{l}\hat{\mathbf{u}}_{A}^{b}\right)$$
(3.5.18)

Finally, corrupting  $\mathbf{V}_{\hat{\mathbf{u}}_A}$  with airspeed error as expressed in Equation (3.4.12) and  $n_A$  measurement noise, and denoting  $\mathbf{w}_{eb}^l$  as  $\mathbf{w}^l$  to simplify notation, airspeed observation model is obtained as following.

$$Z_A = \left(1 + e_{SF}^A\right) \left(\mathbf{v}_e^l + \left(\mathbf{C}_b^l \mathbf{\Omega}_{ib}^b - \mathbf{\Omega}_{ie}^l \mathbf{C}_b^l\right) \mathbf{r}_{bA}^b - \mathbf{w}_{eb}^l\right)^T \left(\mathbf{C}_b^l \hat{\mathbf{u}}_A^b\right) + e_{RW}^A + n_A \qquad (3.5.19)$$

## Chapter 4

# **Fixed-Wing Implementation**

#### Overview

This chapter presents the implementation of the VDM-based navigation proposed in Chapter 3 for a specific fixed-wing platform. Development of the VDM for this platform with models of aerodynamic forces and moments is presented, as well as description of the experimental setup and the navigation filter setup used in Chapters 5 and 6. A considerable amount of wordings and equations throughout this chapter are borrowed

form peer reviewed published papers [36, 37, 38, 39, 41] and an under review paper [40].

#### 4.1 VDM for a Typical Fixed-Wing UAV

As depicted schematically in Figure 4.1, the fixed-wing UAV considered here has a single propeller in front and four (or five) control surfaces; two [coupled] ailerons, an elevator (or two coupled ones), and a rudder. The models for aerodynamic forces and moments for this UAV are borrowed from [19] and presented in Sections 4.1.1 and 4.1.2. These are, however, just sample models for aerodynamic forces and moments, and can be replaced by other models.

#### 4.1.1 Aerodynamic Forces

Four components of aerodynamic forces are recognizable in this model. The thrust force is expressed in body frame (along  $\mathbf{x}^b$  axis), while the lift, lateral, and drag forces are expressed in wind frame.

Thrust force is expressed as

$$F_T = \rho \omega_p^2 D^4 C_{F_T}(J), \tag{4.1.1}$$

where  $\rho$  is the air density,  $\omega_p$  is the propeller speed, D is the propeller diameter, and J is called thrust advance ratio and defined as  $J = \frac{V}{D\pi\omega_p}$  with V being the airspeed magnitude. The dimensionless thrust force coefficient  $C_{F_T}$  is expressed as

$$C_{F_T}(J) = C_{F_T 1} + C_{F_T 2} J + C_{F_T 3} J^2, (4.1.2)$$



Figure 4.1 – Schematic of the fixed-wing UAV used in this research with propeller, aileron(s), elevator(s), and rudder

with  $C_{F_T...}$  s being the thrust force derivatives for the particular UAV at hand.

**Drag force** is expressed as

$$F_r^w = \bar{q}SC_{F_r}(\alpha, \beta), \tag{4.1.3}$$

where S is the wing surface, and  $\bar{q}$ ,  $\alpha$  and  $\beta$  denote dynamic pressure, angle of attack and sideslip angle, respectively, defined in Equation (1.1.17). The dimensionless drag force coefficient  $C_{F_x}$ is expressed as

$$C_{F_x}(\alpha,\beta) = C_{F_x1} + C_{F_x\alpha}\alpha + C_{F_x\alpha_2}\alpha^2 + C_{F_x\beta_2}\beta^2,$$
(4.1.4)

with  $C_{F_{x...}}$  s being the drag force derivatives for the particular UAV at hand.

Lateral force is expressed as

$$F_y^w = \bar{q}SC_{F_y}(\beta), \tag{4.1.5}$$

where the dimensionless lateral force coefficient  $C_{F_y}$  is stated as

$$C_{F_y}(\beta) = C_{F_y 1}\beta,\tag{4.1.6}$$

with  $C_{F_y1}$  being the lateral force derivative for the particular UAV at hand.

Lift force is expressed as

$$F_z^w = \bar{q}SC_{F_z}(\alpha), \tag{4.1.7}$$

where the dimensionless lift force coefficient  $C_{F_z}$  is expressed as

$$C_{F_z}(\alpha) = C_{F_z 1} + C_{F_z \alpha} \alpha, \tag{4.1.8}$$

with  $C_{F_{z}...}$ s being the lift force derivatives for the particular UAV at hand.

The specific force vector  $\mathbf{f}$  is composed of these four components (summed up with necessary rotations applied) divided by the mass of the UAV.

$$\mathbf{f}^{b} = \frac{1}{m} \left( \begin{bmatrix} F_{T} \\ 0 \\ 0 \end{bmatrix} + \mathbf{C}_{w}^{b} \begin{bmatrix} F_{x}^{w} \\ F_{y}^{w} \\ F_{z}^{w} \end{bmatrix} \right)$$
(4.1.9)

The rotation matrix between wind frame and body frame ( $\mathbf{C}_b^w$ ) is defined in (1.1.16) as a function of  $\alpha$  and  $\beta$ .

#### 4.1.2 Aerodynamic Moments

In addition to aerodynamic forces, three components of aerodynamic moments are also recognizable in the model. The dimensionless angular velocities used in expressions for aerodynamic moments are defined as

$$\widetilde{\omega}_x = \frac{b\omega_{l,x}}{2V}, \qquad \widetilde{\omega}_y = \frac{\overline{c}\omega_{l,y}}{2V}, \qquad \widetilde{\omega}_z = \frac{b\omega_{l,z}}{2V},$$
(4.1.10)

with  $\omega_{l,x}$ ,  $\omega_{l,y}$ , and  $\omega_{l,z}$  being components of angular velocity (rotation rate) of body frame with respect to local level frame, expressed in body frame defined as seen in Equation (1.1.30) and repeated here.

$$\boldsymbol{\omega}_{lb}^{b} = \boldsymbol{\omega}_{ib}^{b} - \left(\mathbf{C}_{b}^{l}\right)^{T} \left(\boldsymbol{\omega}_{ie}^{l} + \boldsymbol{\omega}_{el}^{l}\right)$$
(4.1.11)

Roll moment is expressed as

$$M_x^b = \bar{q}SbC_{M_x}(\delta_a, \beta, \tilde{\omega}_x, \tilde{\omega}_z), \tag{4.1.12}$$

where  $\bar{q}$ , S, and  $\beta$  were just mentioned in Section 4.1.1, b is the wing span, and  $\delta_a$  is the normalized aileron deflection ( $\delta_a \in [-1, 1]$ ). The dimensionless roll moment coefficient  $C_{M_x}$  is expressed as

$$C_{M_x}(\delta_a, \beta, \tilde{\omega}_x, \tilde{\omega}_z) = C_{M_x a} \delta_a + C_{M_x \beta} \beta + C_{M_x \tilde{\omega}_x} \tilde{\omega}_x + C_{M_x \tilde{\omega}_z} \tilde{\omega}_z, \qquad (4.1.13)$$

with  $C_{M_x...}$ s being the roll moment derivatives for the particular UAV at hand.

Pitch moment is expressed as

$$M_y^b = \bar{q} S \bar{c} C_{M_y}(\delta_e, \alpha, \tilde{\omega}_y), \tag{4.1.14}$$

where  $\alpha$  was just mentioned in Section 4.1.1,  $\bar{c}$  is the mean aerodynamic chord, and  $\delta_e$  is the normalized elevator deflection ( $\delta_e \in [-1, 1]$ ). The dimensionless pitch moment coefficient is expressed as

$$C_{M_y}(\delta_e, \alpha, \tilde{\omega}_y) = C_{M_y 1} + C_{M_y e} \delta_e + C_{M_y \tilde{\omega}_y} \tilde{\omega}_y + C_{M_y \alpha} \alpha.$$
(4.1.15)

with  $C_{M_{y...}}$ 's being the pitch moment derivatives for the particular UAV at hand.

Yaw moment is expressed as

$$M_z^b = \bar{q}SbC_{M_z}(\delta_r, \tilde{\omega}_z, \beta), \tag{4.1.16}$$

where  $\delta_r$  is the normalized rudder deflection ( $\delta_r \in [-1, 1]$ ). The dimensionless yaw moment coefficient is expressed as

$$C_{M_z}(\delta_r, \tilde{\omega}_z, \beta) = C_{M_z r} \delta_r + C_{M_z \tilde{\omega}_z} \tilde{\omega}_z + C_{M_z \beta} \beta.$$
(4.1.17)

with  $C_{M_z...}$ s being the yaw moment derivatives for the particular UAV at hand.

The moments vector  $\mathbf{M}$  is composed of these three components.

$$\mathbf{M}^{b} = \begin{bmatrix} M_{x}^{b} \\ M_{y}^{b} \\ M_{z}^{b} \end{bmatrix}$$
(4.1.18)

Equation (3.3.2) together with Equations (4.1.1) to (4.1.18) form the VDM for the specific fixed-wing UAV employed in this research that is schematically shown in Figure 4.1.

#### 4.1.3 Air Density Model

Depending on accuracy requirements and fidelity of modeling, the air density may be considered as a constant or as a function of local pressure and temperature according to a suitable model for the application. Local pressure and temperature can theoretically be observed on board or expressed as functions of the altitude as is the case here. For example, according to the International Standard Atmosphere model for low altitude, the air density can be presented as [19]

$$\rho = \frac{p_0 [1 + ah/T_0]^{4.2561}}{R_a T_0},\tag{4.1.19}$$

with  $p_0$  and  $T_0$  denoting local pressure and temperature at the reference altitude, h being the altitude,  $R_a = 287.3m^2K^{-1}s^{-2}$  being the specific gas constant of air, and  $a = -6.5 \times 10^{-3}Km^{-1}$  being the air temperature gradient.

However, in cases that were considered in this thesis, using a fixed value for air density or using the mentioned model did not make a meaningful difference in results. The reason is likely the high correlation between air density and in-flight recalibrated aerodynamic coefficients, which causes the effects of considering a variable air density model to be absorbed by the aerodynamic coefficients to a high extent, especially when considering the limited change in altitude in the trajectories.

#### 4.1.4 Actuator Dynamics

As described in Section 3.4.1, whenever applicable, dynamics of actuators is modeled considering first-order delay, scaling, and shifting, as in Equation (3.4.1). For the employed fixed-wing UAV in this research, there are four actuator states, being the propeller speed n, aileron deflection  $\delta_a$ , elevator deflection  $\delta_e$ , and rudder deflection  $\delta_r$ . The last three, are normalized by maximum deflections, therefore being numbers in the range [-1, 1]. Note that the two ailerons are coupled, which is why only one state is considered for aileron deflection.

#### 4.2 Experimental Setup

This section provides details on the experimental setup used in this research.

#### 4.2.1 Platforms

The UAVs used for this study are custom made fixed-wing UAVs of the same type but in two realizations. They were constructed in "Geodetic Engineering Laboratory TOPO" at EPFL for serving the research in navigation and precise mapping. The first realization (called "TOPO plane 1") taken from [60] is shown in Figure 4.2(a), and the second realization (called "TOPO plane 2") is shown in Figure 4.2(b). Both planes are equipped with the opensource autopilot Pixhawk [50] and have a maximum payload capacity of around 0.8 kg, with operational mass around 2.5 to 2.7 kg. The geometric measures of the two platforms used in the VDM are shown in Table 4.1. Flying endurance is about 45 minutes, and nominal airspeed is around 15 m/s. Thanks to its lightweight construction, the launching can be done from hand and it requires only a small place for landing.



Figure 4.2 – Custom made UAVs, TOPO plane 1 (a) and TOPO plane 2 (b), together with the IMU board (c)

Parameter	Symbol	Value	Unit
Wing span	b	1.630	m
Wing surface	S	0.3439	$m^2$
Mean aerodynamic chord	$\bar{c}$	0.225	m
Propeller diameter	D	0.362	m

Table 4.1 – Geometric measures of the fixed-wing platform

#### 4.2.2 Sensor Setup

The sensor setup is presented here, which is very similar for the two platforms. The stochastic models presented in this section are used in simulations to generate realistic sensor data. These are not necessarily the stochastic models used in the navigation filter, which are detailed in Section 4.3.1.

#### $\mathbf{IMU}$

The inertial measurements come from Navchip MEMS IMUs [34] on a custom made board called Gecko4Nav [42] shown in Figure 4.2(c) with a high resolution barometer. The board can host up to four IMUs, that can be software-combined to an R-IMU<sup>1</sup>. To obtain a realistic stochastic model for IMU errors, an in-house identification was performed, using the novel approach of GMWM introduced in Section 1.3.6. Summary of IMU error parameters from the GMWM analysis and the values provided by the manufacturer are provided in Table 4.2. In this research, values from the GMWM analysis are used.

#### **GNSS** Receiver

Both planes are equipped with multi frequency multi constellation high-end GNSS boards (Javad G3T and Topcon B110 for TOPO plane 1 and TOPO plane 2, respectively) receiving RF signals from an L1/L2 GNSS antenna from Maxtena [49] and provide position and velocity data to the autopilot. Both receivers have RTK<sup>2</sup> capability and store internally raw observations (range, carrier phase, and Doppler measurements). These observations were then post processed using a proprietary software to obtain PPK<sup>3</sup> solution at 10 Hz. Throughout the thesis, the standalone solution at 1 Hz is used for navigation –whether in simulations or experimental tests– unless stated otherwise. PPK solution is used either in experimental scenarios either for calibration phase or to provide the reference for evaluating navigation solutions. Although the processing of raw data provides the confidence levels for position and velocity data at each point, typical error statistics for both PPK and standalone modes can be considered as presented in Table 4.3.

TOPO plane 1 carries also low-cost GNSS receiver/antenna from "u-blox" [76] that is connected to the autopilot, as well.

<sup>&</sup>lt;sup>1</sup>Redundant IMU

<sup>&</sup>lt;sup>2</sup>Real Time Kinematic

<sup>&</sup>lt;sup>3</sup>Post Processed Kinematic

Sensor	Error type	Parameter	Data s	heet $[34]$	CMWM	Unit
			TYP*	MAX**	GMWW	
	Bias	σ	3	15	—	mg
Accelerometers	White Noise	σ	50	83	67	$\mu g/\sqrt{Hz}$
	first order	σ	0.04	0.10	0.15	mg
	Gauss-Markov	Т	_	_	>200	s
	Bias	σ	720	1440	—	$^{\circ}/hr$
Gyroscopes	White Noise	σ	0.003	0.005	0.005	$^{\circ}/s/\sqrt{Hz}$
	first order	σ	5	10	31	$^{\circ}/hr$
	Gauss-Markov	Т	_	_	>200	s
	1	1 ( 1	.1 .	• 11	<u>v</u> [o.4]	

Table 4.2 – MEMS IMU error stochastic model

\*"TYP Specs are mean values or  $1\sigma$  for values that are nominally zero." [34] \*\*"MAX Specs are maximum factory test limits unless otherwise specified." [34]

WAA specs are maximum factory test mints unless otherwise specified. [54

Table 4.3 – Stochastic models for GNSS data errors

CNSS mode	Position er	ror (m)	Velocity erro	or $(cm/s)$		
	Horizontal*	Vertical	Horizontal*	Vertical		
PPK	0.03	0.05	4	5		
Standalone	1	3	4			
*Each of the two horizontal channels						

#### Barometer

A high resolution barometer is available on the Gecko4Nav board together with the IMUs. Raw temperature and pressure observations from the barometer needs to be properly calibrated and converted to altitude data, which can be done separately in a Kalman filter [47]. In this research, however, temperature and pressure data are mapped to altitude with some default constant calibration parameters, leaving a bias (in order of tens of meters) and a scale factor (in order of a few percents) in addition to a white noise with  $\sigma = 0.5 m$ .

#### **Airspeed Sensor**

The airspeed sensor provides dynamic pressure measurements, which are then converted to airspeed observations for internal usage of the autopilot. Airspeed is extracted from the autopilot (with velocity dimension) together with control commands. To determine the error statistics of this sensor, over 3 hours of static data (with expected output of zero) were

#### **Fixed-Wing Implementation**

collected and analyzed. The error identification was performed after removal of the mean value at 2.41 m/s, using the novel approach of GMWM introduced in Section 1.3.6. Summary of airspeed error parameters is provided in Table 4.4.

Error type	Parameter	Value	Unit
White Noise	σ	0.99	m/s
first order	$\sigma$	0.15	m/s
Gauss-Markov	Т	24	s
first order	σ	0.21	m/s
Gauss-Markov	Т	284	s

Table 4.4 – Airspeed error stochastic model

#### 4.2.3 Actuators

The control commands from the autopilot are extracted as PWM<sup>1</sup> values. To be used in the VDM, these values needs to be mapped to steady state values of actuator states. In order to obtain the map, some lab tests were performed, measuring actuator states (propeller speed and deflections of ailerons, elevator, and rudder) in several (at least 5) fixed values of associated commands in PWM from zero to maximum value (in both directions for control surface deflections). Control surface deflections were normalized to the maximum values to lie within the range [-1, 1]. As the two ailerons are coupled, their deflections are superposed. A linear relationship was then established between the commands in PWM and the actuator states in steady condition as *Actuator state* = *Scale* × *PWM* + *Shift*. The scale and shift values for TOPO plane 2 are presented in Table 4.5 with the goodness of fit for linear regression presented as  $R^2$  values.

Table 4.5 – Summary of the map between control commands in PWM and actuator states in steady condition for TOPO plane 2  $\,$ 

Actuator state	Unit	Scale	Shift	$R^2$
Propeller speed	[rad/s]	1.15	-1240	0.957
Aileron deflection	[]	$-3.20\times10^{-3}$	4.87	0.999
Elevator deflection	[]	$3.02\times10^{-3}$	-4.63	0.997
Rudder deflection	[]	$3.31 \times 10^{-3}$	-4.94	0.962

#### 4.3 Filter Setup

In this section, main aspects of the setup of the navigation filter are detailed.

<sup>&</sup>lt;sup>1</sup>Pulse Width Modulation

#### 4.3.1 Stochastic Models for Observations

Unless stated otherwise, the stochastic models for observations within the navigation filter, used in both simulation and experimental scenarios in Chapters 5 and 6, are the ones presented in this section.

