Contributions to Tight Formation Flight Control of Small UAS

Bolting Jan

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)*

Présentée et soutenue le *(26/09/2017)* **par :**

JAN BOLTING

Contributions to Tight Formation Flight Control of Small UAS

JURY

PETER HECKER	Rapporteur
MARC JUNGERS	Rapporteur
CHRISTOPHE CROS	Examinateur
ISABELLE QUEINNEC	Examinateur
JEAN-MARC BIANNIC	Directeur de thèse
FRANÇOIS DEFAŸ	Co-directeur de thèse

École doctorale et spécialité :
EDSYS : Automatique 4200046
Unité de Recherche :
ISAE (DCAS)
Directeur(s) de Thèse :
Jean-Marc Biannic et *François Defaÿ*
Rapporteurs :
Peter Hecker et *Marc Jungers*

Abstract

Small, electrically driven unmanned aircraft are likely to suffer from inferior endurance compared to their larger counterparts, attributable mostly to the limited aerodynamic efficiency of small wings of moderate aspect ratio and the comparably low energy density of available civil battery technology. Upwash exploitation by tight formation flight, inspired by migratory birds, as well as aerial recharging are the most promising control-driven approaches to mitigate this disadvantage while simultaneously opening up new possibilities such as distributed payloads and mission-level redundancy.

Flight experiments with manned aircraft as well as wind tunnel data suggest that the region in the wake of a predecessor aircraft where a significant reduction in energy consumption can be achieved is however both laterally and vertically of the magnitude of a mere fraction of a wingspan. The docking procedure involved in exchanging batteries between Unmanned Aircraft Systems (UAS) in flight implies similar, if not even tighter, guidance performance requirements. Precise formation flight has, furthermore, to be performed under the adverse influence of only approximately known wake disturbances.

Continuous time higher order sliding mode control (CTHOSM) has been considered as a candidate for this challenging open problem and was successfully applied to simple kinematic models in simulation, where excellent relative position tracking performance can be demonstrated. In this work we study the implications of the presence of inner loop dynamics and discrete implementation at moderate sampling rates and we find that it precludes the application of CTHOSM control to fixed-wing UAS. We propose a predictive discrete sliding mode guidance scheme to approximate the performance of CTHOSM control assuming realistic fixed-wing UAS dynamics. Comparisons with time sampled continuous-time sliding mode control previously proposed show vastly improved chattering and position tracking performance. We show that the proposed guidance scheme in combination with inner load factor tracking loops and a disturbance observer allows for relative position tracking performance compatible with the requirements of upwash exploitation. We address the scalability of local feedback sliding mode control schemes and point out fundamental limitations of sliding mode control with local feedback information when it comes to string stability.
To evaluate the performance of the proposed guidance scheme and to facilitate the testing and comparison of new, improved formation flight control techniques, an openly accessible benchmark problem is proposed in this work.

Finally, the closeness of UAS in tight formation flight scenarios makes collision avoidance a major concern. However, existing probabilistic localization algorithms cannot provide guaranteed confidence regions of the relative position of other members of the

formation. As an important building block for safe tight formation flight, we present a set membership filter that provides ellipsoidal regions guaranteed to contain the relative positions of the other UAS. Being based on observations of single-frequency Global Satellite Navigation System (GNSS) receivers and ultra-wide band ranging devices, it is compatible with the hardware constraints of small low-cost UAS. Simulations suggest computational efforts compatible with the computational resources typically available onboard small UAS.

Résumé

Les mini-drones à propulsion électrique sont susceptibles d'avoir une endurance inférieure à celle de drones plus grands, en raison, principalement, de l'efficacité aérodynamique limitée des petites ailes de faible allongement. La capacité de stockage limitée des batteries embarquables sur les mini-drones réduit également l'endurance totale. L'exploitation des interactions aérodynamiques, inspirée par les oiseaux migratoires, ainsi que le ravitaillement en vol , sont des approches prometteuses pour améliorer l'endurance des mini-drones tout en permettant une distribution de la charge utile.

Dans le contexte du vol en formation classique des avions, on a observé qu'une réduction significative de la consommation d'énergie de l'avion suiveur peut être obtenue dès lors qu'il se place dans les tourbillons de sillage du prédécesseur. Dans notre contexte, cela implique des déplacements très précis du mini-drone suiveur dont la position relative (latérale et verticale) doit rester à une fraction d'envergure du mini-drone prédécesseur. Par ailleurs, la procédure d'amarrage utilisée dans l'échange de batteries entre drones en vol implique des exigences de performance de guidage similaires, voire plus strictes. Ces fortes contraintes de performance s'accompagnent en outre d'exigences de robustesse particulièrement élevées sur les lois de commande en raison des perturbations mal connues induites par les turbulences de sillage.

La commande par modes glissants d'ordre supérieur en temps continu (CTHOSM) a été considérée comme un candidat prometteur à ce problème ouvert difficile et a été appliquée avec succès à des modèles cinématiques simples. Des performances excellentes de suivi de position relative ont en effet pu être démontrées. Dans nos travaux, nous étudions les implications de la présence de la dynamique de la boucle interne et de l'implémentation en temps discret à des taux d'échantillonnage modérés et constatons alors que l'application de la commande CTHOSM devient impossible. Nous proposons donc un schéma de guidage prédictif discret par modes glissants pour approximer les performances de la commande CTHOSM pour une dynamique réaliste du drone. La comparaison avec la commande par modes glissants en temps continu échantillonnée proposée dans la littérature montre des performances de suivi de position considérablement améliorées. Nous montrons que la loi de guidage proposée, complétée par des boucles de suivi du facteur de charge interne et d'un estimateur des perturbations permet de garantir une précision sur le suivi de position relative compatible avec les exigences de l'exploitation du flux d'air généré par le prédécesseur. Dans nos travaux, nous abordons aussi, avec la notion de « string stability », la problématique liée à la propagation des erreurs de suivi dans des systèmes de commande par modes glissants sous retour d'états locaux et soulignons les limitations fondamentales. Pour évaluer la performance de la loi de guidage proposée et permettre aussi à la communauté de tester diverses améliorations des techniques de commande de vol dédiées au vol en formation de drones, un problème de référence ouvert a été proposé dans le

cadre de ce travail.

Enfin, la proximité des drones dans les scénarios de vol en formation serrée nécessite la mise en place de stratégies efficaces d'évitement de collisions. Cependant, les algorithmes de localisation probabilistes existants ne permettent pas la caractérisation de régions de confiance garanties de la position des autres membres de la formation. Dans ce contexte, nous proposons un nouveau filtre ensembliste caractérisant de telles régions de confiance sous forme ellipsoïdale. La mise en œuvre du filtre proposé repose uniquement sur des mesures accessibles via des récepteurs GNSS à faible coût et des capteurs de distance relatifs. Nos premières évaluations ont montré que les efforts de calcul induits par cette mise en œuvre restent parfaitement compatibles avec les contraintes des systèmes avioniques des petits drones.

Contents

Glossary

List of Acronyms

AHRS	Attitude and Heading Reference System
COTS	Commercial off-the-shelf
DLL	Delay Lock Loop
DSMC	Discrete Sliding Mode Control
EKF	Extended Kalman Filter
ESMF	Extended Set Membership Filter
FOM	Figure of Merit
GNSS	Global Navigation Satellite System
HOSM	Higher Order Sliding Mode
IMU	Inertial Measurement Unit
LMPC	Linear Model predictive control
LQ	Linear Quadratic
LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
MEMS	Microelectromechanical systems
MVE	Minimum Volume Ellipsoid
NDI	Nonlinear Dynamic Inversion
NED	North-East-Down
PDSM	Predictive Discrete Sliding Mode
PDSMC	Predictive Discrete Sliding Mode Control
PID	Proportional-Integral-Derivative
PLL	Phase Lock Loop
QP	Quadratic Program
RMS	Root Mean Square
RTK	Real Time Kinematics
SMC	Sliding Mode Control
SSF	Sliding surface
STSMC	Super Twisting Sliding Mode Control
TFF	Tight Formation Flight

TRL Technology Readiness Level
UAS Unmanned Aircraft System
UKF Unscented Kalman Filter
UWB Ultra-Wide Band

Reference frame indices

b UAS body frame
e North-East-Down local tangent frame
g Predecessor guidance frame
w Wind frame

List of Figures

Introduction

Background and motivation

Migratory birds routinely perform long range flights, and impressive nonstop distances of more than $8000\,\mathrm{km}$ over a flight time of over nine days are reported [1]. This kind of endurance has so far been elusive for man-made unmanned aircraft of similar size. At the same time, the class of small, low cost unmanned aircraft systems (UAS) enjoys a rising market success as their growing maturity and cost efficiency makes them a viable tool for a range of industries, from crop surveillance in agriculture over short-range cargo delivery to mining exploration.
Commercial small electrically driven fixed wing UAS, however, typically are able to stay in the air for not more than a few hours. Research efforts focusing on increasing propulsion efficiency and improving onboard energy storage density have succeeded in pushing this envelope in prototypes to several days of endurance over the last years by employing liquid hydrogen fuel cell technology [2]. Other approaches rely on exploiting exogenous energy sources by adding solar cell arrays [3] or localizing thermal updrafts [4].
Tight formation flight has attracted attention over the last decades as a complementary exogenous energy source inspired by nature. Many bird species use efficient formation patterns that allow them to recover part of the energy stored in the upwash generated by the birds in front of them and achievable energy savings of the order of 10 % are reported in the literature [5]. As early as 1990 the feasibility of this approach has been

Figure 1: Flock of *grus americana* in close formation flight

demonstrated in humanly piloted flight experiments with manned aircraft [6], showing excellent agreement with theoretical predictions. From the NASA AFF program, about a decade later, maximum fuel savings of 18 % of fuel flow for the follower are reported [7] for a pair of F/A-18 fighter aircraft under automatic control, further confirming the feasibility of tight formation flight (TFF) for significant energy gains and thus range enhancements. Subsequent more recent flight experiments [8] with C-17 transport aircraft suggest average fuel flow savings of 7 to 8 %.

While potential benefits are thus considerably smaller than those achievable by improved energy storage, tight formation flight (TFF) requires only readily available consumer grade commercial-off-the-shelf (COTS) sensors for precise localization and very limited, if at all, airframe modifications. On top of range benefits, the TFF capability is an enabling technology that can unlock a number of other applications with the potential to enlarge the operating envelope of small electrically driven UAS. Aerial recharging and aerial docking are two examples that are not possible today due to a lack of available technologies for safe precise navigation in the close vicinity of another UAS. Position tracking performance requirements of aerial docking obviously depends on the docking mechanism catching the other UAS, but it seems reasonable to assume that they are of the same order of magnitude or stricter than those for upwash exploitation. Any advancement made in the field of small UAS towards safe tight formation flight techniques are of obvious great interest to commercial aviation and could bring an appreciable leap in fuel efficiency to commercial air transport. Small UAS provide in this context an inexpensive testing environment of low complexity that allows to rapidly evaluate the maturity of new techniques.

The benefits of upwash exploitation come with supremely challenging guidance performance requirements. Both theoretical considerations [9] and the flight experiments of [7] suggest that significant upwash gains can only be expected within a small region spanning roughly about 20 % of the wingspan vertically and laterally. Longitudinal separations are less critical due to slow vortex decay. For small UAS this translates into decimeter-level relative position tracking performance and corresponding accuracy in relative localization between UAS.

The resulting small separations between UAS make collision avoidance a major concern. Flight safety has gained even more importance with the ongoing integration of civil UAS into National Airspaces (NAS). Despite recent advances, important elements are missing that we consider essential to make tight formation flight a safe mode of operation. In this work, three principal issues are addressed: position tracking performance, the lack of benchmark problems and guaranteed localization under observation errors.

Open problems and contributions

Open problem I: position tracking performance

Reported experimental results for heterogeneous [10] and homogeneous 2-formations [11, 12] of small UAS feature relative guidance errors of the order of meters. In order to enable meaningful range enhancement for small UAS, significant advances need thus to be made.

The performance gap Considerable interest has appeared in the control community over roughly the last two decades to tackle this challenging control problem. Looking at the landscape of existing research, it becomes apparent that two quite different approaches have been taken for outer loop control (at the inner loop level we find mostly attitude tracking controllers, an exception being the use of Nonlinear Dynamic Inversion (NDI) by the authors of [11]). At one end of the theory-application spectrum, established control techniques such as LQ control [13] or in-flight-tuned proportional-integral-derivative (PID) control [14] have been applied and successfully flight tested (see [11], and [15] for a more recent implementation of the same control laws on an electric fixed wing UAS). As mentioned, the achieved position errors exceed the tolerances necessary for significant wake energy gains.