#### **Inertial Measurements**

In navigation on both simulation and experimental data, IMU error was first modeled as a random walk and a first order Gauss-Markov process. Inspecting Table 4.2 for IMU error model, one can see that the time-correlated errors are a bias and a first order Gauss-Markov process with a correlation time of 200 s. Considering the duration of flight, 200 s seems long enough for the first order Gauss-Markov process to have a similar signature to a random walk process. On the other hand, a random walk process naturally covers also a bias (as a non-zero mean process) when its initial uncertainty is set to a non-zero value (as discussed in Section 3.4.4). Therefore, navigation was repeated by replacing the bias and the first order Gauss-Markov process with only a single random walk process in the filter. This covered both the bias and the first order Gauss-Markov process with "long" correlation time and leaded to practically the same results. As this approach reduces the number of states in the filter without affecting the performance, it was preferred and implemented afterwards. The values of error model parameters following from GMWM analysis are summarized in Table 4.2.

#### **GNSS** Position and Velocity

The error for GNSS position and velocity data is considered as white noise with the statistics presented in Table 4.3 for both PPK and standalone modes.

#### **Barometric Altitude**

The error statistics used in the filter are the same as presented in Section 4.2.2. The white noise is considered with  $\sigma = 0.5 m$ , and the bias and scale factor are either removed before navigation, or estimated within the filter thanks to the availability of GNSS position data during the initial part of the flight.

#### Airspeed

According to the error model for the real sensor presented in Section 4.2.2, apart from the bias, a white noise and two first order Gauss-Markov processes are recognized for the sensor. Considering the relatively small powers of the two Gauss-Markov processes, only a bias with initial uncertainty of 2 m/s and a white noise with  $\sigma = 1 m/s$  is considered within the filter.

#### 4.3.2 Initialization

#### **Navigation States**

As the VDM in this research was developed for the flying phase (considering only the aerodynamic forces and moments), it cannot be used to predict navigation states when the UAV is not flying (e.g., before take-off). This means that the VDM-based navigation filter needs to be initialized "in the air" when the UAV is flying. The following procedure is followed for initialization.

- 1. Navigation states (position, velocity, and attitude) are initialized for INS-based navigation.
- 2. INS-based navigation is performed until flight phase is detected. This detection can be conservative, as the passage to VDM-based navigation does not need to be immediate.
- 3. The VDM based navigation is activated. This requires that the navigation states provided by INS-based navigation, and possibly auxiliary states such as IMU errors, are transferred as initial conditions for the VDM-based navigation.

It is worth mentioning that as currently the navigation is performed in post process mode, VDM-based navigation is activated manually. In online implementation, though, the flying condition should be detected automatically [61] to activate VDM-based navigation. Otherwise, the activation will require an input from a human user during the flight.

Unless stated otherwise, the considered initial uncertainties for navigation states in Chapters 5 and 6 are as following. For position and velocity, initial uncertainties correspond to GNSS observations uncertainty presented in Table 4.3. In cases where GNSS velocity observations were not used, an initial uncertainty of 1 m/s was considered for each velocity component. The initial attitude uncertainty was considered at 4° for roll and pitch and 6° for yaw, while  $2^{\circ}/s$  of initial uncertainty was considered for each angular velocity component.

#### VDM Parameters

The nominal values for VDM parameters (aerodynamic coefficients) used in simulations in Chapters 5 and 6 follow from [19] and are presented in Table 4.6. Mass and geometry related parameters for both TOPO plane 1 and TOPO plane 2 are measured in the lab and presented in Section 4.2.1. In simulations, a random error with standard deviation of 10% was applied to the nominal values, unless stated otherwise. These nominal values are also used as the starting point for calibration in experimental scenarios.

#### **Other States**

For wind velocity, sensor errors, and sensor mounting misalignment, initial values of zero were considered, unless stated otherwise. Initial uncertainty for sensor errors were considered in accordance with sensor error statistics presented in Section 4.2.2. Values not provided in this section are presented separately for each case in Chapters 5 and 6.

Parameter	$C_{F_T 1}$	$C_{F_T2}$	$C_{F_T3}$	$C_{F_x1}$	$C_{F_x \alpha}$	$C_{F_x \alpha 2}$	$C_{F_x\beta 2}$
Value	0.0842	-0.136	-0.928	-0.0212	-0.0266	-1.55	-0.401
Parameter	$C_{F_y1}$	$C_{F_z 1}$	$C_{F_z \alpha}$	$C_{M_x a}$	$C_{M_x\beta}$	$C_{M_x \tilde{\omega}_x}$	$C_{M_x \tilde{\omega}_z}$
Value	-0.379	0.0129	-3.25	0.0679	-0.0130	-0.192	0.0361
Parameter	$C_{M_y1}$	$C_{M_y e}$	$C_{M_y \tilde{\omega}_y}$	$C_{M_y \alpha}$	$C_{M_z r}$	$C_{M_z \tilde{\omega}_z}$	$C_{M_z\beta}$
Value	0.0208	0.545	-9.83	-0.0903	0.0534	-0.214	0.0867

Table 4.6 – Nominal values of VDM parameters (aerodynamic coefficients) [19]

#### 4.3.3 Model Linearization for EKF

#### Manual Linearization

As stated in Chapter 3, an EKF is used in this research as the estimator. This implies that the linearized versions of both process model and observation model is required. In the early development stages when the non-rotating flat Earth model was considered and fewer augmented states were included, the linearization was done manually to come close to the most algebraically simplified and compact linearized version in favor of easier implementation and lower computational cost. Considering the complexity of the models and high number of the states, such a manual task is laborious and results in long expressions and implementation/verification process. To reflect the complexity and hint to the motivation behind automation of this process, manual linearization of the main part of a simplified VDM is reported in Appendix A. It is worth mentioning that the validity of the manual linearization was ensured using symbolic toolbox of MATLAB.

#### Automated Linearization

As some simplifications in modeling were dropped and the state vector was further augmented to include new effects/parameters, new sensors, and more detailed observation models, the effort for manual linearization became impractical. Therefore, an object oriented MATLAB software was developed to design and construct the filter entirely, including linearized models, in an automated manner with minimum input required from the user. In fact, the user only provides the bare minimum information required from theoretical point of view to design the filter, including the process and observation models in a symbolic manner. Such a setup, allows for extreme flexibility in filter design and facilitates testing different scenarios.

As an example, all the required input from the user to define main process and observation models in construction of a basic INS-GNSS navigation filter is presented as the following MATLAB script.

Required user input in automated construction of a basic INS/GNSS navigation filter

```
1 %% Process Model: INS
2 syms phi lambda h v_N v_E v_D q0 q1 q2 q3 real
3 syms f_x f_y f_z omega_x omega_y omega_z real
4 syms B_I1 B_I2 B_I3 L_I1 L_I2 L_I3 real
5 syms e_rw_a1 e_rw_a2 e_rw_a3 e_gm1_a1 e_gm1_a2 e_gm1_a3 e_gm2_a1 e_gm2_a2 ...
      e_gm2_a3 real
6 syms e_rw_g1 e_rw_g2 e_rw_g3 e_gm1_g1 e_gm1_g2 e_gm1_g3 e_gm2_g1 e_gm2_g2 ...
      e_gm2_g3 real
  e_a = [e_rw_a1 e_rw_a2 e_rw_a3].' + [e_gm1_a1 e_gm1_a2 e_gm1_a3].' + ...
      [e_gm2_a1 e_gm2_a2 e_gm2_a3].';
  e_g = [e_rw_g1 e_rw_g2 e_rw_g3].' + [e_gm1_g1 e_gm1_g2 e_gm1_g3].' + ...
      [e_gm2_g1 e_gm2_g2 e_gm2_g3].';
9 f_I = [f_x f_y f_z].';
10 omega_iI_I = [omega_x omega_y omega_z].';
11 C_b_I = C_elementary(B_I1,1) * C_elementary(B_I2,2) * C_elementary(B_I3,3);
12 r_bI_b = [L_I1 L_I2 L_I3].';
13 f_b = C_b_I.' * ((f_I-e_a) - (SkSymMat(omega_iI_I)*SkSymMat(omega_iI_I)) * ...
      C_b_I*r_bI_b);
14 omega_ib_b = C_b_I.' * (omega_iI_I-e_g);
15 r_e_l = [phi; lambda; h];
16 v_e_l = [v_N; v_E; v_D];
17 q_b_l = [q0; q1; q2; q3];
18 R_M = WGS84.a * (1 - WGS84.e<sup>2</sup>) / (1 - (WGS84.e*sin(phi))<sup>2</sup>)<sup>(3/2)</sup>;
19 R_P = WGS84.a / sqrt(1 - (WGS84.e*sin(phi))^2);
20 D_inv = diag([1/(R_M+h), 1/((R_P+h)*cos(phi)), -1]);
21 dot__r_e_l = D_inv * v_e_l;
22 dot__phi = dot__r_e_l(1);
23 dot_lambda = dot_r_e_l(2);
24 g_l = [0; 0; WGS84_gravity(WGS84, phi, h)];
25 C_b_1 = quat2dcm(q_b_1);
26 omega_ie_l = [WGS84.w * cos(phi); 0; -WGS84.w * sin(phi)];
27 omega_el_l = [dot_lambda * cos(phi); -dot_phi; -dot_lambda * sin(phi)];
28 omega_lb_b = omega_ib_b - C_b_l.' * (omega_ie_l+omega_el_l);
29 dot__q_b_l = 1/2 * quatMult(q_b_l,[0; omega_lb_b]);
30 dot_v_e_l = C_b_l * f_b + g_l - (SkSymMat(omega_el_l) + ...
      2*SkSymMat(omega_ie_l)) * v_e_l;
31 INS_model_data.X_sym = [r_e_l; v_e_l; q_b_l];
32 INS_model_data.dX_sym = [dot_r_e_l; dot_v_e_l; dot_q_b_l];
33 INS_model_data.ModelName = 'INS';
34 INS = SubModel(INS_model_data);
35 %% Observation Model: GNSS
36 syms L_G1 L_G2 L_G3 real
37 r_bG_b = [L_G1 L_G2 L_G3].';
38 GNSS_position = [phi; lambda; h] + D_inv*C_b_l*r_bG_b;
39 GNSS_velocity = [v_N; v_E; v_D] + ...
      (C_b_1*SkSymMat(omega_ib_b)-SkSymMat(omega_ie_1)*C_b_1)*r_bG_b;
40 GNSS_model_data.obsMdl = [GNSS_position; GNSS_velocity];
41 GNSS_model_data.name = 'GNSS';
42 GNSS = Sensor(GNSS_model_data);
```

# Part III

# **Results and Analyses**

## Chapter 5

## Main Results

#### Overview

This section provides simulation and experimental results that demonstrate performance of the proposed VDM-based navigation system. The results are organized in 4 sections, each verifying a different aspect of navigation performance and reliability. A short description of these 4 sections is represented in the following list, with a summary in Table 5.1. This list also provides references to previously published papers on current research from which most wordings and results are borrowed.

Section 5.1: Proof of concept was done via Monte-Carlo simulation. There were 100 runs with different realizations of random terms on sensor errors, initialization errors, and VDM parameters error. The simulated flight follows a trajectory with ascending, 90 degree turns, level flight, and descending. Most wordings and results in this section are taken from two peer reviewed papers [37, 38].

Section 5.2: Effects of wind on navigation performance was evaluated via Monte-Carlo simulation using real 3D wind data [43]. Wind is a major factor affecting the behavior of the UAV, and its estimation is critical in VDM predictions. There were 50 runs with different 3D wind velocity profiles applied on the UAV. Wind velocity came from real data [43]. Most wordings and results in this section are taken from a peer reviewed paper [39].

Section 5.3: To validate the simulations and evaluate real-world performance of VDMbased navigation, some experiments were performed using TOPO plane 1 UAV, described in Section 4.2.1. The results of one of the flights are presented here. Data collected form the real flight were processed with VDM-based navigation system. Then, the real trajectory was used to generate way-points in a Monte-Carlo simulation to emulate reference data and sensor data for navigation. Similar to Section 5.1, there were 100 runs with different realizations of random terms on sensor errors, initialization errors, and VDM parameters error. Comparison between the navigation results in experimental scenario and emulation scenario reveals the level of validity of simulations. Most wordings and results in this section are taken from a peer reviewed paper [41] and an under review paper [40].

Section 5.4: To further evaluate VDM-based navigation performance, more experiments were performed using a second UAV, TOPO plane 2, described in Section 4.2.1. The results of

two of these flights are presented here. The first flight, containing some high dynamics parts in the trajectory that was used to estimate the VDM parameters in the navigation system. Navigation was then performed on the second flight with a area mapping type trajectory to estimate navigation performance during GNSS outages.

Section	Main Goal	Type	Main variations in Monte-Carlo	Trajectory	UAV
5.1	proof of concept	simulation	sensor errors, initialization errors, VDM parameters error	ascending, 90 degrees turns, level flight, and descending	synthetic
5.2	evaluation of wind effects	simulation	wind velocity	same as 5.1	synthetic
5.3	validation of simulations	emulation & experiment	same as 5.1	area mapping type	TOPO plane 1
5.4	further experiments	experiment	_	high dynamics for calibration & area mapping type	TOPO plane 2

	Table $5.1 -$	Summarv	of	results	section	organization
--	---------------	---------	----	---------	---------	--------------

#### **General Remarks**

Unless stated otherwise, stochastic models to simulate sensor error come from Section 4.2.2, and filter setup, such as initialization and observation error statistics, is in accordance with the details presented in Section 4.3.

For better interpretation of results, position is presented in Cartesian coordinates in a local level frame  $\text{ENU}^1$  fixed at some point close to/coinciding the initial position, and attitude is presented by Euler angles (roll, pitch, and yaw).

The main focus of the results presented in this section is the navigation system performance during GNSS outages of a few minutes. However, the added value of the navigation system in precise attitude estimation for photogrammetry/mapping applications was investigated via Monte-Carlo simulations, whose results were published in a peer reviewed paper [36].

 $<sup>^{1}{\</sup>rm East-North-Up}$ 

#### 5.1 Proof of Concept

This section presents some simulation results, performed primarily as a proof of concept that the proposed VDM-based navigation system works and has potential to significantly improve the accuracy of autonomous navigation with respect to conventional INS-based navigation. A minimum sensor setup was used in this simulation, comprising an IMU and GNSS position observations. Most wordings and results in this section are taken from a previously published paper [38].

The Monte-Carlo simulation was performed with 100 runs, using real 3D wind velocity data [43]. While the trajectory and the wind were kept the same in each realization, the error in observations, initialization, and VDM parameters changed randomly for each individual run. Figure 5.1 depicts the reference trajectory, as well as the solution from a sample run. The trajectory has an approximate ground footprint of  $2 \ km \times 2 \ km$  and a total change of 1km in altitude. Detailed results are presented in sections 5.1.1 and 5.1.2, and a discussion on observability is provided in Section 5.1.3.



Figure 5.1 – Reference trajectory and the solution from a sample run with GNSS signals available during first 100s only (adapted from [38])

#### 5.1.1 Navigation States

The position error is presented in Figure 5.2 for all the 100 Monte-Carlo runs. The graphs show how the error grows as time passes after GNSS outage starts, and how the overall behavior is similar for individual runs. An empirical RMS<sup>1</sup> is calculated from these individual errors and plotted against the predicted confidence level  $(1\sigma)$  in the same figure.



Figure 5.2 – Position errors for all the 100 Monte-Carlo runs with empirical RMS and predicted confidence level (adapted from [38])

Figure 5.3 depicts the empirical RMS of attitude errors for all the 100 runs, with associated predicted values of confidence  $(1\sigma)$ . In order to avoid clattering the graph, error values are depicted at discrete times with an interval of 40 seconds. First thing to notice is how closely (and slightly conservatively) the error is predicted within the filter, which reveals the relevance of stochastic setup. The results are promising in terms of preserved navigation accuracy as well, with the RMS of position error being less than 110m after 5 minutes of autonomous navigation during GNSS outage. This error is only  $0.007^{\circ}$  for roll,  $0.020^{\circ}$  for pitch, and  $0.38^{\circ}$  for yaw. In comparison, the classical INS coasting would result in errors of 11.7km for position,  $2.6^{\circ}$  for roll,  $1.5^{\circ}$  for pitch, and  $16.6^{\circ}$  for yaw under exactly the same situations.

Figure 5.4 represents the comparison of the RMS of position and yaw errors for all the 100 runs between proposed VDM-based and INS-based navigation approaches over the whole interval. The improved performance of the proposed filter is noticeable also during the availability of GNSS in estimating yaw.

#### 5.1.2 Auxiliary States

Figure 5.5 shows the empirical RMS of errors in estimation of IMU error terms for all the 100 runs, which were modeled as random walk processes. The predicted confidence values  $(1\sigma)$  are again close to empirical RMS, and the accuracy of the estimation is satisfactory in comparison to the amplitude of simulated errors (realization values) plotted on the same graph.

<sup>&</sup>lt;sup>1</sup>Root Mean Square



Figure 5.3 – RMS of attitude errors from 100 Monte-Carlo runs (adapted from [38])



Figure 5.4 – Comparison between INS-based and VDM-based navigation: RMS of position and yaw errors from 100 Monte-Carlo runs (adapted from [38])



Figure 5.5 - RMS of estimation error for IMU error terms from 100 Monte-Carlo runs, together with the amplitude of simulated errors (adapted from [38])

The empirical RMS of mean error in estimation of VDM parameters for all the 100 runs is plotted in Figure 5.6. Although the prediction of confidence levels of estimation by the filter is slightly optimistic here, the difference does not exceed 24%. The evolution of this error reveals that there is a sharp decrease in the mean error of parameter estimation from the initial 10% error to 6% during the first 40 seconds with GNSS available, which is followed by a slowly decreasing trend. The reason behind the second regime is the correlation between some parameters within the set. In such situations, some parameters are estimated as a group (in some sense) rather than individually, and the remaining individual errors contribute to increasing the mean error for the whole set. More details on this can be found in section 5.1.3 when discussing the observability of the states. Note that the mean error is calculated in an RMS sense for all the parameters at any given time to be stricter and to cover both the mean value and the standard deviation.



Figure 5.6 – RMS of estimation error for VDM parameters from 100 Monte-Carlo runs (adapted from [38])

Finally, the RMS of error in estimation of wind speed for all the 100 runs is depicted in Figure 5.7. Again, the filter has been able to provide a close and slightly conservative prediction on the estimation error, and the accuracy of the estimation is good enough for navigation purposes, especially when considering the fact that no airspeed sensor was considered in the system and the estimation is provided just as a courtesy of the UAV dynamic model provided with IMU data. As revealed by the slow growth of predicted error, the wind speed estimation uncertainty is expected to increase as time passes during GNSS outage. However, the rate of this growth is well controlled, reaching only 0.7m/s after 5 minutes of GNSS outage. It is worth noting that all the three components of the wind velocity are estimated within the filter, which are converted to the wind speed when presenting results in favor of the plot being less clattered.