On the other hand, more advanced techniques such as Higher Order Sliding Mode control (HOSM) have been applied to heavily simplified kinematic representations of the relative position tracking problem in simulation, and very successfully so in terms of relative position errors, see [16, 17, 18, 19, 20, 21]. Sliding mode control exhibits excellent disturbance rejection properties in continuous time for a very specific class of input-affine systems with matched disturbances. Real-world aircraft formation control violates both central assumptions – input switching in continuous time and the matching condition – that form the very basis of SMC techniques. One approach (see e.g. [16]) to circumvent this fundamental issue is to simplify the relative tracking dynamics until the matching condition is satisfied, corresponding to some form of double-integrator dynamics, raising however questions about the applicability of the thus defined control to real fixed-wing aircraft featuring bandwidth-limited rotatory dynamics and actuators.

Contribution I: Predictive sliding mode guidance law

It is considered a prerequisite to successful tight formation flight to close the gap between insufficient position holding performance reported from flight experiments and the superior performance HOSM control can demonstrate on heavily simplified approximations of the relative position tracking problem. In chapter 2 we present Pre-

dictive Discrete Sliding Mode (PDSM) guidance laws that enable position holding performance compatible with the tight requirements of TFF in the vertical channel on a realistic benchmark. We show that, in simulation, PDSM control can provide performance comparable to super twisting sliding mode control for fast sampling and fast inner loops, and vastly superior performance under more realistic conditions. A key ingredient of the control strategy studied in this work is an inner/outer loop interface based on load factor tracking. We provide a prototype LQ vertical load factor tracking control law employing Direct Lift Control (DLC), see section 2.4.4.

Open problem II: benchmark problems

Binary properties of a system under automatic control such as robust stability can usually be evaluated unambiguously for a given system. In contrast to that, control performance is intimately tied to the problem that is used to evaluate a certain control approach. An important obstacle to the systematic and performance-driven advancement of tight formation flight control as of today is the absence of agreed-upon benchmarks that allow to evaluate competing approaches and to make an informed judgment about performance and implementability. Especially for control techniques involving high-frequency control inputs, differences in the chosen modeling of aircraft dynamics and more subtle details such as the choice of simulation sampling times can make the difference between superior control performance and total failure for the same control law, as is elaborated in more detail in chapter 2.

Contribution II: openly available formation flight benchmark

We therefore propose in chapter 1 a benchmark accompanied by an openly available reference implementation. Making available an implementation is expected to significantly lower the threshold towards adopting the benchmark by other researchers, taking into account the corresponding implementation effort of several man-weeks (according to our experience).

Open problem III: guidance safety guarantees

Trajectory planning in the vicinity of other UAS needs to take into account both localization errors and UAS guidance errors to avoid inter-vehicle collisions as well as entering regions in the wake where roll control saturation can occur. The most popular localization filtering strategies based on the Unscented (UKF) [22] or Extended Kalman filter (EKF) [23], provide confidence measures by estimating the state estimate covariances that can be used to approximate performance bounds. Their stochastic nature preclude however that collision avoidance guarantees can be given.

Contribution III: Set membership localization filter and constrained sliding surface

In chapter 3 we describe a set membership localization filter that computes guaranteed bounds on the position of another UAS based on GNSS carrier phase and UWB range observations.
In chapter 2 we exploit the guaranteed relative position sets by designing an integral discrete sliding surface using standard Model Predictive Control (MPC) techniques that enables guaranteed collision avoidance in sliding mode.

List of publications

- J. Bolting, J.-M. Biannic, and F. Defay, "Structured control law design and robustness assessment for the automatic launch of small UAVs," in Proceedings of the CEAS EuroGNC conference, 2015
- J. Bolting, S. Fergani, J.-M. Biannic, F. Defay, and M. Stolle, "Discrete sliding mode control of small UAS in tight formation flight under information constraints," in Proceedings of IFAC Symposium on Automatic Control in Aerospace (ACA) 2016, 2016
- J. Bolting, M. Stolle, S. Fergani, J.-M. Biannic, and F. Defay, "An open benchmark for distributed formation flight control of fixed-wing unmanned aircraft systems," in Proceedings of the 20th IFAC World Congress, 2017
- M. Stolle, J. Bolting, C. Döll, and Y. Watanabe, "A vision-based flight guidance and navigation system for autonomous cross-country soaring UAVs," in Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, pp. 109–117, IEEE, 2015
- M. Stolle, Y. Watanabe, C. Döll, and J. Bolting, "Vision-based lifespan and strength estimation of sub-cumulus thermal updrafts for autonomous soaring," in Unmanned Aircraft Systems (ICUAS), 2016 International Conference on, pp. 162–169, IEEE, 2016
- J. Bolting, J.-M. Biannic, and F. Defay, "Set membership localization for safe tight formation flight," submitted to Robotics and Autonomous Systems

Chapter 1

A formation flight benchmark

Benchmark problems play an essential role in every performance-driven field of research, as they allow to quantitatively compare different approaches and to overcome the problems of solution-specific modeling [1]. As of today, there is no lack of contributions that hold the promise of enabling the performance leap necessary for tight formation flight, a lack of reference problems is however evident. This section presents a benchmark problem for automatic aircraft formation flight [26]. It provides a heavily simplified kinematic model including first-order load factor tracking loop approximations as well as a realistic six degrees of freedom nonlinear model of an electric glider UAS including servo actuator and engine models. A 3D benchmark trajectory and wind and turbulence modeling put guidance laws through a sequence of tracking problems of increasing difficulty, from a straight cruise flight section over climb-descent maneuvers through a level turn into successive helical climb-descent-turn maneuvers. To be useful, a benchmark must be accessible for other researchers, and to this end a Matlab® /Simulink® implementation has been made accessible as an open-source project (Formation Flight Benchmark (FFB) [29]).

1.1 Standard Vehicle Model

Every controller design model inevitably involves simplifications that make it a more or less rough approximation of the actual process to be controlled. The performance of various control techniques applied to the problem of tight formation flight is intimately related to the set of simplifying assumptions adopted for modeling the system dynamics. As an example, excellent performance can be achieved by second order sliding mode control when assuming double integrator dynamics and high controller and

[1] By this we understand choosing a problem that is tailored to a specific solution approach and inherently favors its success. One example is the evaluation of techniques involving high-frequency control inputs such as sliding mode control on plant approximations that lack important dynamics

model sampling rates (see e.g. [30]), i.e. conditions close to the controller design model. Sampling times compatible with avionics hardware constraints and the introduction of inner loop tracking dynamics lead to unacceptable chattering in the input channel and unacceptable position errors, see sc. 2.4.4. To systematically explore these issues, an extension of the Standard Car Model widely used in research on ground vehicle platooning is proposed in this work. It represents the 3D double integrator kinematics of a UAS, further introducing linear first order input dynamics representing load factor tracking loops. Dynamic limitations of the UAS (stall angle, maximum thrust etc.) are represented by input magnitude saturations.
This Standard Vehicle Model (SVM) is of low complexity compared to the more realistic model of a small UAS presented later, but is well suited to systematically study the impact of dynamic effects such as inner loop dynamics that pose serious challenges to the application of SMC techniques and in the development of new approaches to tackle these challenges.
The equations of motion are given in a local North-East-Down tangent frame (index e). We denote at some points this frame as local inertial frame, since, as common in short-term flight simulation, effects of the earths motion w.r.t. inertial space are neglected, as are local variations of gravity.
The continuous-time vehicle position dynamics of vehicle i w.r.t. the local inertial frame follow from Newton's second law to

$$\mathbf{x}_i^e(t) = (\mathbf{p}_i^{e^T}(t)\ \mathbf{v}_i^{e^T}(t))^T \tag{1.1.1}$$

$$\dot{\mathbf{p}}_i^e(t) = \mathbf{v}_i^e(t) \tag{1.1.2}$$

$$\dot{\mathbf{v}}_i^e(t) = \mathbf{g}^e + \mathbf{R}_{\mathbf{eg},i}\mathbf{n}_i^g g \tag{1.1.3}$$

where $\mathbf{p}_i^e(t)$ is the vehicle's position w.r.t. the NED frame, $\mathbf{v}_i^e(t)$ denotes its velocity, $\mathbf{n}_i^g$ are load factors due to aerodynamic forces including thrust acting on vehicle i expressed in the local guidance frame, $\mathbf{R}_{\mathbf{eg},i}$ performs the rotation from that frame to the NED frame, $\mathbf{g}^e$ is the local gravity vector and g the norm of the nominal gravity vector converting load factors into accelerations.

Exogenous disturbances

The load factors acting on the aircraft are assumed to be composed of nominal load factor tracking outputs and unknown but bounded parasitic load factors due to atmospheric turbulence, wake vortices of neighboring UAS and imperfect tracking:

$$\mathbf{n}_i^g = \tilde{\mathbf{n}}_i^g + \mathbf{n}_{w,i}^g \tag{1.1.4}$$

To obtain synthetic load factor disturbances representative of those faced during cruise flight, load factor tracking errors of the Cularis (see section 1.2) baseline inner loop

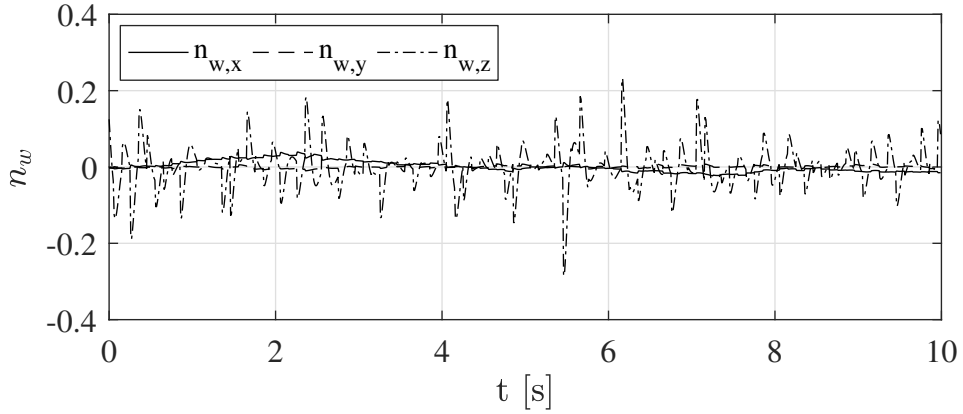

Figure 1.1: Example time series of SVM load factor disturbances

controllers (see section 2.4.4) are recorded during a simulated straight flight in turbulent air under an ambient headwind of 20% of the airspeed. See figure 1.1 for an example time series of the resulting load factor perturbations.

Load factor tracking dynamics

It is assumed that inner load factor controllers track a decoupled first order dynamics reference model

$$\dot{\tilde{\mathbf{n}}}_i^g = \begin{bmatrix} -\tau_x^{-1} & 0 & 0 \\ 0 & -\tau_y^{-1} & 0 \\ 0 & 0 & -\tau_z^{-1} \end{bmatrix} (\tilde{\mathbf{n}}_i^g - \mathrm{sat}(\hat{\mathbf{n}}_i^g(t), \underline{\mathbf{n}}_i^g(t), \overline{\mathbf{n}}_i^g(t))) \tag{1.1.5}$$

where $\hat{\mathbf{n}}_i^g$ are commanded load factors, $\tilde{\mathbf{n}}_i^g$ is the state of the tracking loop, with the corresponding time constants τ_p for $p = x, y, z$ and $\underline{\mathbf{n}}_i^g, \overline{\mathbf{n}}_i^g$ define time-varying box bounds on the admissible load factors.
As will be shown later on, inner loop bandwidth and sampling time of the guidance laws pose the principal issues for continuous time sliding mode control. By varying these parameters, the SVM can cover both the simple double integrator kinematic model in its various forms used in the sliding mode literature (see e.g. [30]) and more realistic configurations. The specific vehicle dynamics are captured by the inner loop load factor controllers. Load factor tracking perturbations and tracking reference saturations $\underline{\mathbf{n}}_i^g(t), \overline{\mathbf{n}}_i^g(t)$ are specific to a given vehicle and mission environment.