Figure 5.7 – RMS of estimation error for wind speed from 100 Monte-Carlo runs, together with real wind speed [43] (realization values) (adapted from [38])

#### 5.1.3 Discussion on Observability

This section discusses some aspects of the observability of filter states. The discussion will be based on analysis of covariance matrix throughout the time. Close agreement between the empirical errors and the predicted confidence levels through covariance matrix as revealed in sections 5.1.1 and 5.1.2, provides the validity of stochastic parameters setup in the filter, so that the subsequent discussion based on analyzing the covariance matrix can take place.

One should note that although the plots presented in this section are related to one sample Monte-Carlo run, the stochastic aspects discussed here are very similar for all the 100 runs.

Figure 5.8 shows the ratios of uncertainties of states during GNSS presence period (top) and GNSS outage period (bottom). During GNSS presence period, the uncertainties of all the states except a few of VDM parameters were reduced significantly, which indicates the observability of them, even during this relatively short availability period (100 seconds) for stand-alone satellite positioning. The uncertainties of those few VDM parameters were not increased either, which indicates no divergence is expected on their estimation.

As expected, during GNSS outage (bottom part of Figure 5.8), the uncertainty of position and velocity has grown. The uncertainty of attitude was reduced by a factor of 0.3 for roll, and increased by factors of only 1.2 and 2.3 for pitch and yaw, respectively, which is considered an excellent performance in attitude determination after 5 minutes of autonomous navigation. The uncertainty of rotation rates was reduced, thanks to direct measurements from the IMU. The uncertainties of all VDM parameters were kept more or less constant during GNSS outage, which means the filter has been able to maintain the estimation of them even in the absence of GNSS measurements. Conceptually, this is achieved by the redundancy of information on linear accelerations and rotation rates provided by VDM and IMU. The uncertainty of IMU error terms was almost unchanged, as well. Finally, the uncertainty of wind velocity has

experienced an increase of almost 4 times, which is still a quite useful result for navigation. Further improvement can also be expected with the use of airspeed sensor observations, as long as the uncertainty of such observations are low with respect to the uncertainty of estimated values by the filter without them.



Figure 5.8 – Ratios of uncertainties of states during GNSS presence and GNSS outage periods (navigation starts with GNSS available at t = 0s, GNSS outage begins at t = 100s, and navigation ends at t = 400s) (adapted from [38])

Figure 5.9 depicts the correlation coefficients among all the states within the filter at the end of the navigation. As can be seen, the navigation, IMU error, and wind velocity states are well decorrelated in general, which indicates their good observability. The VDM parameters block shows the highest level of correlations, especially among different aerodynamic coefficients for any specific moment component (this is distinguishable as almost all white small square blocks on the diagonal). This means that rather than individual VDM parameters, groups of VDM parameters are getting estimated within the filter. Decorrelating individual VDM parameters demands longer time, more dynamic maneuvers, and possibly longer periods of GNSS availability. Considering the fact that these are practically constants for a particular platform, their better estimation can be carried only once during a calibration scenario. This shall result in lower uncertainties of their initial values for subsequent scenarios. Anyway, for

the purpose of navigation, and not necessarily parameter estimation, Monte-Carlo simulations reveal that the VDM parameters are estimated well enough to enable the proposed VDM-based navigation reaching much better performance compared to inertial coasting.

The other point to notice is how well the VDM parameters are decorrelated from other states of the filter. There remains also a considerable level of correlation between wind velocity components and position and velocity of the UAV. As depicted in Figure 5.7, the uncertainty of wind velocity estimation grows steadily, yet slowly as time passes during GNSS outage. This is in relation to the level of process noise on the wind model. When gustier conditions are expected, higher values on this process noise can be used to allow capturing more sudden changes in wind velocity, at the price of increasing the uncertainty of navigation solution during GNSS outage, as well. However, including airspeed observations might be a good answer to such compromise.



Figure 5.9 – Correlation matrix at the end of navigation (adapted from [38])

Last but not least, as the observability concerns during GNSS outage are mainly related to VDM parameters and they are expected to be more or less constant under normal condition, one may exclude such parameters from estimation during GNSS outages. This means that the last estimated values of these parameters are kept constant and used within the filter. To be extremely cautious, the same can be done for IMU error terms, although the results presented in this section do not hint at necessity of such action.

#### 5.2 Evaluation of Wind Effects

This section presents some simulation results, performed to evaluate the effects of wind velocity on performance of the proposed VDM-based navigation system. The sensor setup, flight plan (waypoints), and error statistics are the same as those used in Section 5.1. Most wordings and results in this section are borrowed from a peer reviewed published paper [39].

In Section 5.1 (and [38]), 100 runs were performed using the same trajectory and realization of time-varying wind velocity, while the sensor errors and initialization errors changed randomly in each individual run (Called "case A" throughout this section). Here in contrary, 50 runs have been performed with wind velocity, coming from real 3D data [43], changing in each run while keeping one representative realization of sensor errors and initialization errors for all the runs (Called "case B" throughout this section). The selected realization of sensor errors and initialization errors here is the one with closest position error to the RMS of all the 100 runs in case A. To investigate further the effect of wind power on navigation performance, the 50 runs of case B have been performed again with scaled up (2 times) real wind velocity. The process model noise within the filter have been modified accordingly once (Called "case C" throughout this section), and kept unchanged another time (Called "case D" throughout this section). Table 5.2 presents a summary of mentioned Monte-Carlo simulation cases.

	Variations on			
Case	Sensor and initialization errors	Wind velocity data	Scaled up wind velocity	Modified process noise
A [38]	Х			
В		Х		
С		Х	Х	Х
D		Х	Х	

Table 5.2 – Summary of Monte-Carlo simulation cases (adapted from [39])

Although the trajectory may be slightly different in each realization here due to changes in wind velocity, the way-points has been kept the same, so all the 50 trajectories are close to each other, corresponding to the one depicted in Figure 5.1.

Table 5.3 reveals some statistics of the 50 wind velocity realizations for case B. Turbulence intensity at each direction is defined as the RMS of eddy velocity (equivalent to standard deviation of velocity) at that direction divided by mean wind speed. Turbulence kinetic energy is defined as the mean kinetic energy per mass due to eddy velocity. Eddy velocity is the difference between instantaneous and mean wind velocity.

The results of Monte-Carlo simulations are presented in Sections 5.2.1 and 5.2.2.

#### 5.2.1 Navigation States

Figure 5.10 shows comparison of RMS of position and yaw errors for all the 50 runs between proposed VDM-based and classical INS-based navigation systems over the whole 400 s interval

Characteristic	Minimum	Maximum	Mean	RMS
Mean Wind Speed	2.0	5.0	3.5	3.6
Streamwise Turbulence Intensity	0.017	0.097	0.042	0.046
Transversal Turbulence Intensity	0.014	0.072	0.036	0.038
Vertical Turbulence Intensity	0.004	0.032	0.013	0.014
Turbulence Kinetic Energy	0.003	0.057	0.022	0.026
[j/kg]				

Table 5.3 – Wind velocity statistics for case B (adapted from [39])

for case B. The availability of GNSS positioning is assumed only during the first 100 s, while the remaining 300 s is left to autonomous navigation. While the RMS of position error is 14.3 km for classical INS coasting after 5 minutes (300 s), this is reduced to less than 86 m with VDM-based navigation under exactly the same situations. This represents an improvement of more than two orders of magnitude in position accuracy. The attitude determination also shows an improvement of 1 to 2 orders of magnitude, which will be detailed shortly. It is worth mentioning here that the improved estimation of yaw is noticeable with the proposed filter also during the availability of GNSS observations.



Figure 5.10 – Comparison between INS-based and VDM-based navigation: RMS of position and yaw errors from 50 Monte-Carlo runs (case B) (adapted from [39])

The position error for all the 50 Monte-Carlo runs is presented in Figure 5.11 for case B. The graphs show how the error grows as time passes after GNSS outage starts, and how the overall behavior is more or less similar for individual runs. An empirical RMS is calculated from these individual errors and plotted against the predicted confidence level  $(1\sigma)$ . The

close correspondence between empirical and predicted confidence levels of positioning reveals relevant stochastic assumptions in filter setup.



Figure 5.11 – Position errors for all the 50 Monte-Carlo runs (case B) (adapted from [39])

Figure 5.12 depicts the empirical RMS of attitude errors for all the 50 runs, with associated predicted values of confidence  $(1\sigma)$  for case B. The results are promising in terms of preserved navigation accuracy, with the RMS of error to be only  $0.04^{\circ}$  for roll,  $0.08^{\circ}$  for pitch, and  $1.34^{\circ}$  for yaw after 5 minutes of autonomous navigation during GNSS outage. In comparison, the classical INS coasting would result in errors of  $3.36^{\circ}$  for roll,  $1.77^{\circ}$  for pitch, and  $26.87^{\circ}$  for yaw under exactly the same situations.

For all the cases, A to D, the maximum RMS of errors after 5 minutes of autonomous navigation during GNSS outage is summarized in Table 5.4. As can be seen in this table, in cases C and D where the wind speed was two times bigger, the error in navigation states experienced also an increase of almost two time. This is considered here as a positive result, since the navigation error grows only proportionally against wind speed, yet stays significantly smaller compared to INS coasting. As mentioned at the beginning of Section 5.2, the only difference between cases C and D is the modification to the process model noise within the filter to adapt to the change in wind speed. This modification was done in case C and avoided in case D. The navigation errors were quite comparable in these two cases, revealing low sensitivity to the stochastic setup of wind model within the filter. The main difference was in the predicted confidence levels  $(1\sigma)$ . While in cases A, B, and C, they were in good agreement with actual errors, the predicted confidence levels were too optimistic (by a factor of 2) with respect to the actual errors in case D.

#### 5.2.2 Auxiliary States

Successful estimation of auxiliary states is a key enabler of navigation improvement within the filter. Due to general similarity of results to case A, the results for case B are presented in



Figure 5.12 – RMS of position and attitude errors from 50 Monte-Carlo runs (case B) (adapted from [39])

Case	Navigation	RMS of errors					
Case	ravigation	position	roll [°]	pitch [°]	yaw [°]		
Δ	INS-based	$11.7 \mathrm{~km}$	2.62	1.47	16.61		
A	VDM-based	$110 \mathrm{~m}$	0.01	0.02	0.38		
в	INS-based	$14.3 \mathrm{~km}$	3.35	1.77	26.87		
D	VDM-based	86 m	0.04	0.08	1.34		
С	INS-based	$14.1 \mathrm{km}$	2.49	2.80	25.10		
U	VDM-based	$185 \mathrm{~m}$	0.09	0.15	2.20		
σ	INS-based	14.1 km	2.49	2.80	25.10		
	VDM-based	$199 \mathrm{~m}$	0.11	0.14	2.24		

Table 5.4 – Summary of maximum navigation errors for cases A to D [39]

this section in a less detailed manner. All the reported values were calculated as an RMS of the values for all the 50 Monte-Carlo runs for case B.

Figure 5.13 shows the empirical RMS of errors in estimation of IMU error terms for all the 50 runs. The time correlated part of the IMU error gets estimated quickly during the first tens of seconds of navigation and remains rather unchanged afterwards, even during the GNSS outage period. The estimation error has an average of 4.9% for the three accelerometers and an average of 5.2% for the three gyroscopes at the end of the whole navigation period.



Figure 5.13 - RMS of estimation error for IMU error terms from 50 Monte-Carlo runs, together with the amplitude of simulated errors

The mean error in estimation of VDM parameters shows a sharp decrease from the initial value of 9.13% to 5.86% during the first 40 seconds with GNSS available, which is followed by a slowly decreasing trend until the end. The reason behind the second regime is the correlation between some parameters within the set. In such situations, the groups of parameters are estimated rather than individual parameters, and those individual errors contribute to increasing the mean error for the whole set.



Figure 5.14 – RMS of estimation error for VDM parameters from 50 Monte-Carlo runs

Finally, the wind velocity is estimated well during GNSS availability period, reaching an error of only 5.1% for wind speed after 100 seconds. As depicted in Figure 5.15, the estimation error starts to grow when GNSS outage begins. However, the rate of this growth is well controlled, and the error is still below 12.7% after 5 minutes of GNSS outage.

#### 5.3 Validation of Simulations

This sections presents the results of one of the experiments performed to validate the simulations. The flight was performed with TOPO plane 1, described in Section 4.2. While only the standalone GNSS data were used in navigation, PPK GNSS data provided cm-level position information ad reference for evaluation of navigation performance. Another use for the high accuracy GNSS data was generating waypoints for the emulation scenario. Similar to Section 5.1, a Monte-Carlo simulation was performed on emulated sensor data with 100 runs each having a different realization of random terms on sensor errors, initialization errors, and VDM parameters error. As presented in Section 5.3.1 and 5.3.2, closeness of navigation results in simulation and experimental scenarios reveals the realistic setup and validity of the former. Most wordings and results in this section are taken from a peer reviewed paper [41] and an under review paper [40]

As real data are involved in experiments in this section, Earth rotation and curvature were considered in navigation, which was not the case in synthetic scenario presented in Sections 5.1 and 5.2. There, a flat non-rotating Earth model was considered for both generating the reference and sensor data as well as the navigation. Therefore, no considerable difference was expected by ignoring Earth rotation and curvature, there. This was also confirmed in some complementary tests.



Figure 5.15 - RMS of wind estimation errors from 50 Monte-Carlo runs (case B) (adapted from [39])

Barometric altimeter measurements were used in all simulation and experimental results, for both conventional INS-based and proposed VDM-based navigation systems, unless stated otherwise.

In order to make the emulation scenario as close as possible to the experimental one, the following points were considered:

- The waypoints used in flight simulation were taken from the high accuracy GNSS data in experimental trajectory. Then, the flight simulation was run to fly through these waypoints and emulate reference data and sensor data for Monte-Carlo simulation. As can be seen in Figure 5.16 and 5.22, the trajectories for simulation and experimental scenarios were very similar and visually indistinguishable, at least in terms of position.
- The wind data used in the above-mentioned flight simulation came from real wind measurements [43].
- The values of VDM parameters used in simulations were the same as the ones related to the real platform used in experimental scenario navigation.
- The sensor errors used in the Monte-Carlo emulation scenario were generated by stochastic models that were identified for the real sensors onboard the real plane in static conditions, as described in Section 4.2.2.
- The sensor error stochastic models that were used in the navigation filter were the same for both simulation and experimental scenarios. These are simplified versions of what was obtained from sensor identification. More details are provided in Section 4.3.1.

#### 5.3.1 Emulation Scenario

A Monte-Carlo simulation has been performed with 100 runs, using real 3D wind velocity data [43]. While the trajectory and the real wind velocity has been kept the same in all realizations, the errors in observations, initialization, and VDM parameters have changed randomly for each individual run (realization).

Figure 5.16 provides horizontal view of the reference trajectory, as well as the solutions from a sample run of Monte-Carlo simulation. The trajectory emulates that of the experimental flight. It has an approximate ground footprint of  $1000 \ m \times 500 \ m$  and about 50 m change in altitude. The sensor setup is also simulated as close as possible to the one used in the real flight. Detailed results of Monte-Carlo simulations are presented in Sections 5.3.1 and 5.3.1.



Figure 5.16 – Horizontal view of reference trajectory in Monte-Carlo simulations and navigation solutions from a sample run with GNSS signals available during first 100s only (adapted from [40])

#### **Navigation States**

Figure 5.17 depicts position errors for all the 100 Monte Carlo runs, as well as the RMS of them and predicted uncertainty by the filter  $(1\sigma)$ . As can be seen, during 3 minutes of GNSS outage, maximum RMS of position errors for VDM-based navigation reached only 38 m and this error is in good agreement with filter predicted confidence, which confirms
proper stochastic configuration. Under exact same emulation scenario, the error for INS-based navigation was 681 m, almost 18 times larger.



Figure 5.17 – Position error in Monte-Carlo simulation for VDM-based navigation, with maximum RMS of 38 m during 3 minutes of GNSS outage. The same error for INS-based navigation was 681 m (adapted from [40]).

RMS of attitude errors for all the 100 Monte Carlo runs, as well as predicted uncertainty  $(1\sigma)$  are depicted in Figure 5.18. As can be seen, during 3 minutes of GNSS outage, maximum RMS of errors for VDM-based navigation reached only  $0.20^{\circ}$ ,  $0.19^{\circ}$ , and  $0.96^{\circ}$  for roll, pitch, and yaw, respectively. The attitude uncertainty prediction in filter  $(1\sigma)$  was conservative slightly for roll and pitch and rather considerably for yaw. This may be related to over-pessimistic setup of process model noise in rotational dynamics part. Under exact same emulation scenario for INS-based navigation, maximum RMS errors for roll, pitch, and yaw were  $0.67^{\circ}$ ,  $0.68^{\circ}$ , and  $7.28^{\circ}$ , respectively.

Discarding barometer data made navigation errors considerably larger for INS-based navigation as errors in vertical channel were no longer limited, while it had very little to no impact on VDM-based navigation results. A summary of navigation errors with and without barometer data is provided in Table 5.5.

#### **Auxiliary States**

Figure 5.19 shows the empirical RMS of errors in estimation of IMU error terms obtained from all the 100 runs. The predicted uncertainties  $(1\sigma)$  are close to empirical RMS, and the accuracy of the estimation is satisfactory in comparison to the amplitude of simulated errors (i.e., realization values) plotted on the same graph.

The empirical RMS of mean error in estimation of VDM parameters obtained from all the 100 runs is plotted in Figure 5.20. Here, the filter is slightly optimist in predicting the estimation



Figure 5.18 – RMS of attitude errors in 100 Monte-Carlo simulation runs for VDM-based navigation, with maximum RMS of  $0.20^{\circ}$ ,  $0.19^{\circ}$ , and  $0.96^{\circ}$  for roll, pitch, and yaw, respectively, during 3 minutes of GNSS outage. The same errors for INS-based navigation were  $0.67^{\circ}$ ,  $0.68^{\circ}$ , and  $7.28^{\circ}$  (adapted from [40]).

Error	With ba	rometer	Without barometer		
LIIO	VDM- based	VDM- INS-based based		INS-based	
Position [m]	38	681	39	2856	
Roll [°]	0.20	0.67	0.20	2.30	
Pitch [°]	0.19	0.68	0.19	2.30	
Yaw [°]	0.96	7.28	1.02	7.64	

Table 5.5 – Maximum RMS of position and attitude estimation error in Monte-Carlo simulations with and without barometric altimeter [40]



Figure 5.19 – RMS of estimation error for IMU error terms from 100 Monte-Carlo runs for VDM-based navigation, together with the amplitude of simulated errors (realization values) (adapted from [40])

accuracy, but the difference does not exceed 20%. The evolution of this error reveals that there is a sharp decrease in the mean error of parameter estimation from the initial 10% error to 6% during the first 40 seconds with GNSS available, which is followed by a slowly decreasing trend afterwards. The reason behind the second regime seems to be the correlation between some parameters within the set. Normally, the groups of parameters are estimated rather than individual parameters in such situations, and those individual errors contribute to increasing the mean error for the whole set. More details on this can be found in Section 5.3.3 when discussing the observability of the filter states. Note that the mean error is calculated in an RMS sense for all the parameters at any given time to cover both the mean value and the standard deviation.