1.2 Electric glider UAS model

While the SVM is useful for preliminary studies, a more realistic 6DOF UAS model including rotatory dynamics, engine and actuator models is provided as a second major component of the FFB benchmark and is proposed to evaluate guidance and algorithms

Figure 1.2: Cularis glider (image taken from www.multiplex-rc.de)

either by interfacing to build-in load factor tracking loops (see sec 2.4.4) or directly accessing aircraft engine and servo actuator inputs.

1.2.1 Aerodynamics and actuators

There is an abundance of possibilities to model the aerodynamic coefficients of an aircraft, reaching from look-up tables to neural networks. Following the principle of minimum necessary complexity, here a simple model of the aerodynamic coefficients based on a truncated Taylor expansion and a quadratic drag polar has been adopted. The aerodynamic force and moment coefficients in the wind frame are given by

$$C_D = C_{D_0} + (C_{L,0} + C_{L,\alpha}\alpha)^2 \frac{1}{\pi e \Lambda} + C_{D_{\delta_e}}\delta_e + C_{D_q} q + C_{D_{\delta_f}}\delta_f \tag{1.2.1}$$

$$C_Y = C_{y_\beta}\beta + C_{Y_p} p + C_{Y_r} r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \tag{1.2.2}$$

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_q} q + C_{L_{\delta_e}}\delta_e + C_{L_{\delta_f}}\delta_f \tag{1.2.3}$$

$$C_l = C_{l_\beta}\beta + C_{l_p} p + C_{l_r} r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \tag{1.2.4}$$

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + C_{m_q} q + C_{m_{\delta_e}}\delta_e + C_{m_{\delta_f}}\delta_f \tag{1.2.5}$$

$$C_n = C_{n_\beta}\beta + C_{n_p} p + C_{n_r} r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r \tag{1.2.6}$$

Aerodynamic coefficients are adopted from [31], where the vortex lattice software AVL is used to obtain an aerodynamic model of a small electric glider airplane (Multiplex Cularis, depicted in flight in figure 1.2).

Control surface actuators and engine A second order approximation of a typical miniature servo actuator has been determined in previous work [32] by system identification:

$$\begin{pmatrix} \dot{\delta}_i \\ \ddot{\delta}_i \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{pmatrix} \delta_i \\ \dot{\delta}_i \end{pmatrix} + \begin{bmatrix} 0 \\ \omega_n \end{bmatrix} \delta_{i,c} \tag{1.2.7}$$

with the natural frequency $\omega_n = 62.8\,s^{-1}$ and the damping coefficient $\zeta = 0.8$. This model is augmented here by adding realistic saturations on deflection and deflection rate

$$\delta_i \in [-\frac{\pi}{4}, \frac{\pi}{4}]\,rad \tag{1.2.8}$$

$$\dot{\delta}_i \in [-5.8, 5.8]\frac{rad}{s} \tag{1.2.9}$$

The engine is modeled as first order system

$$\dot{\delta}_{en} = \tau_{en}^{-1}(\delta_{en,c} - \delta_{en}) \tag{1.2.10}$$

where $\tau_{en} = 0.27^{-1}s^{-1}$. The thrust T is then computed using a simple model based on momentum theory (from [33]) as

$$T = \frac{\rho}{2}S_{en}C_{en}((k_{en}\delta_{en})^2 - V_a^2) \tag{1.2.11}$$

with the propeller disc surface S_{en}, the thrust constant C_{en} and the engine constant k_{en}.

1.2.2 Kinematics

Using 1.2.1-1.2.6, the aerodynamic forces and moments in the body frame are computed as

$$\mathbf{F}_a = \mathbf{R_{bw}}\bar{q}S\begin{pmatrix} C_D \\ C_Y \\ C_L \end{pmatrix} + \begin{pmatrix} T \\ 0 \\ 0 \end{pmatrix} \tag{1.2.12}$$

$$\mathbf{M}_a = \mathbf{R_{bw}}\begin{bmatrix} \bar{q}Sb & 0 & 0 \\ 0 & \bar{q}Sc & 0 \\ 0 & 0 & \bar{q}Sb \end{bmatrix}\begin{pmatrix} C_l \\ C_m \\ C_n \end{pmatrix} \tag{1.2.13}$$

with the vector of forces $\mathbf{F}_a$ and the vector of moments $\mathbf{M}_a$, assuming that the thrust vector is aligned with the body frame x axis and coincides with the center of gravity. Translational accelerations in the body frame result to

$$\dot{\mathbf{v}}^b = \frac{1}{m}\mathbf{F_a} - (\omega^b \times \mathbf{v}^b) + \mathbf{R}_{\mathrm{be}}\mathbf{g}^e \tag{1.2.14}$$

where $\mathbf{v}$ is the translational velocity of the body frame w.r.t the inertial frame expressed in body frame axes, and ω^b is the angular velocity of the body frame w.r.t the inertial frame expressed in body frame axes. The rotation rate of the earth can be neglected due to the short duration and range of the considered UAS flight maneuvers (the full benchmark takes about 4 minutes to complete at a nominal trajectory speed of 15 m/s)

and our focus on local relative guidance.
Angular accelerations about body frame axes follow to

$$\dot{\omega}^b = -\mathbf{J}^{-1}(\omega^b \times (\mathbf{J}\,\omega^b)) + \mathbf{J}^{-1}\mathbf{M_a} \tag{1.2.15}$$

with the inertia tensor $\mathbf{J}$. With these accelerations, standard integration techniques and transformations can be used to propagate angular and translational velocities and positions over time to compute the aircraft's full state, which is given by

$$\mathbf{x} = \left(\mathbf{p}^{e^T}\ \mathbf{v}^{e^T}\ \mathbf{q}_{eb}^T\ \omega^{b^T}\right)^T \tag{1.2.16}$$

with NED position $\mathbf{p}^e$, NED velocity $\mathbf{v}^e$, the attitude quaternion $\mathbf{q}_{eb}$ and the vector of rotation rate about body axes ω^b.

1.2.3 Wind disturbance modeling

The UAS is subject to exogenous disturbances due to atmospheric free air turbulence and the wake vortices of surrounding UAS. These are taken into account as additional system inputs, extending the input vector to

$$\mathbf{u} = \left(\delta_e\ \delta_a\ \delta_r\ \delta_{en}\ \delta_f\ \mathbf{v}_w^T\ \omega_w^T\right)^T \tag{1.2.17}$$

with the control surface deflections δ_e (elevator), δ_a (differential aileron), δ_r (rudder), δ_{en} (throttle setting), δ_f (flaps) and the induced wind $\mathbf{v}_w$ and induced angular rates ω_w.

Wind The atmospheric wind is assumed to be a linear combination of a constant ambient part and a time-varying stochastic part, leading to induced velocities in the body frame

$$\mathbf{v}_w^b(t) = \mathbf{R}_{\mathrm{be}}(\mathbf{v}_{w,a}^e + \mathbf{v}_{w,s}^e(t)) \tag{1.2.18}$$

and corresponding perturbations in airspeed V_a and wind angles α, β and induced angular rates $\omega_w^b(t)$. Wind time series are generated according to the Dryden turbulence spectrum [34].

Wake vortex disturbances A variety of approaches has been proposed to approximate the effects of trailing vortices on the following UAS, mostly based on modified Horseshoe Vortex models (HVM) (e.g. [9]), Vortex Lattice methods (VLM) (e.g. [35]), or even, very recently, real-time CFD approaches [36]. For the proposed benchmark, a HVM with modified core model presented in ([37]) is adopted. It is reported to provide

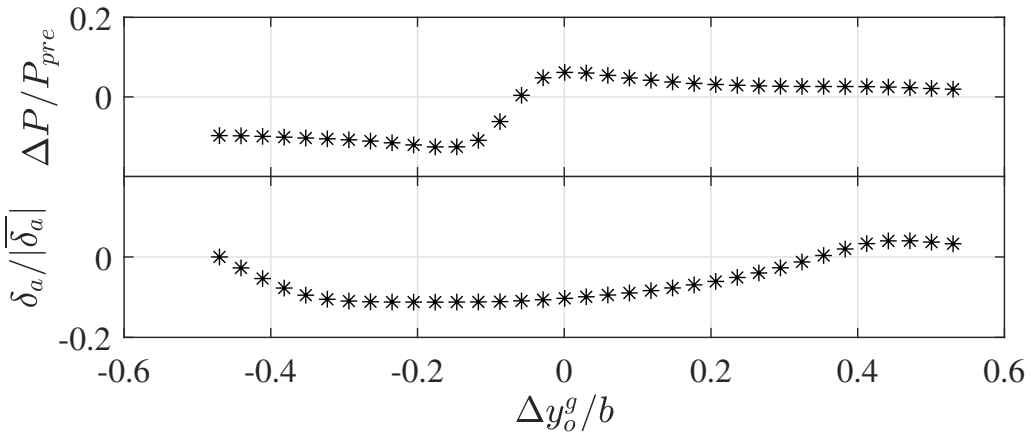

Figure 1.3: Normalized differential thrust and normalized aileron deflection of follower UAS i, $\Delta x = 2b$, $\Delta z = 0$, $V_a = 15\frac{m}{s}$, variation of lateral separation

predictions that are in good agreement with both VLM models and wind tunnel measurements while allowing for reasonable simulation times on a standard PC. From a set of induced velocities at discrete points on the airframe, by using an averaging scheme, an approximate induced airspeed vector and induced angular rates are generated in function of the inter-UAS separation vector, see [37] for details.
In figs. 1.3-1.4 the resulting normalized aileron deflection and normalized propulsion power gain for the Cularis model over a range of lateral and vertical separations are displayed. The normalized propulsion power gain is defined as

$$\frac{\Delta P}{P_j} = \frac{P_j - P_i}{P_j} \tag{1.2.19}$$

where P_i, P_j are the mean engine power in cruise flight of the follower and the predecessor, respectively. The lateral and vertical separations $\Delta y_o^g, \Delta z_o^g$ are deviations from the position of minimum power draw.

Note that for the Cularis UAS model, the model predicts rather modest maximum gains of just over 6 %. This does not come unexpectedly due to the approximate nature of the vortex model and the quite complex aerodynamic interactions considered. Note, however, that the objective of the benchmark is not accurate quantitative prediction, but comparison of competing flight control strategies under identical conditions. To this end, a vortex interaction model that predicts the major trends is considered fully sufficient.
Trim aileron deflections displayed in figures 1.3 and 1.4 suggest that in this particular setting, vortex disturbances are not a major factor for attitude control, since input resources are only reduced by maximally about 10 %.

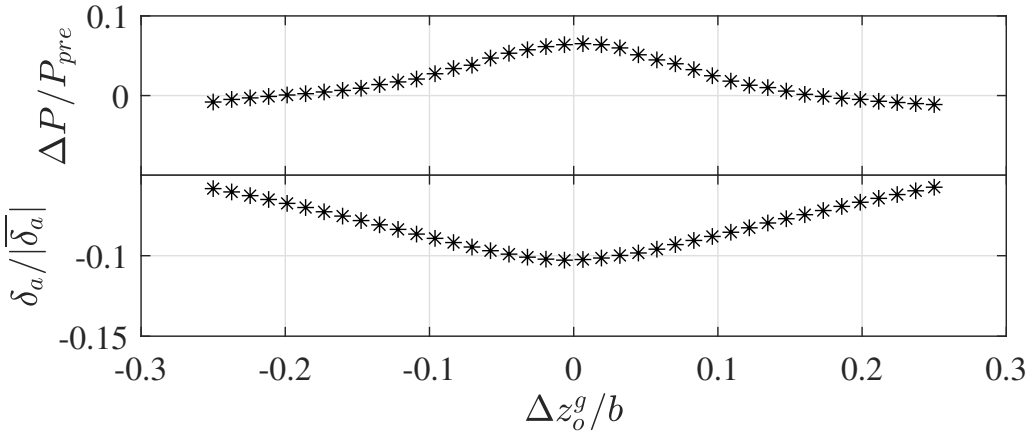

Figure 1.4: Normalized differential thrust and normalized aileron deflection of follower UAS i, $\Delta x = 2b$, $\Delta z = 0$, $V_a = 15\frac{m}{s}$, variation of vertical separation

1.3 Benchmark trajectory

To be feasible, a flight trajectory needs to be consistent with the dynamics of the UAS. Strictly, this implies continuity of flight path derivatives down to servo control inputs, taken even more strictly down to the second derivative of servo inputs, assuming second order servo dynamics. For a fixed wing UAS this translates into C_6 continuity. In reality, C_2 continuity, corresponding to continuous accelerations, is usually sufficient due to very small differences in resulting trajectories when requiring higher order continuity.