Finally, the RMS of error in estimation of wind speed obtained from all the 100 runs is depicted in Figure 5.21. Again, the filter has been able to provide a close prediction on the estimation uncertainty, and the accuracy is good enough for navigation purposes, especially when considering the fact that no airspeed sensor was considered in the system and the estimation is provided just as a courtesy of the UAV dynamic model. As revealed by the slow growth of predicted error, the wind speed estimation accuracy is expected to decrease as time



Figure 5.20 – RMS of estimation error for VDM parameters from 100 Monte-Carlo runs (adapted from [40])

passes during GNSS outage. However, the rate of this growth is well controlled, reaching only  $0.3 \ m/s$  after 3 minutes of GNSS outage. It is worth noting that all the three components of the wind velocity are estimated within the filter, which are converted to the wind speed for more comprehensive presentation in the plot of Figure 5.21.

#### 5.3.2 Experimental Scenario

The results of a real flight are presented in this section. The experimental setup used for the tests is introduced in Section 4.2.

Figures 5.22 shows the reference trajectory obtained from PPK GNSS solution (not used within the navigation system), as well as the navigation solution from proposed VDM-based navigation. Available standalone GNSS position data that are used in navigation are also depicted in this figure. The duration of GNSS outage is 3 minutes.

As can be seen in Figure 5.22, in contrary to inertial coasting, the VDM-based solution after 3 minutes of autonomous navigation (during GNSS outage) is still following the reference trajectory qualitatively. To get quantitative measures on navigation quality, the 3D position error over the whole trajectory is depicted in Figure 5.23.

The maximum position error after 3 minutes of GNSS outage is 53 m for proposed VDM-based navigation. Under exact same conditions, this error is almost 39 times higher at 2076m for the INS-based navigation. Note that the error in vertical channel is limited by barometer data. For both conventional and proposed navigation systems, the predicted uncertainty is in good agreement with the actual error. The experimental results are also in good agreement with those obtained by simulation. Based on experimenting with longer availability periods for GNSS, one main reason why experimental errors are slightly higher than the ones in



Figure 5.21 - RMS of estimation error for wind speed from 100 Monte-Carlo runs, together with real wind speed (adapted from [40])

simulations seems to be the unresolved part of initialization errors. Unmodeled sensor errors and misalignments among sensors, unmodeled dynamics of the UAV, higher uncertainty in initial values of VDM parameters, and the noise on control input from autopilot that was not considered in simulations, can be other reasons for this difference.

Figure 5.24 depicts the velocity components provided by the VDM-based navigation versus reference values coming from PPK GNSS data. Despite being a bit noisy, VDM-based results capture the trend and follow the reference values closely, even in absence of GNSS position updates for the last 3 minutes of the flight. Note that no GNSS velocity measurement was used in the navigation.

It is worth mentioning that prior to running the VDM-based navigation in this experimental scenario, the VDM parameters were calibrated to some extent. On another piece of the trajectory in the same flight, VDM-based navigation was performed without any GNSS outage to mainly estimate the VDM parameters. Then the estimated parameters at the end of that phase were used as priors in the evaluation phase, results of which were presented in this section. However, a more systematic and reliable approach to VDM parameters calibration with two separate flights for calibration and evaluation will be detailed in Section 5.4.

#### 5.3.3 Observability Discussions for Emulation Scenario

The results and discussions presented in this section are very similar to what was presented in Section 5.1.3 for a different trajectory. There are subtle differences, though, which is why such discussions are presented here, as well.

This section discusses some aspects of observability of filter states. The discussion is based on analysis of covariance matrix throughout the time. Close agreement between the resulted errors



Figure 5.22 – Horizontal view of reference trajectory in the real flight and navigation solutions with GNSS signals available during first 100s only (altitude change  $\simeq 50m$ ) (adapted from [40])

and the predicted confidence levels through the covariance matrix as revealed in Section 5.3.1 and 5.3.2, supports the validity of the stochastic configuration, so that the subsequent discussion can take place.

The covariance and correlation plots presented in this section are related to one sample Monte-Carlo run, since the stochastic aspects discussed here are very similar for all the 100 runs.

Figure 5.25 shows the ratios of uncertainties of states during GNSS presence (top) and GNSS outage (bottom) periods. During GNSS presence period, uncertainties of all the states except a few of VDM parameters were reduced significantly, which indicates observability of them, even during this relatively short duration (100 seconds). Uncertainties of weakly observable VDM parameters were not increased either, which indicates no divergence is expected in their estimation.

During GNSS outage, as expected, uncertainties of position and velocity have grown. This growth was small in vertical channel, thanks to barometric altimeter data. Uncertainties of attitude were reduced by a factor of 0.8 for pitch, and increased by factors of only 1.3 and 5.6 for roll and yaw, respectively, which is considered an excellent performance in attitude determination after 3 minutes of autonomous navigation in relation to the inertial sensor quality. Uncertainties of rotation rates slightly decreased, thanks to the direct measurements from IMU. The uncertainty of none of VDM parameters increased during GNSS outage, which



Figure 5.23 – Position error in the real flight for VDM-based navigation, with maximum value of 53 m during 3 minutes of GNSS outage. The same error for INS-based navigation was 2076 m (adapted from [40]).

means their estimation is preserved even in the absence of GNSS measurements. The same is true for IMU error terms. Finally, uncertainties of wind velocity have increased 2.8 times in horizontal channels, and decreased 0.8 times in vertical channel, the latter thanks to barometric altimeter data. The combined uncertainty for wind speed increased by almost 4 times, which is still quite a useful result for the navigation purposes. Further improvement can also be expected with the use of airspeed sensor observations, as long as the uncertainty of such observations are low with respect to the uncertainty of estimated values by the filter without them.

Figure 5.26 depicts correlation coefficients among all the states within the filter at the end of the trajectory++. As can be seen, the states are well decorrelated in general, which is an indicator of their observability. The VDM parameters block shows the highest level of correlations, especially among different aerodynamic coefficients for any specific moment component (this is distinguishable as very bright small square blocks on the diagonal). This means that rather than individual VDM parameters, groups of VDM parameters are getting estimated within the filter. In this scenario, decorrelating individual VDM parameters demands longer time, higher dynamic maneuvers, and possibly longer periods of GNSS availability. Considering the fact that the VDM parameters are normally constant for a particular platform, their better estimation can be potentially carried out during a calibration scenario. This shall result in lower uncertainties of their initial values for subsequent usages. Anyway, for the purpose of navigation, and not necessarily for VDM parameter estimation, the simulation results reveal



Figure 5.24 – Velocity components for VDM-based navigation with GNSS signals available during first 100s only (adapted from [40])

that the VDM parameters are estimated well enough to enable the proposed VDM-based navigation reaching much better performance compared to inertial coasting.

The other point to notice is how well the VDM parameters are decorrelated from other states of the filter. There remains a considerable level of correlation between wind velocity components and position and velocity of the UAV. As depicted in Figure 5.21, the uncertainty of wind velocity estimation grows steadily, yet slowly, as time passes during GNSS outage. This is in relation to the level of process noise on the wind model. When more gusty conditions are expected, higher values on this process noise can be used to allow capturing more sudden changes in wind velocity, at the price of increasing the uncertainty of navigation solution during GNSS outage, though. Therefore, including airspeed observations might be a good answer to such compromise.

As mentioned in Section 5.1.3, the observability of VDM parameters are limited during GNSS outage. Since they are expected to be more or less constant under normal condition, one may exclude such parameters from estimation during GNSS outages. This means that the last estimated values for them will be kept constant and used within the filter. To be extremely cautious, the same can be done for IMU error terms, although the results presented in this section does not hint at necessity of such action.



Figure 5.25 – Ratios of uncertainties of states during GNSS presence and GNSS outage periods in a sample Monte-Carlo run (GNSS outage begins at t = 100s, and navigation ends at t = 280s) (adapted from [40])

#### 5.3.4 Observability Discussions for Experimental Scenario

As depicted in Figure 5.27 and Figure 5.28, results on covariance matrix in real tests are very similar to those obtained in emulations. This makes most of the arguments made in Section 5.3.3 directly applicable to experimental scenario.

However, one noticeable difference is uncertainty evolution of lateral and vertical velocity components during GNSS presence period being greater than 1, while the same values in simulation results were smaller than 1. Again, this seems to be due to initialization errors that are likely larger and not fully resolved within 100 s of GNSS presence. Inclusion of GNSS velocity observations is expected to improve this. Another difference to notice in Figure 5.28 is that the VDM parameters are decorrelated better among themselves, which may be caused by more dynamic excitation of the platform and stabilization commands by the autopilot in real scenario.



Figure 5.26 – Correlation matrix at the end of navigation in in a sample Monte-Carlo run (adapted from [40])



Figure 5.27 – Ratios of uncertainties of states during GNSS presence and GNSS outage periods in the real flight (GNSS outage begins at t = 100s, and navigation ends at t = 280s) (adapted from [40])



Figure 5.28 – Correlation matrix at the end of navigation in the real flight (adapted from [40])

#### 5.4 Further Experimental Results

In the last section of this chapter, some results from two additional flights are presented, employing a different UAV, TOPO plane 2, as described in Section 4.2.1. In order to ensure about validity of model calibration and applicability of the designed navigation system in real scenarios, two separate flights were performed on two different days. The first flight contained some highly dynamic maneuvers, and the second one contained an area mapping type trajectory with less harsh maneuvers. Data from the first flight were used to calibrate VDM parameters, using internal capabilities of the navigation system for estimation of them. In this phase, cm-level PPK GNSS observations (position and velocity) were used. These VDM parameters were then used in the second flight, to evaluate navigation performance in GNSS outages.

#### 5.4.1 Calibration Flight

The procedure to calibrate the VDM parameters is briefed as following.

- Step 1. The center of mass of the UAV and the lever-arms for sensors was directly measured with cm-level accuracy.
- Step 2. Flight data were processed with INS-based navigation filter employing PPK GNSS solution. The outputs of this step were the navigation states (PVA), IMU error, and possibly refined GNSS lever-arm. One should note that this provides the lever-arm between the IMU and GNSS antenna phase center. For use in VDM-based navigation, this needs to be then superposed with the lever-arm between the IMU and body frame origin. Barometer was pre-calibrated using PPK GNSS altitude data.
- Step 3. Flight data were processed with VDM-based navigation filter employing the INS-based PVA solution (from previous step) as observations. Considering the high number of states and limited observability of some of them, this strategy was found much more effective than using only the unprocessed sensor data in the VDM-based navigation filter. The prioirs on IMU errors, as well as refined GNSS lever-arm were directly imported from INS-based solution. Considering the large initial uncertainty of VDM parameters at this stage, sensor mounting misalignments were not estimated within the filter, but the coarse values known a priori were applied. This yielded a better performance in evaluation phase (next step) compared to the case in which sensor mounting misalignments were estimated in the filter. However, the difference was not drastic and no divergence was observed when trying to estimate sensor mounting misalignments.
- Step 4. One should note that in the real flight there was no reference on the values of VDM parameters and the ultimate goal was improving navigation performance rather than precise determination of VDM parameters. Therefore, position error in VDM-based navigation was used to evaluate indirectly the goodness of VDM parameter calibration. Using the calibrated VDM parameters, flight data were processed with VDM-based navigation once again, this time using only stand alone GNSS solution (position and velocity) and introducing a GNSS outage of 3 minutes at the end. The position error during GNSS outage was then checked, and a few iterations were made to previous step to refine the estimation of VDM parameters so that the quality is acceptable for VDM-based navigation purposes.

#### **Discussion on VDM Parameter Calibration**

One should consider the possibility that the calibrated VDM parameters as just described, may not have converged to true values (which are unknown) necessarily. However, certain sufficiency of this set for VDM-based navigation purposes was established by assessing the navigation error not only for the same trajectory as in calibration flight, but also for a different trajectory from an evaluation flight performed on a different day. Getting closer to true values for at least some of VDM parameters for any specific platform at hand could be possible via methods such as CFD analysis, wind tunnel testing, and performing exhaustive calibration maneuvers exciting only a subset of dynamics at a time. However, the goal here is to minimize the design effort by exploiting internal estimation capabilities of the VDM-based navigation to a level that achieves a substantial improvement in the quality of autonomous navigation in comparison to inertial coasting.

#### Main Results

#### **Calibration Trajectory**

The trajectory of the calibration flight is shown in Figure 5.29. The trajectory has an approximate ground footprint of 350  $m \times 600 m$  and a total change of almost 120 m in altitude. Its duration was over 6.5 minutes, over which the average speed was 16.3 m/s.



Figure 5.29 – Horizontal view (left) and altitude profile (right) for the reference trajectory used in VDM parameters calibration

#### State Estimation

The evolution of IMU errors determined in Step 2 is shown in Figure 5.30.

Refinement of GNSS lever-arm over time in Step 2 is shown in Figure 5.31. At the end of the calibration, the initial values of  $-45 \ cm$ ,  $0 \ cm$ , and  $-4 \ cm$  along the three axes, were refined to  $-32 \ cm$ ,  $0 \ cm$ , and  $9 \ cm$ , respectively. As the phase center of the GNSS antenna does not necessarily lie within its body, the resulting refinement is not unexpected.

The relative change in some sample VDM parameters  $(C_{F_z\alpha}, C_{F_x\beta_2}, \text{and } C_{M_x\tilde{\omega}_x})$  and the mean of absolute values for all of them during calibration in Step 3 is depicted in Figure 5.32. As can be observed in the figure, this change is relatively big with the mean of absolute values reaching over 350%.

Finally, when applying the calibrated VDM parameters within the same calibration flight, the maximum position error in 3 minutes of GNSS outage for VDM-based navigation was 65 m, while this error for INS-based navigation under the same conditions was 1207 m. Figure 5.33 depicts the evolution of position error for VDM-based and INS-based navigation.

Note that for the evaluations (Step 4 in calibration flight and in evaluation flight), the standalone GNSS observations were used instead of PPK observations. The standalone observations come from the same receiver and the same antenna, which means that the lever-arm is also the same.



Figure 5.30 – IMU error estimated via INS-based navigation filter in Step 2  $\,$ 

#### 5.4.2 Evaluation Flight

The evaluation flight data were processed with VDM-based navigation filter in the same way as in Step 4 of Section 5.4.1 (i.e., using the priors on VDM parameters as obtained in the calibration flight).

The trajectory of the evaluation flight is shown in Figure 5.34. The trajectory has an approximate ground footprint of 350  $m \times 750 m$  and a total change of almost 90 m in altitude. Its duration was 6 minutes, over which the average speed was 16.6 m/s.

The maximum position error in 3 minutes of GNSS outage on the calibration flight trajectory using the calibrated VDM parameters was 75 m, with its evolution being depicted in Figure 5.35. This error for INS-based navigation under the same conditions reached 1220 m.

Figure 5.36 depicts the velocity components provided by the VDM-based navigation versus reference values coming from PPK GNSS. Despite being a bit noisy, VDM-based results capture the trend and follow the reference values closely, even in absence of GNSS position and velocity updates for the last 3 minutes of the flight.



Figure 5.31 – Refinement of GNSS lever-arm via INS-based navigation filter in Step 2  $\,$ 



Figure 5.32 – Relative change in VDM parameters during calibration via VDM-based navigation filter in Step 3: mean of absolute values for all parameters and 3 sample ones



Figure 5.33 – Position error in evaluation of VDM parameters calibration in Step 4 with 3 minutes of GNSS outage at the end

#### Note on Airspeed Sensor Data

Due to high noise level of the airspeed sensor data as mentioned in Section 4.2.2 ( $\sigma = 1 \text{ m/s}$ ), inclusion of such data did not provide any meaningful improvement to navigation results. Therefore, airspeed measurements were not used in navigation. However, such measurements are expected to be helpful in case of higher wind velocities, as well as longer GNSS outages. Further investigation on effects of using airspeed sensor data is the subject of one of the suggestions for future research presented in Section 7.2.



Figure 5.34 – Horizontal view (left) and altitude profile (right) for the reference trajectory used in evaluation flight with VDM-based navigation solution for 3 minutes of GNSS outage at the end



Figure 5.35 – Position error in evaluation flight with 3 minutes of GNSS outage at the end



Figure 5.36 – Velocity components for VDM-based navigation in evaluation flight with 3 minutes of GNSS outage at the end

## Chapter 6

# **Further Analyses**

#### Overview

This chapter presents some further results and analyses on VDM-based navigation. First, a global sensitivity analysis is presented to rank the VDM parameters based on the effect of their uncertainty on navigation performance.

Afterwards, a scenario is studied, in which no IMU data is used in navigation. The main goal is to assess the ability of VDM-based navigation in attitude determination without using IMU data, which might be helpful in case of IMU failure or malfunction for the purpose of UAV control.

Finally, a brief discussion is presented on computational cost of VDM-based navigation.

#### 6.1 Global Sensitivity Analysis

VDM contains a number of parameters related to the specific platform at hand. Some parameters such as mass, or those directly related to geometrical measures, can be determined a priori with rather high confidence. However, the aerodynamic coefficients are in general more difficult to measure or calculate, and are therefore associated with higher levels of uncertainty. The methods to estimate these coefficients vary in time and cost required to setup, such as estimation within the navigation system through a calibration scenario, CFD analyses, and wind tunnel tests. Based on a [rough] prior knowledge of parameter values, sensitivity analysis provides a quantitative base to assess the effect of errors in each parameter on VDM output and order the parameters in terms of their influence. This might help in taking decisions on refinement of parameters and the strategy for calibration.

As explained in the following section, local sensitivity analyses fail to consider the wide range of inputs/parameters in the models. Considering the need to cover a large portion of flight envelope in VDM-navigation, a global sensitivity analysis is performed, theory of which is briefly described in the following.

#### 6.1.1 Theory

Sensitivity analysis is a term of many definitions and interpretations, and even more methods to perform. A possible definition for "sensitivity analysis" is given in [65] as "the study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input". A related practice, which is sometimes mistakenly used instead of sensitivity analysis, is "uncertainty analysis", meaning quantification of uncertainty in model output. Ideally, these two analyses should be performed in a chain with the uncertainty analysis preceding [66].