The proposed benchmark includes a trajectory composed of cubic Bézier splines, based on a two-step feasible path planning approach, presented by the authors of [38] for autonomous helicopter flight. After planning first a waypoint-based trajectory composed of straight lines, a series of C_2 continuous smoothing splines is computed to obtain a feasible approximation. Maximum curvature is a design parameter of the smoothing splines and allows to adapt it to specific vehicle constraints such as maximum lift. The curvature of the benchmark trajectory is depicted in figure 1.5. The benchmark trajectory consists of a straight part to evaluate cruise flight performance, followed by subsequent climb and descent maneuvers, a $90°$ turn and, inspired by [19, 39], an approximately helical part (see figure 1.6) to evaluate performance during simultaneous climb/descent and turn maneuvers. In the context of upwash exploitation, the cruise flight segment allows to compute useful figures of merit, since typically a major part of UAS long range missions is spent in cruise flight. The maneuvering segments of the trajectory add predecessor accelerations as additional disturbance and are suited for the study of maximum control errors as well as the propagation of maneuvering effects.

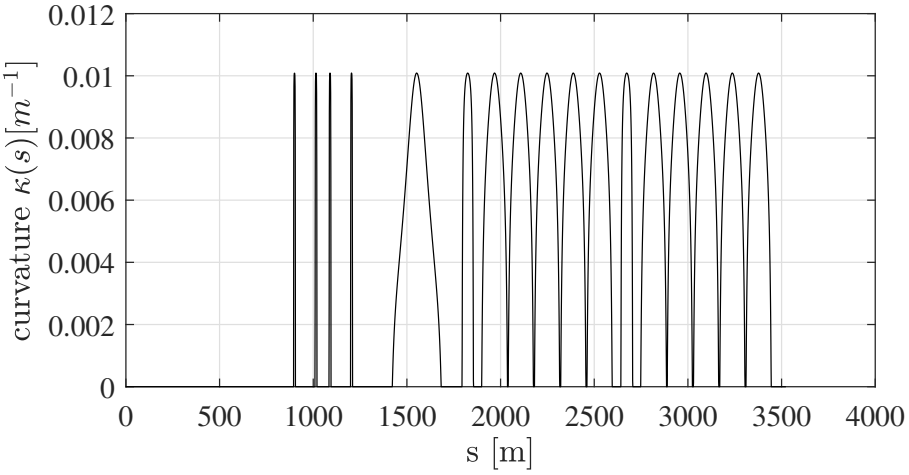

Figure 1.5: Benchmark trajectory, curvature over arc length

1.3.1 Arc length parameterization

Smoothing a waypoint-based trajectory following [38], the resulting curvature-limited sequence of cubic Bézier splines are parameterized with the spline parameter $t \in [0, 1]$. For evaluation of the formation position and velocity over time, an arc length parametrization is more convenient. Using the method proposed by the authors of [40], we first approximate the arc length of each trajectory spline S_i by integration of piecewise linear samples and then fit a piecewise polynomial cubic spline $\tilde{S}_i$ to discrete samples of the spline parameter over the arc length. For a given arc length the corresponding spline parameter is then found by evaluating $\tilde{S}_i$.

1.3.2 Trajectory primitives

Based on the corner smoothing approach [38], a number of trajectory primitives have been implemented that can be chained to form arbitrary maneuver sequences in possible future extensions of the benchmark trajectory. Each trajectory is built up by adding primitives in an incremental manner, i.e. when adding a primitive, the heading of the last leg $\hat{\psi}$ and the end point $\hat{P}$ is modified by an certain increment, avoiding the need for keeping track of frame transformations. In the following we use the rotation matrix between the NED frame and a frame corresponding to an NED frame rotated about the NED z axis by a given heading angle defined by

$$\mathbf{R}_{\text{et}}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1s \end{bmatrix} \tag{1.3.1}$$

Cruise flight The cruise flight primitive adds an additional waypoint to the trajectory in function of a desired incremental trajectory heading $\Delta\psi$ and a desired segment

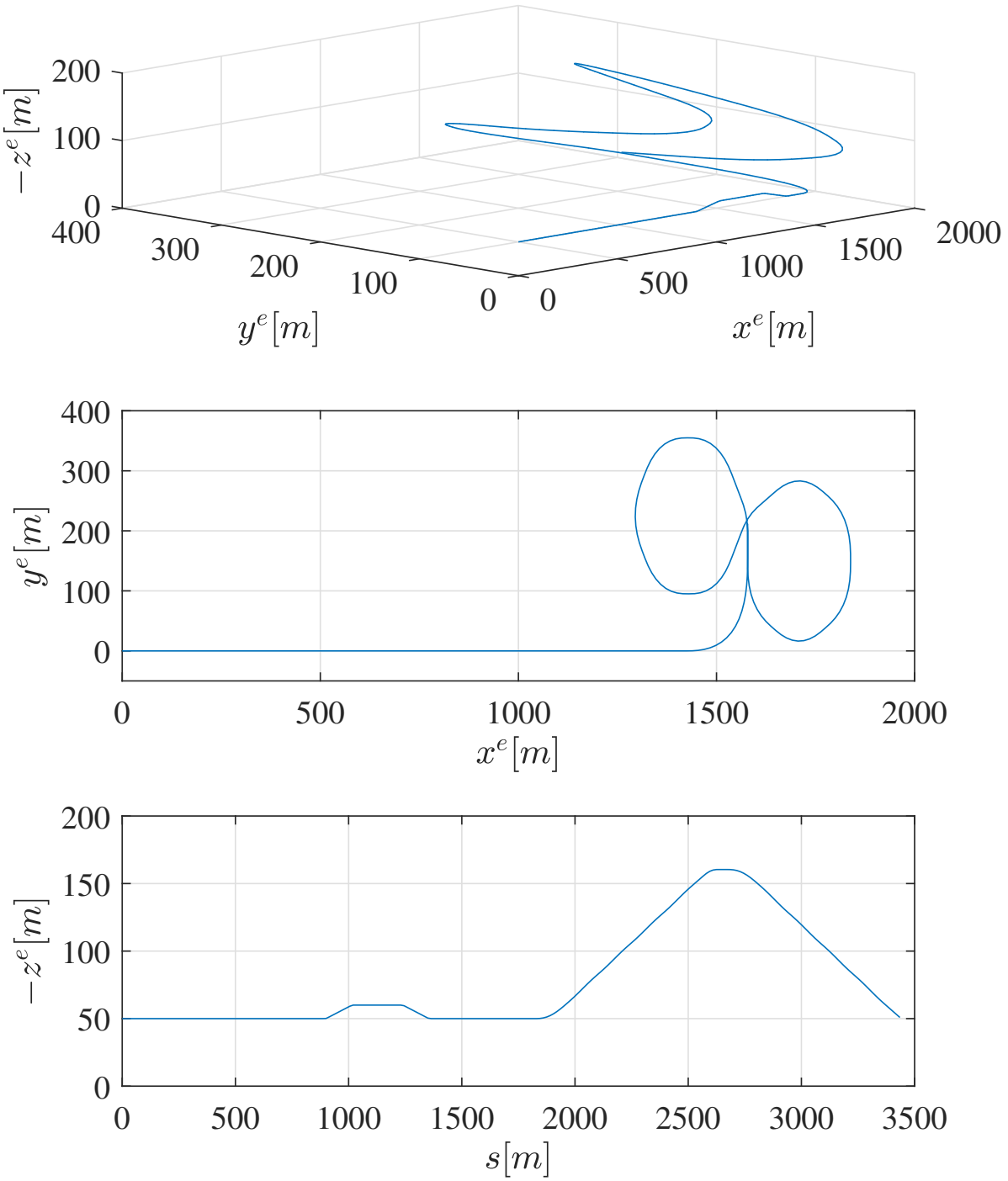

Figure 1.6: Benchmark trajectory: 3D view, projection on x^e, y^e, altitude profile over arc length

length d as

$$\Delta P = \mathbf{R}_{\text{et}}(\hat{\psi} + \Delta\psi) \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix} \tag{1.3.2}$$

Climb-Descent The climb primitive is based on adding a waypoint to the trajectory in function of a desired climb angle γ, altitude increment δh and trajectory heading increment $\Delta\psi$. The waypoint increment results then to

$$\Delta P = \mathbf{R}_{\text{et}}(\hat{\psi} + \Delta\psi) \begin{pmatrix} \frac{\Delta h}{\tan\gamma} \\ 0 \\ -\Delta h \end{pmatrix} \tag{1.3.3}$$

Turn A turn adds two successive waypoint increments according to a desired turn angle $\Delta\psi$ and turn leg distance d as

$$\Delta P_1 = \mathbf{R}_{\text{et}}(\hat{\psi}) \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix}, \ \Delta P_2 = \mathbf{R}_{\text{et}}(\hat{\psi} + \Delta\psi) \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix} \tag{1.3.4}$$

Helix The helix maneuver primitive is parameterized by turn radius r, climb or descent altitude per turn Δh, vertical center unit vector **c** indicating sense of rotation and number of turns n. Due to their non-constant curvature, Bézier splines can only approximate a circular helix. The approximation can be made arbitrarily close to the true helix by increasing the number of waypoints per turn N, with the corresponding increase in memory footprint. Starting from the initial heading $\hat{\psi}$ the initial radius vector and the initial waypoint increment are computed to

$$\mathbf{r}_0 = -r\mathbf{c} \times \mathbf{R}_{\text{et}}(\hat{\psi}) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{1.3.5}$$

$$\Delta P_1 = \mathbf{R}_{\text{et}}(\frac{\pi}{N})\mathbf{r}_0 - \mathbf{r}_0 \tag{1.3.6}$$

and the sequence of waypoint increments is given recursively by

$$\Delta P_i = \mathbf{R}_{\text{et}}(\frac{2\pi}{N})\Delta P_{i-1} + \begin{pmatrix} 0 \\ 0 \\ -\frac{\Delta h}{N} \end{pmatrix} \quad \text{for } \ i = 2...nN \tag{1.3.7}$$

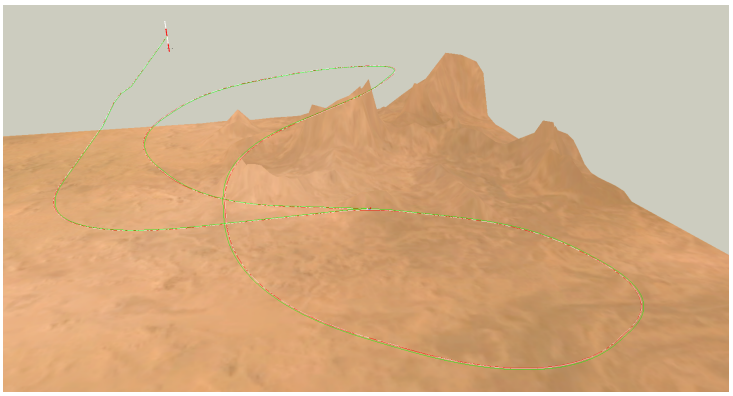

Figure 1.7: Isoview of completed benchmark trajectory in UAV3D

1.4 Visualization

A lightweight visualization environment based on the Java jmonkey game engine [41] has been developed to display the attitude and positions of multiple UAS in a synthetic 3D environment. It has proven to enhance productivity while developing and debugging control laws as well as the UAS simulation model itself. It is interfaced with the Simulink dynamics simulation via a UDP (User Datagram Protocol) network link. Since the visuals have to be updated only at a moderate rate of about 30 Hz for human perception, sending UDP packets only adds a small overhead to the overall simulation time. The tool currently provides several UAS 3D models (glider, electric glider, quadcopter) as well as the possibility to display wake vortex cores.

Computational effort Formations of three UAS have been simulated on an Intel i7 4-core machine close to real time in Simulink® normal simulation mode. Running the benchmark in Simulink® Accelerator mode, where a standalone executable is compiled for the host machine, allows for considerable speedups. It comes however with the downside that control laws have to be implemented in the limited subset of Simulink® /Matlab® compatible with code generation.

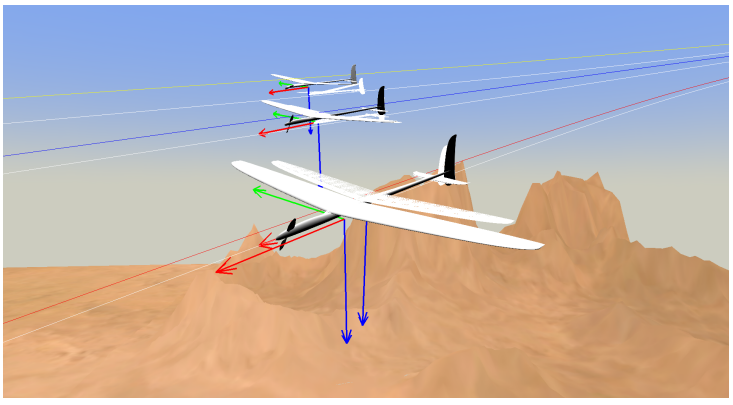

Figure 1.8: N=3 formation in UAV3D

Chapter 2

On guidance and control algorithms for tight formation flight

A variety of guidance architectures has been considered in the formation flight and vehicles platooning community, from fully centralized ones over those employing leader information to ensure string stability, circular architectures and bidirectional to fully decentralized ones relying only on relative states between neighboring vehicles. In this work we consider a unidirectional predecessor-follower configuration, as it minimizes data exchange within the formation and is convenient for visual localization.

2.1 Main objectives

Tight formation flight confronts us with two principal control objectives: avoiding collisions entirely and maximizing the aerodynamic gain from the neighbor's wake. For the first objective, obtaining a useful, i.e. not overly conservative upper bound on the maximum control error is desirable.
For the second objective it is convenient to consider the loss in upwash gains introduced by imperfect tracking of the sweet spot or equivalently the added power consumption. Loss in energy savings correspond to integrated loss in propulsion power savings, which are proportional to relative position tracking errors. The relation can locally be approximated as quadratic (see figs. 1.3,1.4)

$$\Delta P_k \approx (c_y \Delta y_{o,k}^{g^2} + \Delta z_{o,k}^{g^2}) \tag{2.1.1}$$

The added energy consumption for a configuration λ over a flight of N samples with sample time T_s then being approximated by

$$\Delta E_\lambda \approx T_s \sum_{k=1}^{N} \Delta P_{\lambda,k} \tag{2.1.2}$$

we can write the relative efficiency of two configurations λ and κ as

$$\Delta e \approx \frac{T_s \sum_{k=1}^{N} \Delta P_{\lambda,k}}{T_s \sum_{k=1}^{N} \Delta P_{\kappa,k}} \tag{2.1.3}$$

$$= \frac{\frac{1}{N} \sum_{k=1}^{N} (c_y \Delta y_{o,\lambda,k}^{g^2} + c_z \Delta z_{o,\lambda,k}^{g^2})}{\frac{1}{N} \sum_{k=1}^{N} (c_y \Delta y_{o,\kappa,k}^{g^2} + c_z \Delta z_{o,\kappa,k}^{g^2})} \tag{2.1.4}$$

The weighted mean squared (WMS) tracking error

$$e_\lambda(c_y, c_z) = \frac{1}{N} \sum_{k=1}^{N} (c_y \Delta y_{o,k}^{g^2} + c_z \Delta z_{o,k}^{g^2}) \tag{2.1.5}$$

is therefore a simple and meaningful performance index to compare the energy efficiency of different configurations (such as different guidance laws). Recall that in the longitudinal channel, larger errors can be tolerated since the upwash gain is much less sensitive than it is w.r.t. lateral and vertical variations. The predecessor tracking problem we deal with in this chapter is thus the following:

Minimize the weighted sum of squared tracking errors given by relation (2.1.5), while providing guaranteed, i.e. deterministic bounds on the maximum deviation from the nominal relative position.

This tracking problem is generally harder for smaller UAS than larger aircraft as we elaborate on in the next section in some more detail.

Scaling effects affecting small UAS

Tight formation flight demonstrating large upwash gains under automatic control has been demonstrated on a pair of supersonic F/A-18 fighter aircraft in 2001 [7]. Rare demonstrations with small UAS that are close to TFF tracking requirements have been reported more than ten years later [14], raising the question of fundamental obstacles, in more informal terms: if it has been done on large aircraft, why not on small ones? Scale analysis allows some order-of-magnitude insight into how airframe size impacts the TFF problem.

Scaling of sensitivity towards turbulence

Atmospheric turbulence and wake vortices are the predominant source of disturbance in tight formation flight. Disturbances in the vertical translational dynamics due to atmospheric turbulence can be separated into

- disturbances of direction of the airspeed vector, modifying the angle of attack with the corresponding modification of aerodynamic forces and moments
- disturbances of the airspeed norm

Considering the airspeed vector expressed in the aircraft body frame, the angle of attack is given by

$$\alpha = \sin^{-1}\left(\frac{w}{V_a}\right) \tag{2.1.6}$$

where $V_a = \left\|(u\ v\ w)^T\right\|$. Since α is limited to small angles $\alpha < \alpha_{stall}$, equation (2.1.6) can be approximated by

$$\alpha \approx \frac{w}{V_a} \tag{2.1.7}$$

Adopting a truncated first order Taylor expansion of the lift coefficient, the vertical load factor due to lift, using 2.1.7 is approximated by

$$n_z \approx \frac{(C_{L_0} + C_{L_\alpha}\alpha)Sb\frac{\rho}{2}V_a^2}{gm} \tag{2.1.8}$$

$$\approx \frac{(C_{L_0}V_a + C_{L_\alpha}w)Sb\frac{\rho}{2}V_a}{gm} \tag{2.1.9}$$

The partial derivative of 2.1.9 w.r.t. w

$$\frac{\delta n_z}{\delta w} = \frac{C_{L_\alpha}Sb\frac{\rho}{2}V_a}{gm} \tag{2.1.10}$$

and of 2.1.8 w.r.t. V_a

$$\frac{\delta n_z}{\delta V_a} = \frac{C_L Sb\rho V_a}{gm} \tag{2.1.11}$$

give an idea of load factor sensitivity w.r.t. turbulence. Note that for both sensitivities, although the obvious mitigating effect of mass favors larger aircraft, we have the airspeed in the numerator, making the relation not immediately obvious. In planar cruise flight we can compute the lift coefficient irrespective of airframe aerodynamics since

$$C_L = \frac{2mg}{Sb\rho V_a^2} \tag{2.1.12}$$

	Cularis UAS	NASA AFF F/A-18
$b\,[b]$	2.61	11.43
$m\,[kg]$	2	16525
$V_a\,[m/s]$	15	173
$S\,[m^2]$	0.42	37.16
$\rho\,[kg/m^3]$	1.25	0.56
$C_{L_{cruise}}\,[-]$	0.12	0.04
$\frac{\delta n_z}{\delta V_a}\,[s/m]$	0.0667	0.0058

Table 2.1: FFB benchmark and AFF flight condition parameters and load factor sensitivity

simplifying the sensitivity 2.1.11 to

$$\frac{\delta n_z}{\delta V_a} = \frac{1}{V_a} \tag{2.1.13}$$

It is therefore a useful and simple comparative metric.
Typical flight conditions of the small UAS considered in the FFB benchmark (see chapter 1) and the NASA AFF program [7, 42] and the corresponding sensitivities are given in tab. 2.1. Note that for the Cularis UAS, an airspeed perturbation of the same magnitude causes load factor perturbations that are roughly ten times as high as those acting on the F/A-18 aircraft. Similar arguments can be made for the lateral channel and the rotatory dynamics. This snapshot result sheds some light on the relative difficulty of performing TFF with small UAS.

2.2 A brief state of the art

Despite a considerable interest in tight formation flight strategies for small UAS, only a few flight test results are reported so far. This is in contrast to a larger body of published experimental results on automatic TFF with manned aircraft.
After early advancements [13] achieving meter-level formation flight of a heterogeneous pair of two small UAS, incremental improvement could be gained in the following years, see e.g. [11, 15], and a broad variety of guidance schemes tailored to predecessor following has been developed.
In an attempt at automatic aerial drogue docking with two identical small UAS ($b = 2\,m$), the authors of [14] report impressive achieved position errors of $(\Delta x^T_{RMS}\,\Delta y^T_{RMS}\Delta z^T_{RMS})^T = (0.42b\,0.44b\,0.22b)^T$ in cruise flight under an average wind of 30 % of the airspeed, using proportional guidance laws augmented with a feed forward scheme, operating on inertial speed as interface variable. The use of monocular machine vision in combination with single-frequency GNSS makes the presented ap-

proach well suited for low-cost UAS.
Both [13] and [14] choose velocity as interface variable between inner and outer loops In all cases collision avoidance guarantees are not taken into account explicitly and control synthesis involves manual gain tuning. These experimental results suggest that performance close to the requirements of meaningful upwash exploitation and aerial docking is available based on existing guidance and control techniques is within grasp. However, improvement is still necessary, achievable for instance by better system identification and careful control gain tuning.

On the other hand, sliding mode control in theory promises excellent performance and disturbance robustness while requiring little knowledge about the controlled system. Sliding mode control gains can systematically be derived from disturbance bounds. As is well known, adapting sliding mode to the constraints posed by real-world system invariably deteriorates the excellent performance of ideal sliding mode. We consider it worthwhile, as a research effort complementary to established guidance techniques to explore techniques to minimize the extent of this performance deterioration. Excellent position tracking performance could be the reward, as our simulations suggest, see section 2.6. In the following, we first propose a baseline control architecture - adapted to the hardware resources of small UAS - accompanied by baseline LQ guidance and control laws. We then proceed to the design of a novel predictive discrete sliding mode control law in section 2.5.1 that can considerably reduce position tracking errors - find simulation results in section 2.6 - and requires less tuning effort from the control designer than the LQ guidance laws.

Notation and terminology In slight contrast to standard terminology, we denote the separation between UAS along the x axis of the guidance frame as longitudinal, along the y axis as lateral, and along the z axis as vertical separation. This is due to the aforementioned less critical importance of downstream inter-vehicle separation.

2.3 Guidance and control architecture

Maintaining a position in the wake of a preceding aircraft close to that of maximum upwash energy gain constitutes a 4D trajectory tracking problem. Not only a position trajectory needs to be followed (the path tracking problem), but each position needs to be visited at a specific time. By posing the problem in a moving frame attached to the UAS's nominal position, or, equivalently, considering the deviation from a possibly time varying reference position, the problem can, without loss of generality, be reduced to a 3D position regulation problem subject to a number of disturbances the control laws need to reject.
Tracking trajectories within the formation, e.g. during a leader switch in a rotating

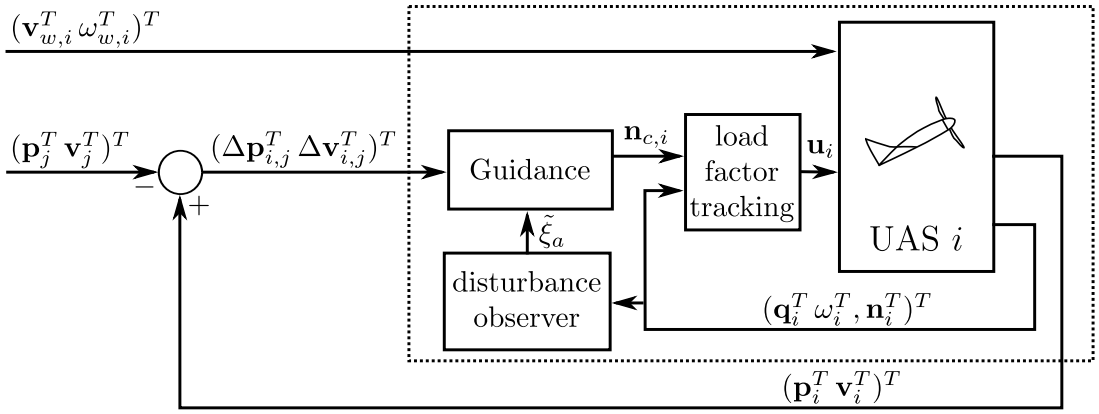

Figure 2.1: Guidance and control architecture of follower i and a predecessor j

leader configuration, can be handled in the same way by treating the change in reference position as further disturbance. We make no assumptions about the orientation of the formation trajectory w.r.t. to gravity other than it is feasible. We make some assumptions on what constitutes the available control inputs of a typical small UAS.