Two main categories for sensitivity analyses are local and global analyses. Local sensitivity analysis is the one with the longer history, as a measure of the effect of a given parameter (or input) of the model on a given output. To assess this measure, a single point in parameter space is considered, and the output variation due to variation of a single parameter is estimated. In case there exists a closed form mathematical model, this task can be accomplished via partial derivatives. Many reports on the sensitivity analyses found in physical science journals are local analyses, which is not always the appropriate approach [65]. For a system of the form  $Y = f(X_1, X_2, \ldots, X_n)$ , sigma normalized derivatives can be considered as local sensitivity indices and are calculated as [66]

$$S_{X_i}^{\sigma} = \frac{\sigma_{X_i} \partial Y}{\sigma_Y \partial X_i}.$$
(6.1.1)

They provide a measure of how changes in  $X_i$  are magnified in Y, via  $\frac{\partial Y}{\partial X_i}$ , which is further normalized by  $\sigma_{X_i}$  to give more importance to parameters with a wider range of variation, and by  $\sigma_Y$  to make interpretation of results easier as  $\sum_{i=1}^n \left(S_{X_i}^{\sigma}\right)^2 = 1$ .

Global analyses, on the other hand, can take into account the effect of all parameters. In such methods, based on a given distribution of all parameters, a measure is obtained for each parameter that reflects how much the output is affected by the variation of that parameter. This gives a measure that is averaged over the entire parameter space and takes into account the combined effect of any group of parameters rather than the single ones. For a system of the form  $Y = f(X_1, X_2, \ldots, X_n)$ , conditional variances can be considered as global sensitivity indices. The first order indices are calculated as

$$S_{i} = \frac{V_{X_{i}}\left(E_{X_{\sim i}}\left(Y|X_{i}\right)\right)}{V\left(Y\right)}.$$
(6.1.2)

The term  $E_{X_{\sim i}}(Y|X_i)$  is the average of Y value when  $X_i$  is fixed at some value  $(Y|X_i)$  and all other parameters  $(X_{\sim i})$  vary.  $V_{X_i}(E_{X_{\sim i}}(Y|X_i))$  is the variance of this conditional average, if the fixed value of  $X_i$  is changed over all its possible values. Normalizing this variance by V(Y)ensures  $0 \leq S_i \leq 1$  for easier interpretations. A high value for  $S_i$  reveals a high importance of  $X_i$ . However, a small  $S_i$  does not necessarily represent an unimportant  $X_i$ , because  $S_i$  contains information only on direct (first order) effects of  $X_i$  on Y, and not on how variation of  $X_i$  in combination with variations of other parameters affects the variance of Y. If the variance of Y is conditioned on more than one parameter, higher order sensitivity indices emerge naturally. For example, for  $i \neq j$  and in general case,

$$\frac{V_{X_i,X_j}\left(E_{X_{\sim i,j}}\left(Y|X_i,X_j\right)\right)}{V\left(Y\right)} \ge \frac{V_{X_i}\left(E_{X_{\sim i}}\left(Y|X_i\right)\right)}{V\left(Y\right)} + \frac{V_{X_j}\left(E_{X_{\sim j}}\left(Y|X_j\right)\right)}{V\left(Y\right)},$$

which hints to define second order sensitivity index  $S_{ij}$  with  $i \neq j$  as

$$S_{ij} = \frac{V_{X_i,X_j} \left( E_{X_{\sim i,j}} \left( Y | X_i, X_j \right) \right)}{V(Y)} - \frac{V_{X_i} \left( E_{X_{\sim i}} \left( Y | X_i \right) \right)}{V(Y)} - \frac{V_{X_j} \left( E_{X_{\sim j}} \left( Y | X_j \right) \right)}{V(Y)}$$
$$= \frac{V_{X_i,X_j} \left( E_{X_{\sim i,j}} \left( Y | X_i, X_j \right) \right)}{V(Y)} - S_i - S_j.$$
(6.1.3)

Total effects index for  $X_i$  is then defined as sum of all possible sensitivity indices of any order that contains  $X_i$ .

$$S_{Ti} = S_i + \sum_{j=1, j \neq i}^n S_{ij} + \sum_{j=1, j \neq i}^n \sum_{k=1, k \neq i, k \neq j}^n S_{ijk} + \dots + S_{123\dots n}$$
(6.1.4)

Since the sum of all possible sensitivity indices of any order should sum up to 1, total effects index  $S_{Ti}$  can be calculated from

$$S_{Ti} = 1 - \frac{V_{X_{\sim i}} \left( E_{X_i} \left( Y | X_{\sim i} \right) \right)}{V \left( Y \right)},\tag{6.1.5}$$

in which  $\frac{V_{X_{\sim i}}(E_{X_i}(Y|X_{\sim i}))}{V(Y)}$  is equivalent to the sum of all sensitivity indices that do not contain  $X_i$ .

Although for a system with n parameters, there are  $2^n - 1$  sensitivity indices, "a good, synthetic, though nonexhaustive characterization of the sensitivity pattern for a model with n factors is given by the total set of first-order terms plus the total effects" [66]. This translates into 2n indices only.

Numerical evaluation of sensitivity indices is performed based on a Monte-Carlo method proposed in Chapter 4 of [66].

#### 6.1.2 Results

Any effect from VDM parameters on navigation quality comes from calculated values of aerodynamic forces and moments utilizing those parameters. Therefore, sensitivity analysis is performed on aerodynamic forces and moments rather than on navigation solution, which is directly influenced by the former.

One has to notice that the global sensitivity analysis performed here gives an overall image of sensitivity over a wide range of values for states and flight conditions. In case of any specific maneuver, the order of importance for parameters may change compared to what global sensitivity analysis suggests. Therefore, if the navigation sensitivity in any specific maneuver is intended to be studied, the described sensitivity analysis should be performed by defining a navigation quality measure as the output function and adjusting the distribution of states according to the specific maneuver. However, due to very limited application of scenarios of such kind and considering the fact that the navigation system is normally expected to work over a wide range of maneuvers, the global sensitivity analysis is preferred in this research over local analyses.

Distribution of states for the global sensitivity analysis was considered as following, where  $\delta_{e(level)}$  denotes the value of  $\delta_e$  to perform a straight leveled flight at 15 m/s, and  $\mathbf{X}_{vp(nominal)}$  is the vector of nominal (unperturbed) VDM parameters presented in Table 4.6.

Velocity:	$v_x^b \sim \mathcal{N}\left(15, 5^2\right),$	$v_y^b \sim \mathcal{N}\left(0, 3^2\right),$	$v_z^b \sim \mathcal{N}\left(0, 3^2\right)$
Attitude:	$r \sim \mathcal{N}\left(0, (\pi/6)^2\right),$	$p \sim \mathcal{N}\left(0, (\pi/6)^2\right)$	
Rotation rate:	$\omega_x \sim \mathcal{N}\left(0, (\pi/3)^2\right),$	$\omega_y \sim \mathcal{N}\left(0, (\pi/3)^2\right),$	$\omega_z \sim \mathcal{N}\left(0, (\pi/3)^2\right)$
Wind velocity:	$w_N \sim \mathcal{N}\left(0, 4^2\right),$	$w_E \sim \mathcal{N}\left(0, 4^2\right),$	$w_D \sim \mathcal{N}\left(0, 1^2\right)$
Actuator states:	$\delta_a \sim \mathcal{N}\left(0, 0.1^2\right),$	$\delta_e \sim \mathcal{N}\left(\delta_{e(level)}, 0.1^2\right),$	$\delta_r \sim \mathcal{N}\left(0, 0.1^2\right),$
			$\omega_p \sim \mathcal{N} \left( 100, 10^2 \right)$

VDM parameters:

$$\mathbf{X}_{vp} \sim \mathcal{N}\left(\mathbf{X}_{vp(nominal)}, (0.2|\mathbf{X}_{vp(nominal)}|)^2\right)$$

#### Aerodynamic Forces

Values of sensitivity indices for VDM parameters affecting the magnitude of deviation in aerodynamic forces are presented in Table 6.1.

Table 6.1 – First order (S) and total effects  $(S_T)$  indices in sensitivity analysis of aerodynamic forces (limited to 3 decimal digits)

Index	$C_{F_z\alpha}$	$C_{F_T1}$	$C_{F_T2}$	$C_{F_T3}$	$C_{F_x1}$	$C_{F_z1}$	$C_{F_y1}$	$C_{F_x\beta 2}$	$C_{F_x \alpha}$	$C_{F_x \alpha 2}$
S	0.055	0.023	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$S_T$	0.101	0.052	0.006	0.005	0.002	0.002	0.002	0.002	0.002	0.002

As can be seen,  $C_{F_z\alpha}$  was identified as the most influential parameter, while  $C_{F_x\alpha 2}$  was among the least influential ones. One should be aware that due to very close values of sensitivity indices for least influential parameters while considering the uncertainties in numerical evaluation of indices, the sorting gets less confident towards the end of the ordered list of parameters. Also, relatively low values of sensitivity indices for VDM parameters shall not be interpreted as low importance of them in general. This is because a large share of the variance in the output is due to variations in factors other than VDM parameters, such as navigation states and wind velocity. As a sample demonstration of how these parameters can affect the navigation performance, the test scenario presented in Section 5.3 was considered again. VDM-based navigation was performed on real and simulated data using nominal set of VDM parameters (as used in Section 5.3) and four additional sets with the most and the least influential parameters ( $C_{F_z\alpha}$  and  $C_{F_x\alpha 2}$ , respectively) perturbed by  $\pm 20\%$ . The reference trajectory is depicted in Figure 5.16 for simulation and in Figure 5.22 for experimental data. Stochastic models to simulate sensor error come from Section 4.2.2, and filter setup, such as initialization and observation error statistics, is in accordance with the details presented in Section 4.3. Table 6.2 summarizes maximum positioning error during 3 minutes of GNSS outage.

Table 6.2 – Maximum position error during 3 minutes of GNSS outage with perturbations on most and least influential VDM parameters in aerodynamic forces

VDM parameters	Nominal	$C_{F_z\alpha} + 20\%$	$C_{F_z\alpha}$ -20%	$C_{F_x\alpha 2}$ +20%	$C_{F_x \alpha 2}$ -20%	
Position error [m] in	38.3	38.6	38.9	38.4	38.1	
simulation results	00.0	00.0	00.2	00.4	50.1	
Position error [m] in	53 /	250.0	50.4	60.2	78.8	
experimental results	00.4	209.9	09.4	00.2	10.0	

There are a number of points to notice in Table 6.2. In simulation results, an error of 20% in neither of the parameters had a considerable effect on positioning error, which can be explained by most of the parameters being estimated very well within the filter. In experimental results, though, one can see that the change in positioning error is in accordance with what the sensitivity analysis suggested, where  $C_{F_z\alpha}$  was expected to be more influential than  $C_{F_x\alpha 2}$ . Also, increasing parameter errors in a specific direction (+ or -) can sometimes work against other error sources such as sensor errors and thus have a very different effect than the same error in the opposite direction. This is, however, highly case-dependent and therefore difficult to predict. For this reason, perturbations were made in both directions in this demonstration.

#### Aerodynamic Moments

Values of sensitivity indices for VDM parameters affecting the magnitude of deviation in aerodynamic moments are presented in Table 6.3.

Table 6.3 – First order (S) and total effects  $(S_T)$  indices in sensitivity analysis of aerodynamic moments (limited to 3 decimal digits)

Index	$C_{M_ye}$	$C_{M_xa}$	$C_{M_z\beta}$	$C_{M_z r}$	$C_{M_z \tilde{\omega}_z}$	$C_{M_x \tilde{\omega}_x}$	$C_{M_y \tilde{\omega}_y}$	$C_{M_y1}$	$C_{M_y \alpha}$	$C_{M_x\beta}$	$C_{M_x \tilde{\omega}_z}$
S	0.012	0.010	0.003	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$S_T$	0.031	0.020	0.014	0.013	0.006	0.004	0.004	0.003	0.000	0.000	0.000

Here,  $C_{M_ye}$  was identified as the most influential parameter, while  $C_{M_x\tilde{\omega}_z}$  was among the least influential ones. The sample demonstration for aerodynamic forces was repeated for aerodynamic moments, as well, results of which are summarized in Table 6.2.

#### **Further Analyses**

Table 6.4 – Maximum position error during 3 minutes of GNSS outage with perturbations on most and least influential VDM parameters in aerodynamic moments

VDM parameters	Nominal	$C_{M_ye}$ +20%	$C_{M_ye}$ -20%	$C_{M_x \tilde{\omega}_z} + 20\%$	$C_{M_x \tilde{\omega}_z}$ -20%	
Position error [m] in	38.3	38.8	38.0	38.7	38.0	
simulation results	00.0	00.0	00.0	00.1	00.0	
Position error [m] in	53 4	67.5	55 9	63.9	62.3	
experimental results	00.1	01.0			02.0	

The observations related to simulation results are very similar to what they were in case of aerodynamic forces. The experimental results, though, do not reflect exactly what the sensitivity analysis suggested on relative importance of parameters. This may highlight two points. First, isolated performance evaluations with specific trajectories may be misleading on how different parameters affect system performance over a larger portion of flight envelope. Therefore, generalization of sample observations shall be avoided. Second, sorting important parameters according to a formal sensitivity analysis in a global sense does not reflect in all the sample cases, and should be interpreted in an average sense over entire distribution of parameters and inputs considered in the analysis.

#### 6.2 VDM-based Navigation without IMU

Although not as probable as GNSS outage, IMU failure or malfunction is a potential problem in navigation/control system of UAVs, a reason for which there is a recent appearance of redundant IMUs in autopilots even for MAVs [4]. Most navigation systems for UAVs are relying mainly on inertial observations for attitude determination, and with faulty or unavailable IMU data, attitude determination becomes impossible or of very poor quality, even in presence of GNSS position and velocity observations. Considering the role of attitude determination in flight control, this can have a severe impact on flight performance and stability of the platform.

This section presents the results of an investigation performed to assess the feasibility of using VDM-based navigation as an emergency solution for attitude determination with no IMU data. The simulation and experimental flight data described in Section 5.4.2 –reference trajectory of which is shown in Figure 5.34– were processed again. VDM-based navigation was performed with aiding from standalone GNSS position and velocity data, results of which were compared with those of INS-based navigation with the same aiding GNSS data. The respective results are described in the Section 6.2.1 for emulation scenario and in Section 4.2.2, and filter setup, such as initialization and observation error statistics, is in accordance with the details presented in Section 4.3.

#### 6.2.1 Emulation Scenario

A Monte-Carlo simulation with 100 runs was performed on the emulated trajectory resulted from the experimental flight, reference trajectory of which shown in Figure 5.34.



Figure 6.1 – Roll errors for all the 100 runs of Monte-Carlo simulation with empirical RMS

Figure 6.1 shows the comparison of roll error for VDM/GNSS and INS/GNSS navigation setups. After the first minute of navigation when the effect of initialization errors was considerably mitigated, the RMS of roll error for 100 runs did not exceed 0.30° for INS/GNSS. This error was 0.69° for VDM/GNSS, which is still acceptable for guidance and control of the UAV.

Pitch errors for VDM/GNSS and INS/GNSS navigation setups are compared in Figure 6.2. After leaving out the first minute for mitigation of initialization errors, the maximum value of RMS of pitch error for 100 runs was 0.20° for INS/GNSS. This error went up to 0.97° for VDM/GNSS, which should be still acceptable for the purpose of guidance and control of the UAV.

Finally, Figure 6.3 shows the comparison of yaw error for VDM/GNSS and INS/GNSS navigation setups. Discarding the first minute of navigation for mitigation of initialization errors, the RMS of yaw error for 100 runs reached a maximum of  $1.05^{\circ}$  for INS/GNSS and  $1.61^{\circ}$  for VDM/GNSS, which is again an acceptable result for UAV guidance and control.

The summary of attitude errors is presented in Table 6.5, providing also the maximum RMS of errors during the first minute.

Table 6.5 – Maximum RMS of attitude errors for VDM/GNSS and INS/GNSS navigation in emulation scenario, during and after the first minute for mitigation of initialization errors

Navigation	Roll error[°]		Pitch e	error[°]	Yaw error[°]		
type	$t \le 60s$	t > 60s	$t \le 60s$	t > 60s	$t \le 60s$	t > 60s	
VDM/GNSS	3.70	0.69	6.87	0.97	5.13	1.61	
INS/GNSS	2.90	0.30	2.94	0.20	5.35	1.05	



Figure 6.2 – Pitch errors for all the 100 runs of Monte-Carlo simulation with empirical RMS



Figure 6.3 – Yaw errors for all the 100 runs of Monte-Carlo simulation with empirical RMS

#### 6.2.2 Experimental Scenario

Attitude errors for experimental scenario are depicted in Figures 6.4 to 6.6. The reference values were obtained from fusion of the same IMU data with PPK GNSS data in an INS/GNSS filter. Due to use of the same IMU data, this is not an entirely independent reference, but the best available. Note that the GNSS data used in the test came from stand-alone mode.



Figure 6.4 – Roll error for experimental data

As evident from the figures and summarized in Table 6.6, attitude errors for VDM/GNSS navigation are typically an order of magnitude bigger than the those for INS/GNSS, and probably too large for the purpose of UAV control. This is considerably different from what was observed in simulation scenario. The reason behind such large errors in VDM/GNSS navigation seems to be unresolved errors in VDM parameters. The calibrated set of VDM parameters was sufficiently good for autonomous positioning during GNSS outage with positioning accuracy improvement of one to two orders of magnitude in comparison to inertial coasting (as seen in Section 5.4.2). However, the remaining errors –supposably in moment related parameters– are too large to perform attitude determination without IMU data.

Table 6.6 - VDM/GNSS and INS/GNSS attitude errors in experimental scenario

Navigation	Roll error[°]		Pitch e	error[°]	Yaw error[°]		
type	Maximum	Mean	Maximum	Mean	Maximum	Mean	
VDM/GNSS	40.6	5.3	17.9	4.8	18.0	3.8	
INS/GNSS	2.3	0.3	2.8	1.1	2.7	0.7	



Figure 6.5 – Pitch error for experimental data



Figure 6.6 – Yaw error for experimental data

#### 6.3 Computational Cost

An increase in computational cost due to higher number of states and more complicated equations may be a concern with VDM-based navigation. This subject is briefly discussed here.

Conventional INS-based navigation is widely used and even older embedded systems have enough processing power to run it. Therefore, a relative comparison is made here between proposed VDM-based and conventional INS-based navigation algorithms. A trajectory of 5 minutes with GNSS observations at 1 Hz was emulated and post-processed 100 times at IMU frequency of 100 Hz on a personal computer. Mean and standard deviation of required computation time for both algorithms are presented in Table 6.7. Normalization is done with respect to 16-states INS-based navigation.