Control resources: assumptions

The minimally available control inputs of a typical fixed-wing UAS configurations are thrust setting and commanded elevator, aileron and sometimes rudder deflection. These inputs naturally limit the control bandwidth (we use the term informally here, since it only applies to linear systems in its strict sense) by which aerodynamic translational forces can be generated and disturbances can be counteracted, since lift is influenced by rotating the airframe about the y axis, and generating side forces involve either banking or yawing. The closed loop rotatory dynamics obviously depend on airframe parameters and available servo actuators, but in very general terms, both longitudinally[1], laterally and vertically control bandwidths of the same order of magnitude are available. Considerable bandwidth in the critical lateral and vertical channel can be gained by adding control surfaces that have a direct impact on aerodynamic lift and side force, rendering the aircraft fully actuated. This approach has been taken by [10]. Side force generators add the benefit of further decoupling lateral and vertical control. There is thus a trade off between available control bandwidth and the design effort of retrofitting direct force generating surfaces. This effort can be considered significant especially in the lateral channel, adding furthermore additional drag, whereas flaps for Direct Lift Control (DLC) can be added comparatively simple. We therefore assume in this work that flap control surfaces are available. Note that some flying wing configurations of low wing sweep do not permit using flaps for DLC.

[1] considering the electric engine model used in the FFB benchmark

Two-layer nested architecture Although by now means imposed by aircraft dynamics it is widely common in flight control to adopt a multiple-loop architecture, subdividing the problem into "guidance" and "control" problems.
Arguments in favor of nested control loops are of practical nature. Both control design and implementation are facilitated, since inner loops can be designed and tested separately. Inner loop control laws can be re-used and outer guidance loops can be developed on simplified models of the inner closed-loop dynamics. Nested architectures provide an abstraction layer bearing strong resemblance to abstraction layers found in software design and share the same benefits, i.e. reducing the complexity of a system to manageable proportions. As a further benefit of multiple loops, guidance laws can be implemented at the typically moderate observation rates of localization sensors, reducing the need for high-rate state estimation. A great many existing UAS trajectory or path tracking control architectures are based on nested control laws, often relying on timescale separation arguments.
As is argued in this chapter, by the right choice of separation into outer and inner loop states, full use can be made of the aforementioned advantages by reducing the inner loop closed loop dynamics to an uncertain, but purely kinematic system. A popular choice of interface variable[2] are Euler angles, since in planar cruise flight the pitch angle Θ is approximately related to the angle of attack α and thus aerodynamic lift; the bank angle is approximately linearly proportional to lateral load factors (see section 2.4.3) for small angles. During trajectory transients, e.g. climb maneuvers, this $\Theta \approx \alpha$ assumption holds less, invariably deteriorating control performance. Tracking angle of attack at the inner loop level circumvents this issue, but requires additional instrumentation. Furthermore, from the perspective of the outer loops, the inner loop appears as time-varying system depending on a number of aerodynamic coefficients such as C_{L_α} of which usually only approximations from system identification are known. Since both inner and outer loops are affected by uncertain aerodynamic coefficients, robust control techniques that can deal with these parametric uncertainties need to be employed on both levels. It would appear desirable to choose a set of interface variables that contain model uncertainty at the inner loop level.
Both load factor and velocity are candidates for such a kinematic inner/outer loop interface. Estimating inertial velocity onboard small UAS relies however exclusively on GNSS Doppler and carrier phase observations. A temporary loss of GNSS observations renders the inner loops nonoperational.
Load factors, on the other hand, can be estimated under loss of GNSS, although with reduced accuracy due to increasing AHRS errors without GNSS.
3D load factors as interface variables, although employed e.g. by [43], appear to have found negligible attention in the flight control community, while providing consider-

[2] By interface variables we mean those aircraft states that are tracked by inner controllers and serve as virtual control inputs for the outer controllers

able advantages that are exploited in this work. Nonlinear dynamic inversion (NDI) based inner loops such as proposed in [11] are a similar approach, making however the strong assumption that aerodynamic forces can be generated instantaneously, i.e. that the vehicle is fully actuated. Load factor tracking, on the other hand does not preclude taking tracking dynamics – due to rotatory and actuator dynamics – explicitly into account. Being able to track load factors allows to base the design of outer laws on a purely kinematic design model. In the ideal case, the dynamics of the vehicle are completely governed by the inner loops. Note that uncertainties of the aerodynamic model of the vehicle affect the outer loops only insofar as they might lead to inner loop tracking errors.
A load factor interface is vehicle agnostic, insofar as it can accommodate fixed wing as well as helicopter and multirotor configurations and could even be used for underwater vehicles. The specificities of the different vehicles manifest themselves in the achievable closed loop load factor tracking dynamics and time-varying constraints on admissible load factors. Guidance laws can be developed on simple kinematic models, provided accurate load factor tracking can be implemented on the concerned vehicle. Furthermore an existing interface based on Euler angles and thrust can be cast as a load factor interface by simple transformations.
Note that the above arguments do not constitute a claim that load factor tracking is an inherently superior choice. We claim however that it is a convenient and elegant choice, largely reiterating arguments given in [43], since it allows to confine aircraft-specific dynamics to the inner loops, thus providing a very useful abstraction layer for guidance loop design.
In the following we consider a two layer guidance and control architecture as depicted in figure 2.1. Its main elements are load factor tracking loops exposing commanded load factors as virtual inputs and a novel predictive discrete sliding mode guidance law complemented by a disturbance estimation scheme. Load factor control as well as guidance laws operate in a local guidance frame defined in the sequel.

2.4 Baseline guidance and control laws

2.4.1 Guidance frame

The system to be controlled is a formation of n UAS flying in an arbitrary pattern. A variety of guidance frames has been proposed in the literature, ranging from a planar frame aligned with the follower's velocity ([11]) to the predecessor's body frame ([44]). The primary objective of tight formation flight for energy saving purposes is to keep each UAS at the position of maximum efficiency in the wake of its predecessor. Being induced by the aerodynamic flow, the wake vortices are approximately aligned with the predecessor's instantaneous wind frame, plus the effect of trajectory curvature. For

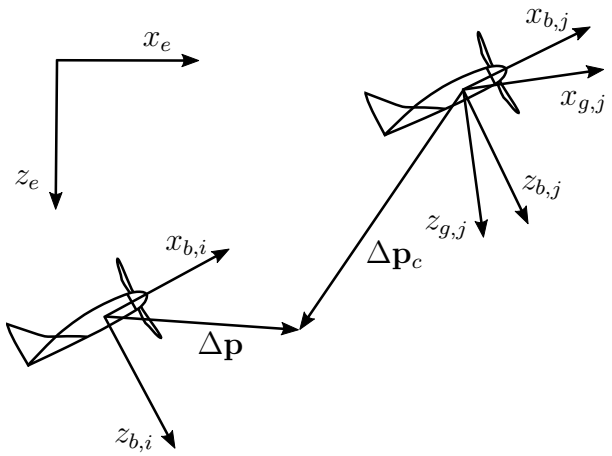

Figure 2.2: Predecessor-follower geometry longitudinal-vertical

maximum energy savings, the follower thus needs to keep its relative position constant in this frame.

Nominal predecessor vortex frame

Since small UAS typically are not equipped with sensors for angle of attack and side slip angle, a local guidance frame (index g) is introduced that approximates the predecessor's vortex frame. Its x axis unit vector $\mathbf{u}_{x,g}$ is aligned with each predecessor's NED velocity vector. Its y axis should be aligned with the y axis of the predecessor's body frame, since the central axis of the vortices lies somewhere on the predecessor's wing. To avoid having to communicate the NED speed and attitude of each UAS to its follower, the nominal speed and acceleration vector of each UAS on the current formation trajectory is used to derive a nominal guidance frame. These frames are computed sequentially for each UAS, since the attitude of each nominal local guidance frame defines the nominal NED position of the next one. For two subsequent UAS, predecessor p and follower f, the nominal NED speed of f is computed as

$$\mathbf{v}_f^e = \mathbf{v}_p^e + \omega_p^e \times \Delta\mathbf{p}_f^e \tag{2.4.1}$$

with the commanded separation vector $\Delta\mathbf{p}_f^e$. Acceleration is given by

$$\dot{\mathbf{v}}_f^e = \dot{\mathbf{v}}_p^e + \omega_p^e \times (\omega_p^e \times \Delta\mathbf{p}^e) \tag{2.4.2}$$

and the instantaneous rotation rate ω_f^e of the guidance frame due to trajectory curvature by

$$|\omega_f^e| = \left(\frac{|\dot{\mathbf{v}}_f^e|}{|\Delta \mathbf{p}^e|} \right)^{\frac{1}{2}} \tag{2.4.3}$$

$$\omega_f^e = |\omega_f^e| \frac{\dot{\mathbf{v}}_f^e \times \mathbf{v}_f^e}{|\dot{\mathbf{v}}_f^e \times \mathbf{v}_f^e|} \tag{2.4.4}$$

The x axis unit vector of the local guidance frame results then to

$$\mathbf{u}_x = |\mathbf{v}_f^e|^{-1} \mathbf{v}_f^e \tag{2.4.5}$$

The orientation of the z axis unit vector $\mathbf{u}_z$ is derived from the simplifying assumption that gravitational acceleration and centrifugal forces due to trajectory curvature are compensated for by the thrust and drag along $\mathbf{x}_g$ and by the aerodynamic lift in a plane normal to $\mathbf{x}_g$. Further assuming that the aerodynamic lift Z lies in the x-z plane of the body frames, first the total centrifugal and gravitational acceleration acting on the aircraft is computed as

$$\mathbf{a}_t = |\omega_p|^2 \mathbf{r} + (0\ 0\ g)^T \tag{2.4.6}$$

The z unit vector $\mathbf{u}_z$ is then computed by projecting and normalizing the total acceleration on the z-y plane of the g frame

$$\mathbf{a}_{t,zy} = \mathbf{a}_t - \frac{\mathbf{a}_t \cdot \mathbf{u}_x}{\mathbf{u}_x \cdot \mathbf{u}_x} \cdot \mathbf{u}_x \tag{2.4.7}$$

$$\mathbf{u}_z = |\mathbf{a}_{t,zy}|^{-1} \mathbf{a}_{t,zy} \tag{2.4.8}$$

and $\mathbf{u}_y$ completes a right-handed Cartesian coordinate frame

$$\mathbf{u}_y = \mathbf{u}_z \times \mathbf{u}_x \tag{2.4.9}$$

leading to the corresponding rotation matrix

$$\mathbf{R_{eg}} = \begin{bmatrix} \mathbf{u}_x & \mathbf{u}_y & \mathbf{u}_z \end{bmatrix} \tag{2.4.10}$$

composed of the unit axes of the guidance frame, see figure 2.2.

2.4.2 Differential dynamics of a pair of vehicles

It complicates modeling, but simplifies control allocation to write vehicle dynamics in the guidance frame. The vehicle position in the guidance frame is given by

$$\mathbf{p}_i^g(t) = \mathbf{R}_{\mathrm{ge}}(t)(\mathbf{p}_i^e - \mathbf{p}_{eg}^e) \tag{2.4.11}$$

where $\mathbf{p}_{eg}^e$ is the position of the g frame w.r.t. the e frame expressed in the e frame. The relative position error of vehicle $i+1$ w.r.t. its predecessor i is then given by

$$\begin{aligned}\Delta\mathbf{p}_i^g(t) &= \mathbf{R}_{\mathrm{ge}}(t)(\mathbf{p}_{i+1}^e - \mathbf{p}_{eg}^e) - \mathbf{R}_{\mathrm{ge}}(t)(\mathbf{p}_i^e - \mathbf{p}_{eg}^e) - \Delta\mathbf{p}_{c,i}^g(t) &(2.4.12)\\ &= \mathbf{R}_{\mathrm{ge}}(t)(\mathbf{p}_{i+1}^e - \mathbf{p}_i^e) - \Delta\mathbf{p}_{c,i}^g(t) &(2.4.13)\end{aligned}$$

with its first derivative

$$\Delta\dot{\mathbf{p}}_i^g(t) = \mathbf{R}_{\mathrm{ge}}(t)(\mathbf{v}_{i+1}^e - \mathbf{v}_i^e) - \Omega^g(t)\Delta\mathbf{p}^g - \Delta\dot{\mathbf{p}}_{c,i}^g(t) \tag{2.4.14}$$

and second derivative

$$\begin{aligned}\Delta\ddot{\mathbf{p}}_i^g(t) =&\mathbf{R}_{\mathrm{ge}}(t)(\dot{\mathbf{v}}_{i+1}^e - \dot{\mathbf{v}}_i^e) + \dot{\mathbf{R}}_{\mathrm{ge}}(t)(\mathbf{v}_{i+1}^e - \mathbf{v}_i^e) - \Omega^g(t)\Delta\dot{\mathbf{p}}^g\\ &- \dot{\Omega}^g(t)\Delta\mathbf{p}^g - \Delta\ddot{\mathbf{p}}_{c,i}^g(t) &(2.4.15)\end{aligned}$$

where $\Delta\mathbf{p}_c^g$ and its first two derivatives define the desired trajectory of the follower in the guidance frame and where Ω^g is the cross product matrix ([45]) of the angular velocity $\omega_g = \begin{pmatrix}\omega_{g,1}\,\omega_{g,2}\,\omega_{g,3}\end{pmatrix}^T$ of the guidance frame w.r.t. the inertial frame

$$\Omega^g = \begin{bmatrix} 0 & -\omega_{g,3} & \omega_{g,2} \\ \omega_{g,3} & 0 & -\omega_{g,1} \\ -\omega_{g,2} & \omega_{g,1} & 0 \end{bmatrix} \tag{2.4.16}$$

and using

$$\dot{\mathbf{R}}_{\mathrm{ge}} = -\Omega^g\mathbf{R}_{\mathrm{ge}} \tag{2.4.17}$$

Introducing

$$\begin{aligned}\Delta\mathbf{v}_i^g(t) =&\mathbf{R}_{\mathrm{ge}}(t)(\mathbf{v}_{i+1}^e - \mathbf{v}_i^e) - \Delta\dot{\mathbf{p}}_{c,i}^g(t) &(2.4.18)\\ \xi_v =&\Omega^g(t)\Delta\mathbf{p}^g &(2.4.19)\\ \xi_a =&\mathbf{R}_{\mathrm{ge}}(t)\dot{\mathbf{v}}_i^e - \Omega^g(t)(\Delta\mathbf{v}_i^g + \Delta\dot{\mathbf{p}}_{c,i}^g(t)) - \Omega^g(t)\Delta\dot{\mathbf{p}}^g - \dot{\Omega}^g(t)\Delta\mathbf{p}^g\\ &- \Delta\ddot{\mathbf{p}}_{c,i}^g(t) + \mathbf{g}^g + g\mathbf{n}_{w,i+1}^g &(2.4.20)\end{aligned}$$

and rearranging, the assembled dynamics in the g frame can be written in a linear form convenient for control design as

$$\begin{aligned} \Delta\dot{\mathbf{p}}_i^g(t) &= \Delta\mathbf{v}_i^g(t) + \xi_v \\ \Delta\dot{\mathbf{v}}_i^g(t) &= g\mathbf{n}_{i+1}^g + \xi_a \end{aligned} \tag{2.4.21}$$

with the control input $\mathbf{n}_{i+1}^g$ and the partly known disturbances ξ_v and ξ_a. The explicit dependence on time is dropped in the following for better readability. Four different sources of disturbances become apparent here:

- disturbances corresponding to inevitable load factor tracking errors of the follower w.r.t. inertial space ($\mathbf{n}_{w,i+1}^g$)
- disturbances corresponding to the rotating guidance frame (i.e. $\dot{\Omega}^g(t)\Delta\mathbf{p}^g$)
- disturbances corresponding to predecessor accelerations ($\dot{\mathbf{v}}_i^e$)
- disturbances corresponding to modifications of the commanded relative position ($\Delta\ddot{\mathbf{p}}_{c,i}^g(t)$)

It is the objective of the guidance laws to track desired positions w.r.t. the predecessor UAS. In the following we first present a set of baseline control laws based on the Linear Quadratic Regulator (LQR) for the system given by equation (2.4.21) and then turn our attention to potentially more powerful sliding mode guidance techniques.

2.4.3 LQ baseline guidance laws

To provide a standard to which to compare new control approaches, we design a set of baseline LQ control laws as part of the FFB benchmark. For controller synthesis, the respective subsystems are extracted from the full linear model. The structure of the control laws is based on the usual assumptions of weak coupling between lateral and longitudinal dynamics and timescale separation of attitude dynamics and translational dynamics.
The synthesis procedure is the same for all four subsystems. It is shortly recalled here, and the specific design systems for every controller are given in the respective subsections. The time-invariant continuous time linear quadratic regulator minimizes the following quadratic performance index

$$J = \int_{t0}^{\infty} (\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u})dt \tag{2.4.22}$$

with the positive semi-definite state cost matrix $\mathbf{Q}$ and the positive definite input cost matrix $\mathbf{R}$ for a given linear system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{2.4.23}$$

To track a given reference state $\mathbf{x}_c$, the tracking error dynamics are derived (dropping in the following the explicit dependence on time) as

$$\dot{\mathbf{x}}_e = \dot{\mathbf{x}} - \dot{\mathbf{x}}_c \tag{2.4.24}$$

$$= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} - \dot{\mathbf{x}}_c \tag{2.4.25}$$

The reference state is either constant, as in the static relative position tracking case, or generated by the outer loop controller (e.g. the commanded bank angle). Its derivative is here simply treated as unknown disturbance and set to zero for controller design, leading to the state error dynamics

$$\dot{\mathbf{x}}_e = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.4.26}$$

that are identical with the state dynamics. Tracking is thus achieved by a simple change of state coordinates

$$\mathbf{x}_e = \mathbf{x} - \mathbf{x}_c \tag{2.4.27}$$

To cope with steady state tracking errors, integral action is added by augmenting the system (2.4.27) with integral states

$$\begin{pmatrix} \dot{\mathbf{x}}_e \\ \dot{\mathbf{x}}_i \end{pmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{x}_e \\ \mathbf{x}_i \end{pmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u} \tag{2.4.28}$$

where G is a sparse matrix defining the states that are selected for integral action. As is common practice, wind disturbances and actuator and engine dynamics are neglected for the LQR design systems. A linear model including models of actuators and engine is used for fast evaluation of candidate controllers during controller tuning. Note that states and inputs are deviations from the operating point where the design model has been obtained by linearization and that for implementation the corresponding operating point trim values need to be added.

Predecessor tracking

Longitudinal The control inputs of the longitudinal position control loop are the commanded longitudinal load factors n_x and n_z. In the absence of external disturbances the integral LQ baseline controllers regulate the relative position and velocity errors

asymptotically to zero. The design system in the guidance frame is given by

$$\begin{pmatrix} \Delta\dot{p}_x^g \\ \Delta\dot{p}_z^g \\ \Delta\dot{v}_x^g \\ \Delta\dot{v}_z^g \\ \Delta\dot{p}_{x,int}^g \\ \Delta\dot{p}_{z,int}^g \end{pmatrix} = \begin{bmatrix} \mathbf{0}_2 & \mathbf{I}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{I}_2 & \mathbf{0}_2 & \mathbf{0}_2 \end{bmatrix} \begin{pmatrix} \Delta p_x^g \\ \Delta p_z^g \\ \Delta v_x^g \\ \Delta v_z^g \\ \Delta p_{x,int}^g \\ \Delta p_{z,int}^g \end{pmatrix} + \begin{bmatrix} \mathbf{0}_2 \\ g\mathbf{I}_2 \\ \mathbf{0}_2 \end{bmatrix} \begin{pmatrix} n_x \\ n_z \end{pmatrix} \tag{2.4.29}$$

where $\mathbf{I}_p$ denotes a unity matrix of size $p \times p$ and $\mathbf{0}_p$ denotes a matrix of zeros of size $p \times p$. Note that virtual accelerations due to the rotating guidance frame are lumped into the unknown disturbances here and neglected in the design model.

Lateral The lateral control strategy is derived from the coordinated turn assumption

$$F_{a,z}^e \approx mg \tag{2.4.30}$$

$$F_a^e \approx \frac{mg}{\cos\Phi} \tag{2.4.31}$$

$$\Rightarrow n_{y,c}^g \approx \frac{\sin\Phi^g}{\cos\Phi} \tag{2.4.32}$$

$$\Rightarrow \ \Phi_c^g = \sin^{-1}(n_y^g \cos\Phi) \tag{2.4.33}$$

thus lateral load factors in the local guidance frame are generated by inclining the approximate lift vector F_a^e via the bank angle w.r.t. the guidance frame, Φ^g. The resulting simple LQ design system is given by

$$\begin{pmatrix} \Delta\dot{p}_y \\ \Delta\dot{v}_y \\ \Delta\dot{p}_{y,int} \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{pmatrix} \Delta p_y \\ \Delta v_y \\ \Delta p_{y,int} \end{pmatrix} + \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} \begin{pmatrix} n_{y,c} \end{pmatrix} \tag{2.4.34}$$

and the commanded bank angle in the local guidance frame is computed using equation (2.4.33).

We proceed to design a set of integral LQ load factor controllers for the vertical and lateral channel.

2.4.4 Some elements of load factor tracking loop design

The first set of controllers tracks pitch and bank angle tracking commands, representing a rather classic approach, and then applies transformations to approximate vertical and lateral load factor tracking. It is part of the benchmark presented in chapter 1. In a

second step, an integral LQ control for vertical load factors is presented based on an augmented state and employing Direct Lift Control for improved tracking performance.

LQ baseline load factor tracking

Commercial as well as open-source UAS autopilots typically provide some kind of basic attitude tracking laws e.g. based on PID control. We keep this assumption for the baseline control laws, emulating the presence of standard inner loop control laws and design a set of LQ attitude tracking laws, but apply transformations to expose a load factor tracking interface to the guidance laws.

Attitude tracking The inner loops consist of separate pitch angle and bank angle tracking laws. Both angles are not the usual Euler angles w.r.t. to the NED frame but are the instantaneous rotations about the y and x axes of the body frame required to rotate the body frame into the desired attitude. Assuming that the derivatives of the commanded attitude are sufficiently bounded, the body frame stays close to its desired attitude and issues arising at Euler angle singularities are avoided. That being said, the commanded attitude rotation matrix w.r.t. to the NED frame is composed as

$$\mathbf{R}_{\mathbf{eb},c} = \mathbf{R}_{\mathbf{eg}}\, \mathbf{R}_{\mathbf{bg},c}^{T}(\Phi_c^g, \Theta_c^g) \tag{2.4.35}$$

where

$$\mathbf{R}_{\mathbf{bg},c}^{T}(\Phi_c, \Theta_c^g) = \begin{bmatrix} \cos\Phi_c^g & \sin\Phi_c^g \cos\Theta_c^g & \sin\Phi_c^g \cos\Theta_c^g \\ -\sin\Phi_c^g & \cos\Phi_c^g \cos\Theta_c^g & \cos\Theta_c^g \sin\Theta_c^g \\ 0 & -\sin\Theta_c^g & \cos\Phi_c^g \end{bmatrix} \tag{2.4.36}$$

Note that the heading angle in the guidance frame is not actively tracked and is set to zero. Virtual angular velocities due to the rotating guidance frame are lumped into the unknown disturbance and neglected in the design models.

Longitudinal The pitch attitude design system is given by

$$\begin{pmatrix} \dot{\Theta}^g \\ \dot{q} \\ \dot{\Theta}_{int}^g \end{pmatrix} = \begin{bmatrix} \mathbf{A}_{lon,att} & 0 \\ \begin{bmatrix} 1 & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{pmatrix} \Theta^g \\ q \\ \Theta_{int}^g \end{pmatrix} + \mathbf{B}_{lon,att} \begin{pmatrix} \delta_e \end{pmatrix} \tag{2.4.37}$$

Vertical load factor tracking Approximate nominal vertical load factor tracking can be achieved by assuming $\Theta^g \approx \alpha$ in level cruise flight and using the nominal lift coeffi-

cient 1.2.3 to compute Θ_c^g, dropping small coefficients to

$$\Theta_c^g = C_{L_\alpha}^{-1}(\frac{n_{z,c}\, g}{\bar{q}Sb} - C_{L_0}) \tag{2.4.38}$$

Longitudinal load factor tracking Approximate load factor tracking in x is obtained by inversion of the engine model 1.2.11, i.e. solving equation (1.2.11) for the engine command $\delta_{en,c}$ given a commanded thrust force $T_c = n_{x,c}gm$.