Table 6.7 – Computation time comparison (VDM-based 1 and 2: without and with VDM parameters estimation, respectively)

Navigation	Numbe	r of states	Normalized			
algorithm	Tumbe	I OI States	computation time			
aigoritiini	Absolute	Normalized	Mean	Standard deviation		
INS-based	16	1	1	0.02		
VDM-based 1	23	1.44	1.7	0.01		
VDM-based 2	49	3.06	2.7	0.07		

As can be seen in Table 6.7, VDM-based navigation (with VDM parameters estimation) has a computation time of only 2.7 times higher than INS-based navigation, which can be further reduced to 1.7 times by disabling VDM parameters estimation (in case these parameters are pre-calibrated, for example). Although these results were obtained using MATLAB codes running on a  $PC^1$ , it seems safe to assume that with available computation resources on most small UAVs today, running the proposed VDM-based navigation should be possible. On the other hand, such difference in computational cost gets less importance as time passes and embedded computers get more powerful.

<sup>&</sup>lt;sup>1</sup>personal computer

# Part IV

# **Conclusion Remarks**

### Chapter 7

# **Conclusions and Suggestions**

This research aimed at development of a new navigation algorithm for UAVs to considerably improve navigation accuracy during GNSS outages, as a prerequisite for safer operations and expanding operational scenarios, with three main characteristics:

- 1. Preserving navigation autonomy in a wide range of conditions. This means that environment dependent sensors such as cameras are not considered to avoid dependency on lighting conditions, signal propagation, terrain texture, etc.
- 2. Avoiding extra cost and weight due to additional sensors, which is a critical issue for small civilian UAVs.
- 3. Avoiding complex and time consuming design and setup procedures compared to normal operations of small civilian UAVs.

With these requirements in mind, a VDM-based navigation system was proposed that benefits from information on motion dynamics of UAVs in navigation. After a brief presentation of the preliminary concepts, the general theory and architecture of the proposed navigation system were detailed. Implementation was then presented for a specific fixed-wing UAV and the experimental setup was described, which was followed by simulation and experimental analyses. Further analyses were performed on sensitivity of navigation accuracy to errors in VDM parameters, and on navigation in absence of IMU data using VDM/GNSS setup as a backup solution for IMU failure and malfunction. In the following, conclusion remarks and suggestions for future research are presented.

#### 7.1 Conclusions Remarks

1. Hundreds of simulations performed in Monte-Carlo scheme revealed improvements of one to two orders of magnitude in position determination through autonomous navigation during GNSS outages of 5 minutes, when compared to conventional INS-based navigation under the same conditions. Consistent prediction of error levels for both navigation and auxiliary states by the navigation filter revealed proper stochastic setup of the filter and provided a basis for observability discussions based on covariance matrix analysis.

#### **Conclusions and Suggestions**

- 2. Monte-Carlo simulations using real 3D wind velocity data revealed an almost linear dependency of navigation error to wind speed, when only internal capabilities of the navigation filter were exploited to estimate wind velocity (i.e., in absence of airspeed sensor). Navigation accuracy was insensitive to improper settings of process noise for wind velocity within the filter.
- 3. In Monte-Carlo simulations, IMU errors and wind velocity were estimated with errors typically bellow 10%. The average error on VDM parameters estimation, on the other hand, stayed almost unchanged after an initial sharp decrease. Covariance matrix analysis revealed the reason being some degree of unobservability for VDM parameters that prevented individual parameters to be decorrelated with the available observations and the performed maneuver. Carefully designed calibration maneuvers with higher dynamics and longer periods are expected to result in better decorrelation of VDM parameters.
- 4. Simulations were validated by exploiting experimental flight data collected by a custom made fixed-wing MAV. First, Monte-Carlo simulation on emulated flight data based on the trajectory of the real flight showed again an improvement of one to two orders of magnitude in navigation accuracy during 3 minutes of GNSS outage. Second, similar improvement and comparable absolute errors in autonomous navigation were observed in reprocessing real flight data using cm-level GNSS positioning as the reference.
- 5. Further experimental tests were conducted with another realization of the same fixedwing UAV platform. VDM parameters were estimated in a "calibration flight" and then used for navigation in an "evaluation flight" performed on another day with a different trajectory. Maximum position error during 3 minutes of navigation with GNSS outage was similar to the previous emulation and experiment, and also differed only by 15% between the calibration and evaluation flights, showing the applicability of the performed VDM calibration as far as autonomous navigation is concerned.
- 6. Computation requirements were evaluated for VDM-based navigation in a post processing scheme on an emulated trajectory, for a setup with 23 states (fixed VDM parameters) and 49 states (re-estimated VDM parameters). The measured computation times were respectively 1.7 and 2.7 times longer than those of conventional INS-based navigation with 16 states. Repeating the simulations 100 times resulted in respective standard deviations of 0.01 and 0.07 for the reported numbers. The excessive computational cost gets less importance as time passes and processing capabilities grow.
- 7. Simplified calculations estimated the impacts of ignoring Earth rotation and curvature on autonomous navigation. It was shown that even in flights limited to speed of 20 m/s, range of 1 km, and duration of 3 minutes, a flat non-rotating Earth assumption would result in positioning errors of several tens of meters for inertial coasting, which is at the same level of errors produced by VDM-based navigation during GNSS outages.
- 8. Results of individual local sensitivity analyses differed considerably from those of the global sensitivity analysis that considered a wider range of possible flight conditions and ranked the VDM parameters based on how their errors affect VDM output and therefore VDM-based navigation accuracy. This emphasizes the limitations of making general conclusions based on local sensitivity analyses.
9. Simulation results showed that in absence of IMU data, due to sensor failure for example, VDM-based navigation was able to provide attitude estimation with an accuracy comparable to that of INS-based navigation with healthy data from the considered MEMES IMU. The attitude error for VDM/GNSS navigation was at most 5 times larger than that of INS/GNSS, keeping error of under 1 degree for roll and pitch and under 2 degrees for yaw. This is sufficient for control of the UAV in case of IMU failure or malfunction, provided that proper fault detection and exclusion is implemented. In the experimental scenario, however, such performance for VDM/GNSS navigation was not achieved.

Some unanswered questions or unexpected observations encountered in this research, such as the one just mentioned on VDM/GNSS attitude estimation, are mentioned in the next section where suggestions for future research are presented.

## 7.2 Suggestions for Future Research

Compared to INS-based navigation with more than half a century of history for INS and more than a quarter of century for INS/GNSS integration, VDM-based navigation is still in its infancy. Therefore, there are various directions for further research and development on the subject. This section briefly mentions a few, more in line with the observations made in this research.

- 1. As one of the main objectives of this research was keeping the effort required for implementation of the proposed VDM-based navigation for any specific UAV, VDM parameters calibration was performed using only internal estimation capabilities of the navigation filter. This approach proved to be effective as far as autonomous positioning accuracy in GNSS outage conditions is concerned. However, further investigations related to navigation without IMU data revealed a vast difference between experimental and simulation results, hinting at insufficiency of the calibration procedure performed for "IMU-free" attitude estimation. Furthermore, no direct evaluation of calibrated VDM parameters was performed. Since VDM parameters are of great importance on navigation performance, establishing and implementing more rigorous procedures for VDM parameters calibration and evaluation can be a major contribution to the field.
- 2. In this research ,the integration with GNSS observations was performed in a loose coupling scheme, where range and Doppler frequency measurements are preprocessed to provide position and velocity data. However, direct use of range and Doppler frequency measurements, which is possible in a tight coupling scheme, has numerous benefits for navigation quality and is well worthy of investigation. Benefiting from partial GNSS data in marginal conditions when there are less than four usable ranges to get a position fix in loose coupling scheme is one of the benefits. Direct use of raw measurements also provides better stochastic modeling and hence higher accuracy.
- 3. Although models for including airspeed sensor in navigation were developed and implemented, the practical benefits of using this instrument for VDM-based navigation are yet to be confirmed. The main reason is the relatively high noise level ( $\sigma \approx 1 \ m/s$ )

of the instrument that prevented the beneficial usage within the navigation system in current setup. Therefore, there is still the need for further investigation on how a better sensor could improve wind velocity estimation in VDM-based navigation, or whether the same sensor is still useful above a certain level of wind speed or in longer GNSS outages.

- 4. There are some auxiliary parameters, further refinement of which may let VDM-based navigation to further improve the autonomous navigation performance. Some sensor mounting misalignment parameters are examples of those. However, high correlation of these parameters within themselves and with other parameters and states such as sensor errors and VDM parameters makes their estimation challenging, on which further research can be focused.
- 5. In this research, actuator dynamics was considered for the propeller and ignored for the control surfaces, due to fast response of the employed servos. However, further research may be conducted on modeling actuator dynamics and to evaluate in detail the effect of unmodeled or mismodeled dynamics.
- 6. An alternative to modeling actuator dynamics and feeding such models with control commands is direct measurement of actuator states (such as propeller speed and aileron deflection) using proper sensors. Further research can investigate the feasibility and added value of such approach.
- 7. As VDM-based navigation treats IMU as a sensor providing observations, multiple IMUs can be independently used within the navigation system. While redundant IMUs are proven to be able to deliver better performance in INS-based navigation systems, the extent of their effectiveness in VDM-based navigation can be further investigated.
- 8. Online implementation on a small platform is a big step ahead for VDM-based navigation and requires overcoming many scientific and engineering challenges worth of further research, including but not limited to possible reduction in number of states and model complexity guided by analysis of remaining correlations among some states, online access to control commands or direct measurement of actuator states, as well as design and implementation of a switching algorithm to automatically perform the transition between INS-based and VDM-based navigations when the UAV takes off or lands.
- 9. Finally, implementation of VDM-based navigation on other types of platforms such as quadcopters is a valuable research direction. Early simulation results, not provided in this thesis, were promising on the improvements that VDM-based navigation can bring to quadcopters for positioning during GNSS outages.

## Appendix A

# Manual Model Linearization for a Simplified Case

In this appendix, the manual linearization of the process model for a simplified case. Compared to the final model used in this research and presented in Chapters 3 and 4, the following simplifications/changes are recognized, despite of which the linearization is tedious and rather complex.

- The state vector here contains only the navigation states  $\mathbf{X}_n$ .
- A flat non rotating Earth model is assumed with a constant gravity model.
- Attitude is represented using Euler angles rather than quaternions.
- Velocity is expressed in body frame rather than the local level frame.

The state vector is presented as

$$\mathbf{X} = \begin{bmatrix} x_N, x_E, x_D, v_x^b, v_y^b, v_z^b, r, p, y, \omega_x, \omega_y, \omega_z \end{bmatrix}^T.$$
(A.0.1)

The VDM can now be formulated as following.

$$\begin{bmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_D \end{bmatrix} = \mathbf{C}_b^l \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix}, \qquad \mathbf{C}_l^b = \mathbf{C}_1(r) \, \mathbf{C}_2(p) \, \mathbf{C}_3(y)$$
(A.0.2)

$$\begin{bmatrix} \dot{v}_x^b \\ \dot{v}_y^b \\ \dot{v}_z^b \end{bmatrix} = \begin{bmatrix} -g\sin p \\ g\sin r\cos p \\ g\cos r\cos p \end{bmatrix} + \frac{1}{m} \begin{bmatrix} \begin{pmatrix} F_T \\ 0 \\ 0 \end{pmatrix} + \mathbf{C}_w^b \begin{pmatrix} F_x^w \\ F_y^w \\ F_z^w \end{pmatrix} \end{bmatrix} - \begin{bmatrix} \omega_y v_z^b - \omega_z v_y^b \\ \omega_z v_x^b - \omega_x v_z^b \\ \omega_x v_y^b - \omega_y v_x^b \end{bmatrix}$$
(A.0.3)

$$\begin{bmatrix} \dot{r} \\ \dot{p} \\ \dot{y} \end{bmatrix} = \mathbf{C}_{\omega} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \qquad \mathbf{C}_{\omega} = \begin{bmatrix} 1 & \tan p \sin r & \tan p \cos r \\ 0 & \cos r & -\sin r \\ 0 & \sin r / \cos p & \cos r / \cos p \end{bmatrix}$$
(A.0.4)

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = (\mathbf{I}^b)^{-1} \left( \begin{bmatrix} M_x^b \\ M_y^b \\ M_z^b \end{bmatrix} - \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \mathbf{I}^b \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \right)$$
(A.0.5)

Calculating the partial derivative of  $\dot{\mathbf{X}}$  with respect to  $\mathbf{X}$  produces the F matrix.

$$\mathbf{F} = \frac{\partial \dot{\mathbf{X}}}{\partial \mathbf{X}} \tag{A.0.6}$$

### Rows 1 to 3 of F matrix

The first three rows are easily calculated according to Equation (A.0.2).

$$\mathbf{F}([1\ 2\ 3], [4\ 5\ 6]) = \frac{\partial}{\partial [v_x^b\ v_y^b\ v_z^b]^T} \begin{bmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_D \end{bmatrix} = \mathbf{C}_b^l \tag{A.0.7}$$

$$\mathbf{F}([1\ 2\ 3],7) = \frac{\partial}{\partial r} \begin{bmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_D \end{bmatrix} = \frac{\partial \mathbf{C}_b^l}{\partial r} \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} = \mathbf{C}_3^T(y) \mathbf{C}_2^T(p) \left(\frac{\partial}{\partial r} \mathbf{C}_1^T(r)\right) \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix}$$
(A.0.8)

$$\mathbf{F}([1\ 2\ 3], 8) = \frac{\partial}{\partial p} \begin{bmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_D \end{bmatrix} = \frac{\partial \mathbf{C}_b^l}{\partial p} \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} = \mathbf{C}_3^T(y) \left(\frac{\partial}{\partial p} \mathbf{C}_2^T(p)\right) \mathbf{C}_1^T(r) \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix}$$
(A.0.9)

$$\mathbf{F}([1\ 2\ 3],9) = \frac{\partial}{\partial y} \begin{bmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_D \end{bmatrix} = \frac{\partial \mathbf{C}_b^l}{\partial y} \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} = \left(\frac{\partial}{\partial y} \mathbf{C}_3^T(y)\right) \mathbf{C}_2^T(p) \mathbf{C}_1^T(r) \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix}$$
(A.0.10)

$$\mathbf{F}([1\ 2\ 3], [1\ 2\ 3\ 10\ 11\ 12]) = \frac{\partial}{\partial [x_N\ x_E\ x_D\ \omega_x\ \omega_y\ \omega_z]^T} \begin{bmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_D \end{bmatrix} = [0]_{3\times 6}$$
(A.0.11)

### Rows 4 to 6 of F matrix

To calculate rows 4, 5, and 6 of F, Equation (A.0.3) is rewritten as

$$\begin{bmatrix} \dot{v}_x^b \\ \dot{v}_y^b \\ \dot{v}_z^b \end{bmatrix} = \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3,$$
 (A.0.12)

where

$$\mathbf{P}_{1} = \begin{bmatrix} -g\sin p\\ g\sin r\cos p\\ g\cos r\cos p \end{bmatrix} - \begin{bmatrix} \omega_{y}v_{z}^{b} - \omega_{z}v_{y}^{b}\\ \omega_{z}v_{z}^{b} - \omega_{x}v_{z}^{b}\\ \omega_{x}v_{y}^{b} - \omega_{y}v_{x}^{b} \end{bmatrix}, \\ \mathbf{P}_{2} = \frac{1}{m} \begin{bmatrix} F_{T}\\ 0\\ 0 \end{bmatrix}, \\ \mathbf{P}_{3} = \frac{1}{m}\mathbf{C}_{w}^{b} \begin{bmatrix} F_{x}^{w}\\ F_{y}^{w}\\ F_{z}^{w} \end{bmatrix}.$$
(A.0.13)

The partial derivatives of these three parts  $(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)$  will be calculated separately, and then summed up at the end. The first part is the easiest to handle. Considering Equation (A.0.13):

$$\mathbf{F}_{1}(4, [5\ 6\ 8\ 11\ 12]) = \frac{\partial}{\partial [v_{y}^{b}\ v_{z}^{b}\ p\ \omega_{y}\ \omega_{z}]^{T}} \mathbf{P}_{1}(1, 1) = [\omega_{z} \ -\omega_{y} \ -g\cos p \ -v_{z}^{b}\ v_{y}^{b}], \ (A.0.14)$$

$$\mathbf{F}_{1}(5, [4\ 6\ 7\ 8\ 10\ 12]) = \frac{\partial}{\partial [v_{x}^{b}\ v_{z}^{b}\ r\ p\ \omega_{x}\ \omega_{z}]^{T}} \mathbf{P}_{1}(2, 1)$$
$$= [-\omega_{z}\ \omega_{x}\ g\cos r\cos p\ -g\sin r\sin p\ v_{z}^{b}\ -v_{x}^{b}], \qquad (A.0.15)$$

$$\mathbf{F}_{1}(6, [4\ 5\ 7\ 8\ 10\ 11]) = \frac{\partial}{\partial [v_{x}^{b}\ v_{y}^{b}\ r\ p\ \omega_{x}\ \omega_{y}]^{T}} \mathbf{P}_{1}(3, 1)$$
$$= [\omega_{y} - \omega_{x} - g\sin r\cos p - g\cos r\sin p - v_{y}^{b}\ v_{x}^{b}].$$
(A.0.16)

Before handling the second and the third parts, it is worth calculating the partial derivatives of V,  $\alpha$ , and  $\beta$ , since they will appear frequently in those parts. Recalling Equations (1.1.15) and (1.1.17), and grouping the concerned variables into  $\eta \in \{v_x^b, v_y^b, v_z^b, r, p, y\}$ , it can be easily seen that

$$\frac{\partial V}{\partial \eta} = \frac{1}{V} \left( \mathbf{V} \cdot \frac{\partial \mathbf{V}}{\partial \eta} \right), \eta \in \{ v_x^b, v_y^b, v_z^b, r, p, y \},$$
(A.0.17)

$$\frac{\partial \alpha}{\partial \eta} = \frac{1}{V_v^2} \left( V_x^b \frac{\partial V_z^b}{\partial \eta} - V_z^b \frac{\partial V_x^b}{\partial \eta} \right),\tag{A.0.18}$$

$$\frac{\partial\beta}{\partial\eta} = \frac{1}{VV_v} \left( V \frac{\partial V_y^b}{\partial\eta} - V_y^b \frac{\partial V}{\partial\eta} \right),\tag{A.0.19}$$

\_

where

\_

~

$$V_{v} = \sqrt{V_{x}^{b^{2}} + V_{z}^{b^{2}}} = \sqrt{V^{2} - V_{y}^{b^{2}}},$$

$$\frac{\partial \mathbf{V}}{\partial u} = \begin{bmatrix} 1\\0\\0 \end{bmatrix}, \frac{\partial \mathbf{V}}{\partial v} = \begin{bmatrix} 0\\1\\0 \end{bmatrix}, \frac{\partial \mathbf{V}}{\partial w} = \begin{bmatrix} 0\\0\\1 \end{bmatrix},$$

$$\frac{\partial \mathbf{V}}{\partial r} = -\frac{\partial \mathbf{C}_{l}^{b}}{\partial r} \begin{bmatrix} w_{N}\\w_{E}\\w_{D} \end{bmatrix} = -\frac{\partial \mathbf{C}_{1}(r)}{\partial r} \mathbf{C}_{2}(p) \mathbf{C}_{3}(y) \begin{bmatrix} w_{N}\\w_{E}\\w_{D} \end{bmatrix},$$

$$\frac{\partial \mathbf{V}}{\partial p} = -\frac{\partial \mathbf{C}_{l}^{b}}{\partial p} \begin{bmatrix} w_{N}\\w_{E}\\w_{D} \end{bmatrix} = -\mathbf{C}_{1}(r) \frac{\partial \mathbf{C}_{2}(p)}{\partial p} \mathbf{C}_{3}(y) \begin{bmatrix} w_{N}\\w_{E}\\w_{D} \end{bmatrix},$$

$$\frac{\partial \mathbf{V}}{\partial y} = -\frac{\partial \mathbf{C}_{l}^{b}}{\partial y} \begin{bmatrix} w_{N}\\w_{E}\\w_{D} \end{bmatrix} = -\mathbf{C}_{1}(r) \mathbf{C}_{2}(p) \frac{\partial \mathbf{C}_{3}(y)}{\partial y} \begin{bmatrix} w_{N}\\w_{E}\\w_{D} \end{bmatrix}.$$
(A.0.20)

#### Manual Model Linearization for a Simplified Case

Now the second part can be handled more or less easily. According to Equations (A.0.13), (4.1.19) (replacing h by  $-x_D$ ), and (4.1.1):

$$\mathbf{F}_{2}(4,3) = \frac{\partial}{\partial x_{D}} \mathbf{P}_{2}(1,1) = \frac{1}{m} \frac{\partial F_{T}}{\partial x_{D}} = \frac{1}{m} \frac{\partial F_{T}}{\partial \rho} \frac{\partial \rho}{\partial x_{D}} = \frac{1}{m} \omega_{p}^{2} D^{4} C_{F_{T}} \frac{\partial \rho}{\partial x_{D}}.$$
 (A.0.21)

Considering Equations (A.0.13), (1.1.17), (4.1.1), and (4.1.2), and using Equation (A.0.17):

$$F_{2}(4, [4\ 5\ 6\ 7\ 8\ 9]) = \frac{\partial}{\partial [v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y]^{T}} \mathbf{P}_{2}(1, 1) = \frac{1}{m} \frac{\partial F_{T}}{\partial [v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y]^{T}} = \frac{1}{m} \frac{\partial F_{T}}{\partial [v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y]^{T}} = \frac{1}{m} \frac{\rho \omega_{p} D^{3}}{\pi} \left( C_{F_{T}2} + 2C_{F_{T}3} J \right) \frac{\partial V}{\partial [v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y]^{T}}.$$
(A.0.22)

Finally, the third part is handled as follows. Considering Equations (A.0.13), (1.1.16), (4.1.7), (4.1.5), and (4.1.3):

$$\mathbf{F}_{3}([4\ 5\ 6],3) = \frac{\partial \mathbf{P}_{3}}{\partial x_{D}} = \frac{1}{m} \mathbf{C}_{w}^{b} \frac{\partial}{\partial x_{D}} \begin{bmatrix} F_{w}^{w} \\ F_{y}^{w} \\ F_{z}^{w} \end{bmatrix}$$
$$= \frac{1}{m} \mathbf{C}_{w}^{b} \frac{\partial}{\partial \rho} \begin{bmatrix} F_{x}^{w} \\ F_{y}^{w} \\ F_{z}^{w} \end{bmatrix} \frac{\partial \rho}{\partial x_{D}} = \frac{1}{m} \mathbf{C}_{w}^{b} \frac{1}{\rho} \begin{bmatrix} F_{x}^{w} \\ F_{y}^{w} \\ F_{z}^{w} \end{bmatrix} \frac{\partial \rho}{\partial x_{D}}.$$
(A.0.23)

Note that from Equations (4.1.7), (4.1.5), and (4.1.3), it is obvious that

$$\frac{\partial}{\partial \rho} \begin{bmatrix} F_x^w \\ F_y^w \\ F_z^w \end{bmatrix} = \frac{1}{\rho} \begin{bmatrix} F_x^w \\ F_y^w \\ F_z^w \end{bmatrix}.$$
(A.0.24)

Before proceeding, partial derivatives of  $\mathbf{C}_w^b$ ,  $F_x^w$ ,  $F_y^w$ , and  $F_z^w$  with respect to  $\eta \in \{v_x^b, v_y^b, v_z^b, r, p, y\}$  are calculated. Considering Equations (1.1.16) and (1.1.17), and using Equations (A.0.18) and (A.0.19):

$$\frac{\partial \mathbf{C}_{w}^{b}}{\partial \eta} = \frac{\partial \left(\mathbf{C}_{2}(\alpha)\mathbf{C}_{3}^{T}(\beta)\right)}{\partial \eta} = \frac{\partial \mathbf{C}_{2}(\alpha)}{\partial \eta}\mathbf{C}_{3}^{T}(\beta) + \mathbf{C}_{2}(\alpha)\frac{\partial \mathbf{C}_{3}^{T}(\beta)}{\partial \eta} 
= \frac{\partial \mathbf{C}_{2}(\alpha)}{\partial \alpha}\frac{\partial \alpha}{\partial \eta}\mathbf{C}_{3}^{T}(\beta) + \mathbf{C}_{2}(\alpha)\frac{\partial \mathbf{C}_{3}^{T}(\beta)}{\partial \beta}\frac{\partial \beta}{\partial \eta} \tag{A.0.25}$$

According to Equations (4.1.3), (4.1.4), and (1.1.17), and using Equations (A.0.17), (A.0.18), and (A.0.19):

$$\frac{\partial F_x^w}{\partial \eta} = SC_{F_x} \frac{\partial \bar{q}}{\partial \eta} + \bar{q}S \frac{\partial C_{F_x}}{\partial \eta} = SC_{F_x} \rho V \frac{\partial V}{\partial \eta} + \bar{q}S \left( \frac{\partial C_{F_x}}{\partial \alpha} \frac{\partial \alpha}{\partial \eta} + \frac{\partial C_{F_x}}{\partial \beta} \frac{\partial \beta}{\partial \eta} \right)$$
$$= SC_{F_x} \rho V \frac{\partial V}{\partial \eta} + \bar{q}S \left( (C_{F_x\alpha} + 2C_{F_x\alpha 2}\alpha) \frac{\partial \alpha}{\partial \eta} + 2C_{F_x\beta 2}\beta \frac{\partial \beta}{\partial \eta} \right)$$
(A.0.26)

Similarly:

$$\frac{\partial F_y^w}{\partial \eta} = SC_{F_y} \frac{\partial \bar{q}}{\partial \eta} + \bar{q}S \frac{\partial C_{F_y}}{\partial \eta} = SC_{F_y} \rho V \frac{\partial V}{\partial \eta} + \bar{q}S \frac{\partial C_{F_y}}{\partial \beta} \frac{\partial \beta}{\partial \eta}$$
$$= SC_{F_y} \rho V \frac{\partial V}{\partial \eta} + \bar{q}SC_{F_y1} \frac{\partial \beta}{\partial \eta}$$
(A.0.27)

$$\frac{\partial F_z^w}{\partial \eta} = SC_{F_z} \frac{\partial \bar{q}}{\partial \eta} + \bar{q}S \frac{\partial C_{F_z}}{\partial \eta} = SC_{F_z} \rho V \frac{\partial V}{\partial \eta} + \bar{q}S \frac{\partial C_{F_z}}{\partial \alpha} \frac{\partial \alpha}{\partial \eta} 
= SC_{F_z} \rho V \frac{\partial V}{\partial \eta} + \bar{q}SC_{F_z\alpha} \frac{\partial \alpha}{\partial \eta}$$
(A.0.28)

Now the remaining elements of  $\mathbf{F}_3([4\ 5\ 6],:)$  can be calculated as

$$F_{3}([4\ 5\ 6], [4\ 5\ 6\ 7\ 8\ 9]) = \frac{\partial \mathbf{P}_{3}}{\partial [v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y]^{T}} = \frac{1}{m} \left( \frac{\partial \mathbf{C}_{w}^{b}}{\partial [v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y]^{T}} \begin{bmatrix} F_{x}^{w} \\ F_{y}^{w} \\ F_{z}^{w} \end{bmatrix} + \mathbf{C}_{w}^{b} \frac{\partial}{\partial [v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y]^{T}} \begin{bmatrix} F_{x}^{w} \\ F_{y}^{w} \\ F_{z}^{w} \end{bmatrix} \right).$$
(A.0.29)

Eventually,  $\mathbf{F}([4 \ 5 \ 6], :)$  is calculated as

$$\mathbf{F}([4\ 5\ 6],:) = \mathbf{F}_1([4\ 5\ 6],:) + \mathbf{F}_2([4\ 5\ 6],:) + \mathbf{F}_3([4\ 5\ 6],:).$$
(A.0.30)

Note that all the elements of  $\mathbf{F}_1([4 \ 5 \ 6], :), \ \mathbf{F}_2([4 \ 5 \ 6], :), \ and \ \mathbf{F}_3([4 \ 5 \ 6], :)$  that are not represented, are zero.

## Rows 7 to 9 of F matrix

In a more or less similar manner to rows 4, 5, and 6; rows 7, 8, and 9 of F are calculated according to Equation (A.0.4).

$$\mathbf{F}([7\ 8\ 9], [10\ 11\ 12]) = \frac{\partial}{\partial [\omega_x \ \omega_y \ \omega_z]^T} \begin{bmatrix} \dot{r} \\ \dot{p} \\ \dot{y} \end{bmatrix} = \mathbf{C}_{\omega}$$
(A.0.31)

$$\mathbf{F}([7\ 8\ 9], 7) = \frac{\partial}{\partial r} \begin{bmatrix} \dot{r} \\ \dot{p} \\ \dot{y} \end{bmatrix} = \frac{\partial \mathbf{C}_{\omega}}{\partial r} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$
(A.0.32)

$$\mathbf{F}([7\ 8\ 9], 8) = \frac{\partial}{\partial p} \begin{bmatrix} \dot{r} \\ \dot{p} \\ \dot{y} \end{bmatrix} = \frac{\partial \mathbf{C}_{\omega}}{\partial p} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$
(A.0.33)

$$\mathbf{F}([7\ 8\ 9], [1\ 2\ 3\ 4\ 5\ 6\ 9]) = \frac{\partial}{\partial [x_N\ x_E\ x_D\ v_x^b\ v_y^b\ v_z^b\ y]^T} \begin{bmatrix} \dot{r} \\ \dot{p} \\ \dot{y} \end{bmatrix} = [0]_{3\times7}$$
(A.0.34)

#### Rows 10 to 12 of F matrix

To calculate rows 10, 11, and 12 of F, Equation (A.0.5) is rewritten as

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \mathbf{Q}_1 + \mathbf{Q}_2, \tag{A.0.35}$$

where

$$\mathbf{Q}_1 = -(\mathbf{I}^b)^{-1} \left( \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \mathbf{I}^b \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \right), \mathbf{Q}_2 = (\mathbf{I}^b)^{-1} \begin{bmatrix} M_x^b \\ M_y^b \\ M_z^b \end{bmatrix}.$$
(A.0.36)

The partial derivatives of these two parts  $(\mathbf{Q}_1, \mathbf{Q}_2)$  will be calculated separately, and then summed up at the end. The first part is easy to handle. It is first rewritten as

$$\mathbf{Q}_{1} = \begin{bmatrix} \omega_{y}\omega_{z} + J_{1}J_{2}\omega_{y}(I_{xz}\omega_{x} + I_{z}\omega_{z}) \\ \frac{1}{I_{y}}\left(I_{xz}(\omega_{x}^{2} - \omega_{z}^{2}) - (I_{x} - I_{z})\omega_{x}\omega_{z}\right) \\ -\omega_{x}\omega_{y} - J_{1}J_{2}\omega_{y}(I_{x}p + I_{xz}\omega_{z}) \end{bmatrix}, \begin{cases} J_{1} = \frac{1}{I_{x}I_{z} - I_{xz}^{2}} \\ J_{2} = -I_{x} + I_{y} - I_{z}. \end{cases}$$
(A.0.37)

Now  $\mathbf{F}_1([10\ 11\ 12],:)$  can be easily calculated as

$$\mathbf{F}_{1}([10\ 11\ 12], [10\ 11\ 12]) = \frac{\partial \mathbf{Q}_{1}}{\partial [\omega_{x}\ \omega_{y}\ \omega_{z}]^{T}} = \\ \begin{bmatrix} J_{1}J_{2}I_{xz}\omega_{y} & J_{1}J_{2}I_{xz}\omega_{x} + (1+J_{1}J_{2}I_{z})\omega_{z} & (1+J_{1}J_{2}I_{z})\omega_{y} \\ \frac{1}{I_{y}}\left(2I_{xz}\omega_{x} - (I_{x} - I_{z})\omega_{z}\right) & 0 & \frac{1}{I_{y}}\left(-2I_{xz}\omega_{z} - (I_{x} - I_{z})\omega_{x}\right) \\ -(1+J_{1}J_{2}I_{x})\omega_{y} & -(1+J_{1}J_{2}I_{x})\omega_{x} - J_{1}J_{2}I_{xz}\omega_{z} & -J_{1}J_{2}I_{xz}\omega_{y} \end{bmatrix}$$

$$(A.0.38)$$

To handle the second part,  $\mathbf{Q}_2$  is first rewritten as

$$\mathbf{Q}_{2} = \begin{bmatrix} J_{x}M_{x}^{b} + J_{xz}M_{z}^{b} \\ J_{y}M_{y}^{b} \\ J_{xz}M_{x}^{b} + J_{z}M_{z}^{b} \end{bmatrix}, \begin{bmatrix} J_{x} & 0 & J_{xz} \\ 0 & J_{y} & 0 \\ J_{xz} & 0 & J_{z} \end{bmatrix} = (\mathbf{I}^{b})^{-1} = \begin{bmatrix} I_{x} & 0 & I_{xz} \\ 0 & I_{y} & 0 \\ I_{xz} & 0 & I_{z} \end{bmatrix}^{-1}, \quad (A.0.39)$$

and then partial derivatives of  $M_x^b$ ,  $M_y^b$ , and  $M_z^b$  with respect to  $x_D$ ,  $\eta \in \{v_x^b, v_y^b, v_z^b, r, p, y\}$ , p, q, and r are calculated. According to Equations (4.1.12), (1.1.17), (4.1.10), and (4.1.13), and using Equations (A.0.17), (A.0.18), and (A.0.19):

$$\frac{\partial M_x^b}{\partial x_D} = \frac{\partial M_x^b}{\partial \rho} \frac{\partial \rho}{\partial x_D} = \frac{M_x^b}{\rho} \frac{\partial \rho}{\partial x_D}$$
(A.0.40)

$$\frac{\partial M_x^b}{\partial \eta} = SbC_{M_x} \frac{\partial \bar{q}}{\partial \eta} + \bar{q}Sb \frac{\partial C_{M_x}}{\partial \eta} 
= SbC_{M_x} \rho V \frac{\partial V}{\partial \eta} + \bar{q}Sb \left( C_{M_x\beta} \frac{\partial \beta}{\partial \eta} - \frac{1}{V} C_{M_x \tilde{\omega}_x} \tilde{\omega}_x \frac{\partial V}{\partial \eta} - \frac{1}{V} C_{M_x \tilde{\omega}_z} \tilde{\omega}_z \frac{\partial V}{\partial \eta} \right) 
= \bar{q}SbC_{M_x\beta} \frac{\partial \beta}{\partial \eta} + \rho VSb \left( C_{M_x} - \frac{C_{M_x \tilde{\omega}_x} \tilde{\omega}_x + C_{M_x \tilde{\omega}_z} \tilde{\omega}_z}{2} \right) \frac{\partial V}{\partial \eta}$$
(A.0.41)

$$\frac{\partial M_x^b}{\partial p} = \frac{\bar{q}Sb^2}{2V} C_{M_x \tilde{\omega}_x} \tag{A.0.42}$$

$$\frac{\partial M_x^b}{\partial q} = 0 \tag{A.0.43}$$

$$\frac{\partial M_x^b}{\partial r} = \frac{\bar{q}Sb^2}{2V} C_{M_x \tilde{\omega}_z} \tag{A.0.44}$$

Similarly:

$$\frac{\partial M_y^b}{\partial x_D} = \frac{\partial M_y^b}{\partial \rho} \frac{\partial \rho}{\partial x_D} = \frac{M_y^b}{\rho} \frac{\partial \rho}{\partial x_D} \tag{A.0.45}$$

$$\frac{\partial M_y^b}{\partial \eta} = \bar{q} S \bar{c} C_{M_y \alpha} \frac{\partial \alpha}{\partial \eta} + \rho V S \bar{c} \left( C_{M_y} - \frac{C_{M_y \tilde{\omega}_y} \tilde{\omega}_y}{2} \right) \frac{\partial V}{\partial \eta}$$
(A.0.46)

$$\frac{\partial M_y^b}{\partial p} = 0 \tag{A.0.47}$$

$$\frac{\partial M_y^b}{\partial q} = \frac{1}{4} S \bar{c}^2 \rho C_{M_y \tilde{\omega}_y} V \tag{A.0.48}$$

$$\frac{\partial M_y^b}{\partial r} = 0 \tag{A.0.49}$$

Also:

$$\frac{\partial M_z^b}{\partial x_D} = \frac{\partial M_z^b}{\partial \rho} \frac{\partial \rho}{\partial x_D} = \frac{M_z^b}{\rho} \frac{\partial \rho}{\partial x_D}$$
(A.0.50)

$$\frac{\partial M_z^b}{\partial \eta} = \bar{q} S b C_{M_z \beta} \frac{\partial \beta}{\partial \eta} + \rho V S b \left( C_{M_z} - \frac{C_{M_z \tilde{\omega}_z} \tilde{\omega}_z}{2} \right) \frac{\partial V}{\partial \eta}$$
(A.0.51)

$$\frac{\partial M_z^b}{\partial p} = 0 \tag{A.0.52}$$

$$\frac{\partial M_z^b}{\partial q} = 0 \tag{A.0.53}$$

$$\frac{\partial M_z^b}{\partial r} = \frac{1}{4} S b^2 \rho C_{M_z \tilde{\omega}_z} V \tag{A.0.54}$$

Using Equation (A.0.39) along with Equations (A.0.40) to (A.0.54),  $\mathbf{F}_2([10\ 11\ 12],:)$  can be expressed as

$$\mathbf{F}_{2}([10\ 11\ 12], [3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12]) = \frac{\partial [\dot{\omega}_{x}\ \dot{\omega}_{y}\ \dot{\omega}_{z}\ ]^{T}}{\partial [x_{D}\ v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y\ \omega_{x}\ \omega_{y}\ \omega_{z}\ ]^{T}} = \begin{bmatrix} J_{x} \frac{\partial M_{x}^{b}}{\partial [x_{D}\ v_{x}^{b}\ v_{y}^{b}\ r\ p\ y\ \omega_{x}\ \omega_{y}\ \omega_{z}\ ]^{T}} + J_{xz} \frac{\partial M_{z}^{b}}{\partial [x_{D}\ v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y\ \omega_{x}\ \omega_{y}\ \omega_{z}\ ]^{T}} \\ J_{y} \frac{\partial M_{y}^{b}}{\partial [x_{D}\ v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y\ \omega_{x}\ \omega_{y}\ \omega_{z}\ ]^{T}} \\ J_{xz} \frac{\partial M_{x}^{b}}{\partial [x_{D}\ v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y\ \omega_{x}\ \omega_{y}\ \omega_{z}\ ]^{T}} + J_{z} \frac{\partial M_{y}^{b}}{\partial [x_{D}\ v_{x}^{b}\ v_{y}^{b}\ v_{z}^{b}\ r\ p\ y\ \omega_{x}\ \omega_{y}\ \omega_{z}\ ]^{T}} \end{bmatrix}.$$
(A.0.55)

Eventually,  $\mathbf{F}([10\ 11\ 12],:)$  is calculated as

$$\mathbf{F}([10\ 11\ 12],:) = \mathbf{F}_1([10\ 11\ 12],:) + \mathbf{F}_2([10\ 11\ 12],:).$$
(A.0.56)

Note that all the elements of  $\mathbf{F}_1([10\ 11\ 12],:)$  and  $\mathbf{F}_2([10\ 11\ 12],:)$  that are not represented, are zero.

At this point, calculation of F is completed for this simplified model. Considering all the simplifications in VDM mentioned at the beginning of this appendix and the fact that the state vector contained **only the navigation states** and the observation models were not treated here, one can realize how complex the manual linearization would be for the final model used in this research as presented in Chapters 3 and 4.

## Bibliography

- Federal Aviation Administration. Unmanned Aircraft Systems (UAS) Regulations & Policies. https://www.faa.gov/uas/resources/uas\_regulations\_policy. Accessed 2016-12-08.
- [2] Simon L. Altmann. Rotation, Quaternions, and Double Groups. Clarendon Press, Oxford, England, 1986.
- [3] Cesario Vincenzo Angelino, Vincenzo Rosario Baraniello, and Luca Cicala. UAV Position and Attitude Estimation using IMU, GNSS and Camera. In 15th International Conference on Information Fusion (FUSION), pages 735–742, Singapore, Singapore, 2012.
- [4] Ardupilot. Pixhawk 2. http://ardupilot.org/copter/docs/common-pixhawk2-overview. html. Accessed 2018-01-14.
- [5] John W. Betz. Engineering Satellite-Based Navigation and Timing: Global Navigation Satellite Systems, Signals, and Receivers. John Wiley & Sons, Hoboken, New Jersey, USA, 2016.
- [6] Haider Ali Biswas, Timi Fahria Haque, Eliyas Karim, and Ashikur Rahman. Representation of Hamiltonian Formalism in Dissipative Mechanical System. *Applied Mathematical Sciences*, 4(19):931–942, 2010.
- [7] Alexander A. Bogdanov, Eric A. Wan, Magnus Carlsson, Yinglong Zhang, Richard Kieburtz, and Antonio Baptista. Model Predictive Neural Control of a High Fidelity Helicopter Model. In *Proceedings of the AIAA Guidance Navigation and Control Conference*, Montreal, Canada, 2001.
- [8] Mitch Bryson and Salah Sukkarieh. Vehicle Model Aided Inertial Navigation for a UAV using Low-Cost Sensors. In Proceedings of the Australasian Conference on Robotics and Automation, Canberra, Australia, 2004.
- [9] Mitch Bryson and Salah Sukkarieh. UAV Localization Using Inertial Sensors and Satellite Positioning Systems. In *Handbook of Unmanned Aerial Vehicles*, pages 433–460. Springer, Netherlands, 2015.
- [10] Transport Canada. Flying Your Drone Safely and Legally. https://www.tc.gc.ca/eng/ civilaviation/opssvs/flying-drone-safely-legally.html. Accessed 2016-12-08.
- [11] Han-Pang Chiu, Aveek Das, Phillip Miller, Supun Samarasekera, and Rakesh Kumar. Precise Vision-Aided Aerial Navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 688–695, Chicago, Illinois, USA, 2014.

- [12] Philipp Clausen, Jan Skaloud, Roberto Molinari, James Balamuta, and Stéphane Guerrier. An Overview of a New Sensor Calibration Platform. In 2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), pages 364–368, 2017.
- [13] Michael V. Cook. Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control. Butterworth-Heinemann, Waltham, Massachusetts, USA, 2013.
- [14] Lennon R. Cork. Aircraft Dynamic Navigation for Unmanned Aerial Vehicles. PhD thesis, Queensland University of Technology, Brisbane, Australia, 2014.
- [15] Philipp Crocoll, Justus Seibold, Georg Scholz, and Gert F. Trommer. Model-Aided Navigation for a Quadrotor Helicopter: A Novel Navigation System and First Experimental Results. *Navigation*, 61(4):253–271, 2014.
- [16] Philipp Crocoll and Gert F. Trommer. Quadrotor Inertial Navigation Aided by a Vehicle Dynamics Model with In-Flight Parameter Estimation. In Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), pages 1784–1795, Tampa, Florida, USA, 2014.
- [17] Navid Dadkhah, Bernard Mettler, and Demoz Gebre-Egziabher. A Model-Aided AHRS for Micro Aerial Vehicle Application. In Proceedings of the 21st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2008), pages 545–553, Savannah, Georgia, USA, 2008.
- [18] Gamini Dissanayake, Salah Sukkarieh, Eduardo Nebot, and Hugh Durrant-Whyte. The Aiding of a Low-Cost Strapdown Inertial Measurement Unit using Vehicle Model Constraints for Land Vehicle Applications. *IEEE Transactions on Robotics and Automation*, 17(5):731–747, 2001.
- [19] Guillaume Ducard. Fault-Tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles. Springer, London, England, 2009.
- [20] James L. Farrell. Integrated Aircraft Navigation. Academic Press, New York, New York, USA, 1976.
- [21] Jay A. Farrell. Aided Navigation: GPS with High Rate Sensors. McGraw-Hill, New York, New York, USA, 2008.
- [22] Vladislav Gavrilets. Autonomous Aerobatic Maneuvering of Miniature Helicopters. PhD thesis, MIT, Cambridge, Massachusetts, USA, 2003.
- [23] Arthur Gelb, editor. Applied Optimal Estimation. The MIT Press, Cambridge, Massachusetts, USA, 1988.
- [24] Herbert Goldstein, Charles Poole, and John Safko. Classical Mechanics. Addison-Wesley, New York, New York, USA, 3rd edition, 2002.
- [25] Charles A. Greenhall. Recipes for Degrees of Freedom of Frequency Stability Estimators. IEEE Transactions on Instrumentation and Measurement, 40(6):994–999, 1991.
- [26] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. Global Positioning Systems, Inertial Navigation, and Integration. John Wiley & Sons, Hoboken, New Jersey, USA, 2nd edition, 2007.

- [27] Paul D. Groves. Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems. GNSS technology and applications series. Artech House, Boston, Massachusetts, USA, 2008.
- [28] Stéphane Guerrier, Roberto Molinari, and Yannick Stebler. Theoretical Limitations of Allan Variance-based Regression for Time Series Model Estimation. *IEEE Signal Processing Letters*, 23(5):597–601, 2016.
- [29] Stéphane Guerrier, Jan Skaloud, Yannick Stebler, and Maria-Pia Victoria-Feser. Wavelet-Variance-based Estimation for Composite Stochastic Processes. Journal of the American Statistical Association, 108(503):1021–1030, 2013.
- [30] Elder M. Hemerly and Valter R. Schad. Implementation of an AHRS with GNSS Baseline Readings. Journal of Aerospace Engineering, 2(3):1–10, 2010.
- [31] Klaus A. Hoffman and Steve T. Chiang. Computational Fluid Dynamics. Engineering Education System, Wichita, Kansas, USA, 2000.
- [32] Bernhard Hofmann-Wellenhof, Klaus Legat, and Manfred Wieser. Navigation: Principles of Positioning and Guidance. Springer Science & Business Media, New York, New York, USA, 2011.
- [33] Clay Mathematics Institute. Millennium Problems: Navier–Stokes Equation. http: //www.claymath.org/millennium-problems/navier%E2%80%93stokes-equation. Accessed 2017-11-21.
- [34] Intersense. Navchip. http://www.intersense.com/pages/16/246. Accessed 2015-12-10.
- [35] Rudolph E. Kalman. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1):35–45, 1960.
- [36] Mehran Khaghani and Jan Skaloud. Application of Vehicle Dynamic Modeling in UAVs for Precise Determination of Exterior Orientation. In *ISPRS - International Archives of* the Photogrammetry, Remote Sensing and Spatial Information Sciences, volume XLI-B3, pages 827–831, Prague, Czech Republic, 2016.
- [37] Mehran Khaghani and Jan Skaloud. Autonomous Navigation of Small UAVs based on Vehicle Dynamic Model. In ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, volume XL-3/W4, pages 117–122, Lausanne, Switzerland, 2016.
- [38] Mehran Khaghani and Jan Skaloud. Autonomous Vehicle Dynamic Model-Based Navigation for Small UAVs. Navigation, 63(3):345–358, 2016.
- [39] Mehran Khaghani and Jan Skaloud. Evaluation of Wind Effects on UAV Autonomous Navigation Based on Vehicle Dynamic Model. In Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), pages 1432–1440, Portland, Oregon, USA, 2016.
- [40] Mehran Khaghani and Jan Skaloud. Assessment of VDM-based Autonomous Navigation of a UAV under Operational Conditions. *Robotics and Autonomous Systems*, Under review, 2017.

#### Bibliography

- [41] Mehran Khaghani and Jan Skaloud. Positioning Autonomy of a Fixed-Wing UAV through VDM/INS Integration with Experimental Results. In Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017), pages 2332–2337, Portland, Oregon, USA, 2017.
- [42] Ties Kluter. GECKO4NAV Technical Reference Manual. HuCE-mircroLab, Bern University of Applied Sciences, Bern, Switzerland, 2013.
- [43] KNMI and Alterra. Data Provided under the Consortium Agreement on Cabauw Experimental Site for Atmospheric Research (CESAR). Netherlands, 2012.
- [44] Mark Koifman and Itzhack Y. Bar-Itzhack. Inertial Navigation System Aided by Aircraft Dynamics. IEEE Transactions on Control Systems Technology, 7(4):487–493, 1999.
- [45] Lev D. Landau and Evgeny M. Lifshitz. Fluid Mechanics. Pergamon Press, Oxford, England, 2nd edition, 1987.
- [46] Tak Kit Lau, Yun-Hui Liu, and Kai Wun Lin. Inertial-Based Localization for Unmanned Helicopters Against GNSS Outage. *IEEE Transactions on Aerospace and Electronic Systems*, 49(3):1932–1949, 2013.
- [47] Bruno Luiset. Développement d'un baromètre altimètre à bord d'un drone. Master's thesis, CNAM ESGT, Le Mans, France, 2016.
- [48] Yasser M. Madany, Hassan M. Elkamchouchi, and Mostafa M. Ahmed. Modelling and Simulation of Robust Navigation for Unmanned Air Systems (UASs) Based on Integration of Multiple Sensors Fusion Architecture. In *European Modelling Symposium (EMS)*, pages 719–724, Manchester, England, 2013.
- [49] Maxtena. Rugged L1/L2 GPS GLONASS Active Antenna. http://www.maxtena.com/ products/helicore/m1227hct-a2-sma/?v=1ee0bf89c5d1. Accessed 2016-10-31.
- [50] Lorenz Meier, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight using Onboard Computer Vision. Autonomous Robots, 33(1-2):21–39, 2012.
- [51] Ministère de l'environnement, de l'énergie et de la mer. Drones civils: loisir? activité professionnelle? http://www.developpement-durable.gouv.fr/Drones-civils-loisir-activite. Accessed 2016-12-08.
- [52] Karsten Müller, Philipp Crocoll, and Gert F. Trommer. Wind Estimation for a Quadrotor Helicopter in a Model-Aided Navigation System. In 22nd Saint Petersburg International Conference on Integrated Navigation Systems, Saint Petersburg, Russia, 2015.
- [53] Karsten Müller, Philipp Crocoll, and Gert F. Trommer. Model-Aided Navigation with Wind Estimation for Robust Quadrotor Navigation. In Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016), pages 689–696, Monterey, California, USA, 2016.
- [54] Sameh Nassar and Naser El-Sheimy. Wavelet Analysis for Improving INS and INS/DGPS Navigation Accuracy. Journal of Navigation, 58(1):119–134, 2005.

- [55] Sameh Nassar and Naser El-Sheimy. A Combined Algorithm of Improving INS Error Modeling and Sensor Measurements for Accurate INS/GPS Navigation. GPS Solutions, 10(1):29–39, 2006.
- [56] Luat T. Nguyen, Marilyn E. Ogburn, William P. Gilbert, Kemper S. Kibler, Philip W. Brown, and Perry L. Deal. Simulator Study of Stall/Post-Stall Characteristics of a Fighter Airplane with Relaxed Longitudinal Static Stability. Technical report NASA-TP-1538, L-12854, NASA Langley Research Center, Hampton, New Hampshire, USA, 1979.
- [57] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles. Springer, Tokyo, Japan, 2010.
- [58] Aboelmagd Noureldin, Tashfeen B. Karamat, Mark D. Eberts, and Ahmed El-Shafie. Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost Navigation Applications. *IEEE Transactions on Vehicular Technology*, 58(3):1077–1096, 2009.
- [59] NIMA WGS84 Update Committee. Department of Defense. World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems. Technical report, National Imagery and Mapping Agency, Springfield, Virginia, USA, 3rd edition, 2000.
- [60] Martin Rehak and Jan Skaloud. Fixed-wing Micro Aerial Vehicle for Accurate Corridor Mapping. In ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, volume II-1/W1, pages 23–31, Toronto, Canada, 2015.
- [61] Florian Reinhard. Micro Unmanned Aerial Vehicle Motion Classification. Master's thesis, EPFL, Lausanne, Switzerland, 2017.
- [62] Matthew Rhudy, Yu Gu, Jason Gross, and Marcello R. Napolitano. Evaluation of Matrix Square Root Operations for UKF within a UAV GPS/INS Sensor Fusion Application. International Journal of Navigation and Observation, 2011:1–11, 2011.
- [63] Robert M. Rogers. Applied Mathematics in Integrated Navigation Systems. American Institute of Aeronautics and Astronautics, Reston, Virginia, USA, 3rd edition, 2007.
- [64] Jan Roskam. Airplane Flight Dynamics and Automatic Flight Controls. DARcorporation, Lawrence, Kansas, USA, 2001.
- [65] Andrea Saltelli, editor. Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models. John Wiley, Chichester, England, 2004.
- [66] Andrea Saltelli, editor. Global Sensitivity Analysis: The Primer. John Wiley, Chichester, England, 2008.
- [67] Klaus-Peter Schwarz and Ming Wei. Modeling INS/GPS for Attitude and Gravity Applications. In Proceedings of the 3rd International Workshop of High Precision Navigation, pages 200–218, Stuttgart, Germany, 1994.
- [68] Alexander Sendobry. Control System Theoretic Approach to Model Based Navigation. Ingenieurwissenschaftlicher Verlag, Bonn, Germany, 2014.

- [69] Bassem Sheta. Vision based Navigation (VBN) of Unmanned Aerial Vehicles (UAV). PhD thesis, University of Calgary, Calgary, Canada, 2012.
- [70] Eun-Hwan Shin. Estimation Techniques for Low-Cost Inertial Navigation. PhD thesis, University of Calgary, Calgary, Canada, 2005.
- [71] Yannick Stebler. Modeling and Processing Approaches for Integrated Inertial Navigation. PhD thesis, EPFL, Lausanne, Switzerland, 2013.
- [72] 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.
- [73] Stephen T. Thornton and Jerry B. Marion. Classical Dynamics of Particles and Systems. Brooks/Cole, Belmont, USA, 5th edition, 2004.
- [74] David Titterton and John L. Weston. Strapdown Inertial Navigation Technology. The Institution of Engineering and Technology, London, England, 2nd edition, 2004.
- [75] Phillip Tomé. Integration of Inertial and Satellite Navigation Systems for Aircraft Attitude Determination. PhD thesis, University of Oporto, Oporto, Portugal, 2002.
- [76] U-Blox. NEO/LEA-8T. https://www.u-blox.com/en/product/neolea-m8t. Accessed 2016-09-04.
- [77] Maarten Uijt de Haag, Pengfei Duan, and Ananth Vadlamani. Flight Test and Simulation Results of an Integrated Dual Airborne Laser Scanner and Inertial Navigator for UAV Applications. In 11th International Multi-Conference on Systems, Signals & Devices (SSD), pages 1–6, Castelldefels, Spain, 2014.
- [78] Rudolph Van Der Merwe, Eric A. Wan, and Simon Julier. Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion: Applications to Integrated Navigation. In *Proceedings of the AIAA Guidance, Navigation & Control Conference*, pages 16–19, Providence, Rhode Island, USA, 2004.
- [79] José Fernandes Vasconcelos, Carlos Silvestre, Paulo Oliveira, and Bruno J. N. Guerreiro. Embedded UAV Model and LASER Aiding Techniques for Inertial Navigation Systems. *Control Engineering Practice*, 18(3):262–278, 2010.
- [80] Jinling Wang, Matthew Garratt, Andrew Lambert, Jack Jianguo Wang, Songlai Han, and David Sinclair. Integration of GPS/INS/Vision Sensors to Navigate Unmanned Aerial Vehicles. In ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, volume XXXVII-B1, pages 963–970, Beijing, China, 2008.
- [81] Zhiqiang Xing. Over-bounding Integrated INS/GNSS Output Errors. PhD thesis, University of Minnesota, Minneapolis, Minnesota, USA, 2010.
- [82] Sukchang Yun, Young Jae Lee, and Sangkyung Sung. IMU/Vision/Lidar Integrated Navigation System in GNSS Denied Environments. In 2013 IEEE Aerospace Conference, pages 1–10, Big Sky, Montana, USA, 2013.