Lateral load factor tracking The lateral attitude law fulfills two requirements: tracking a commanded local bank angle and adding damping to the weakly damped dutch roll mode. The second objective is achieved by adding the yaw rate r to the state vector of the design system and choosing an appropriate weight in the LQR synthesis matrix $\mathbf{Q}$. The design system is thus given by:

$$\begin{pmatrix} \Delta\dot{\Phi} \\ \dot{p} \\ \dot{r} \\ \Delta\dot{\Phi}_{int} \end{pmatrix} = \begin{bmatrix} \mathbf{A}_{lat,att} & 0 \\ \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{pmatrix} \Delta\Phi \\ p \\ r \\ \Delta\Phi_{int} \end{pmatrix} + \mathbf{B}_{lat,att} \begin{pmatrix} \delta_a \\ \delta_r \end{pmatrix} \tag{2.4.39}$$

Incorporating Direct Lift Control

Direct Lift control (DLC) has been considered as early as the sixties [46] to improve fixed wing aircraft responsiveness in the vertical channel. Used in manned refueling, the authors of [47] report improvements in station holding by flaperons. In [48] Direct Lift Control by dedicated control surfaces is used for improved flight path following. In another very recent contribution [49] an adaptive NDI scheme is successfully employed using DLC to improve altitude tracking of a UAS under the assumption of separate flaps. Using DLC, aerodynamic lift can be influenced with considerably higher bandwidth than through the pitch channel.

The Cularis UAS model that is part of the FFB benchmark features separate flaps for improved glide performance while thermaling and shortened landing distances. Here we exploit these control surfaces for Direct Lift Control. The most simple control structure consists of tracking vertical load factor commands solely by DLC, while stabilizing the short period motion by some auxiliary control. DLC lift resources are however limited due to mechanically limited flap deflections and flap size. The two input channels, pitching – slower, but with maximum lift only limited by the stall angle and available thrust – and DLC surfaces – fast but of limited magnitude – have thus complementary properties when it comes to lift control. As we will see in this section, by augmenting the linearized load factor dynamics with auxiliary states, LQ state feedback control can be designed that exploits these complementary properties in a systematic way via

the usual LQR tuning matrices. Consider the short period dynamics captured by the vertical load factor n_z and the pitch rate q [50]

$$\begin{pmatrix} \dot{n}_z \\ \dot{q} \end{pmatrix} = \mathbf{A} \begin{pmatrix} n_z \\ q \end{pmatrix} + \mathbf{B} \begin{pmatrix} \delta_e \\ \delta_f \end{pmatrix} \tag{2.4.40}$$

$$\begin{pmatrix} n_z \\ q \end{pmatrix} = \mathbf{C} \begin{pmatrix} n_z \\ q \end{pmatrix} + \mathbf{D} \begin{pmatrix} \delta_e \\ \delta_f \end{pmatrix} \tag{2.4.41}$$

where we assume that $\mathbf{C} = \mathbf{I}_2$, i.e. direct observations of load factor and pitch rate are available from the onboard AHRS. Note that the feedthrough matrix $\mathbf{D}$ has nonzero entries, since both control entries generate direct lift. We extend the state vector by defining the input derivatives as new virtual control inputs, rendering $\mathbf{D}$ zero. We further add the integral of n_z to the states for zero steady steady state tracking error.
Furthermore, we would like the control to employ δ_f only for the high-frequency portion of the tracking task, to avoid saturation. This is achieved by including the integral of the flap deflection δ_f as additional state, leading to the augmented LQR design model

$$\begin{pmatrix} \dot{n}_z \\ \dot{q} \\ \dot{\delta}_e \\ \dot{\delta}_f \\ n_z \\ \delta_f \end{pmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \begin{bmatrix} 1 & 0 \end{bmatrix} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \begin{bmatrix} 0 & 1 \end{bmatrix} & \mathbf{0}_{1\times 2} \end{bmatrix} \begin{pmatrix} n_z \\ q \\ \delta_e \\ \delta_f \\ \int n_z \\ \int \delta_f \end{pmatrix} + \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{I}_2 \\ \mathbf{0}_2 \end{bmatrix} \begin{pmatrix} \dot{\delta}_{de} \\ \dot{\delta}_f \end{pmatrix} \tag{2.4.42}$$

The LQR cost-to-go matrices $\mathbf{Q}, \mathbf{R}$ are selected as diagonal matrices. The diagonal entries of the tuning matrices have an intuitive appeal, as most have a direct impact of a specific characteristic of the resulting control, such as an increase in $Q_{2,2}$ increases the short period damping etc.
See figure 2.3 for a tracking example. Note that due to the integral penalization of flap deflections δ_f, flaps are only employed at the beginning of a tracking move and rapidly return to zero, while the lower frequency portion of the load factor is carried by aerodynamic lift due to angle of attack, indicated by non-zero steady state δ_e.

2.4.5 Benchmark performance

When applied to the FFB benchmark, these basic guidance and control laws provide good performance under cruise flight conditions, but insufficient disturbance rejection during predecessor maneuvering, see simulation results in section 2.6.
In what follows, we review the applicability of sliding mode control to the relative guidance problem. We present predictive discrete sliding mode guidance laws that provide improved tracking performance both in cruise flight and during maneuvers.

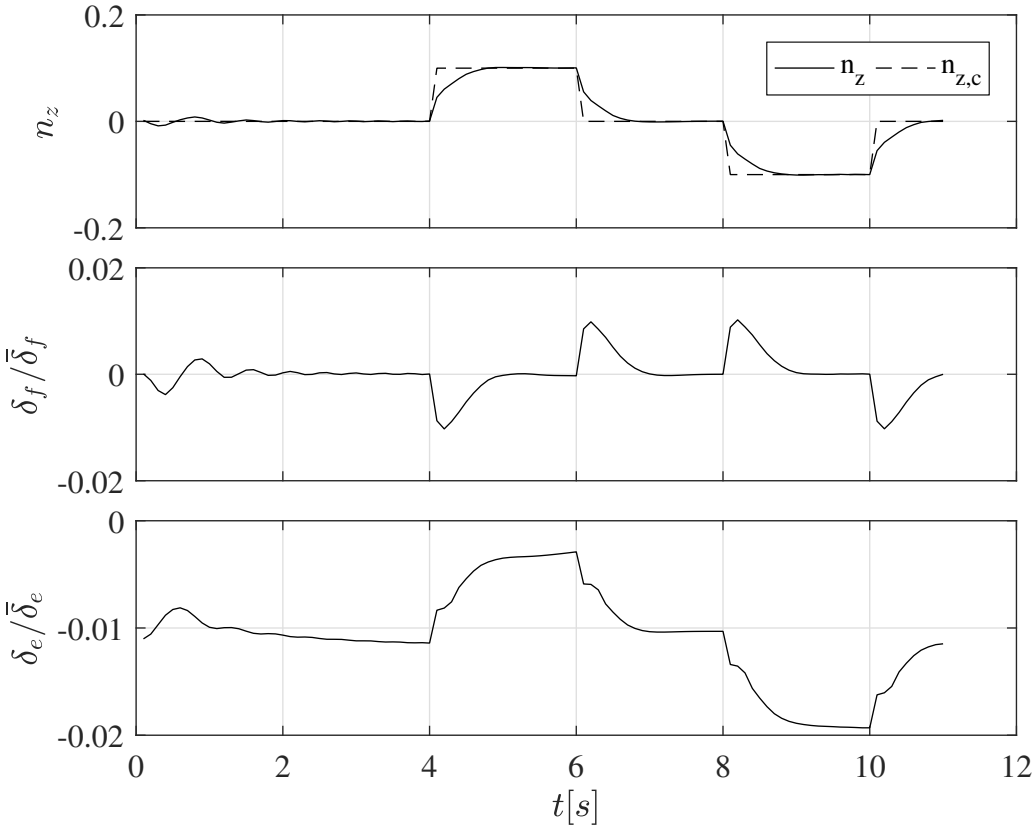

Figure 2.3: n_z tracking example, Cularis UAS model, DLC LQ control

2.5 Predictive discrete sliding mode guidance laws

In most general terms, it is the objective of control to enforce desirable system dynamics. Sliding Mode Control (SMC) approaches this problem by first designing a hypersurface – the sliding surface (SSF) – in state space that represents desirable, stable closed-loop dynamics of the type

$$0 = \sigma(x, x^{(1)}, ...x^{(n)}) \tag{2.5.1}$$

where $x^{(p)} = \frac{d^p x}{dt^p}$ and n is the highest derivative of x involved in the definition of the desired dynamics. This general definition is by now means exhaustive and often output based sliding surfaces are employed. Here we assume that estimates of the relative position and velocity states are available and focus on state-based sliding surfaces. To give a more concrete example, consider a perturbed first order system $\dot{x} = u + w$ and a linear sliding surface

$$0 = \lambda x + \dot{x} \tag{2.5.2}$$

The general disturbances w can lump together nonlinearities, the effects of modeling errors as well as exogenous disturbances. This prescribed behavior forms a hypersurface (in this simple example a line) in state space. In a control synthesis step, a control in-

put u is designed that is guaranteed to drive the system states towards this surface even under worst-case disturbances w (reaching phase), mostly using Lyapunov approaches. Once reached, the control keeps the system states confined to the SSF (sliding phase). While moving along the sliding surface, the closed loop system is indistinguishable from the design system represented by the sliding surface. In our example, the system evolves according to

$$\dot{x} = -\lambda x \tag{2.5.3}$$

which is stable for $\lambda > 0$. While greatly differing in detail, all sliding mode approaches are based on these basic principles.
Note that the underlying idea of SMC is quite different from other control techniques and a quite audacious one: to instantly eliminate disturbances acting on a system by a counteracting control input. In ideal sliding mode, guaranteeing system motion towards the sliding surface at all times implies infinitely fast switching on the sliding surface of a control input of sufficient magnitude to overpower the worst possible disturbance driving the system away from the sliding surface.
It is this worst-case paradigm in concert with the use of continuous-time design models that causes at once the great robustness properties of sliding mode control for certain select classes of dynamic systems with matched[3] uncertainty and potentially very poor performance for many classes of real-world aerospace systems.
For an illustrative example, consider control of the vertical position of an aircraft. Disturbances enter as accelerations. In order to generate counteracting accelerations, control inputs have to rotate the aircraft about the y axis, adding at least two integrators between disturbances and inputs. Even in the presence of DLC devices, actuator dynamics render disturbances unmatched. What is more, guidance laws are implemented at finite discrete sampling rates, invalidating the assumption of instantaneous input switching. Both effects, as is well known, lead to chattering, high frequency oscillations about the sliding surface and in the control inputs.

The approximate nature of both its basic assumptions - matched disturbances and instantaneous switching - on most physical mechanical systems poses the major obstacle to implementation of continuous time sliding mode control and a number of strategies have emerged to improve on this fundamental issue. Boundary layer smoothing approaches are based on replacing the discontinuous control inside a boundary layer around the sliding surface by some linear or nonlinear continuous control law. Continuous approximations of the signum function such as the saturation or the sigmoid function are popular examples of this class of chattering attenuation techniques. While often successfully applied in practice, an important drawback results from the fact that

[3]The matching condition formalizes the property that disturbances enter the system by the same channels as control inputs, in principle enabling the latter to instantaneously eliminate the former

ideal sliding mode is lost even in the disturbance free continuous time case.
A more recent approach is based on including higher-order derivatives of the sliding variable into the feedback. Higher order sliding mode control of order n forces not only the sliding variable, but its time derivatives up to the order of $n-1$ to zero. Input discontinuities appear in the $(n-1)th$ derivative of the control input. The necessity for higher-order derivatives of the sliding variables for feedback purposes, introducing the difficult task of real time signal differentiation, poses the principal implementation issue of HSOM techniques. HOSM observers have been proposed for this purpose, which however remains a challenging task in the presence of discrete sampling and noise. As a notable exception for systems of relative degree one, the Super-Twisting controller (STSMC) does not require derivatives of the sliding variable to enforce second order sliding mode, i.e.

$$\dot{\sigma} = \sigma = 0 \tag{2.5.4}$$

in finite time, making it a very popular choice in SMC applications. It is often proposed as a remedy to recover perfect second order tracking in the ideal case, while providing continuous implementable control input signals.
Discrete Sliding Mode Control (DSMC), treated in more detail in section 2.5.1 approaches the chattering problem from a fundamentally different angle. Instead of designing the control in continuous time and dealing with discretization effects in a second step, a discrete time description of the design system is adopted.
In this chapter we present a brief study of the sensitivity of super twisting control w.r.t. discrete sampling and first order input dynamics. We then address the design of a constrained integral sliding surface that enables collision avoidance guarantees under bounded control errors. We then present a novel predictive sliding mode control approach extending DSMC that approximates the continuous-time performance of classic sliding mode control while exhibiting heavily improved chattering properties in the presence of discrete sampling and input dynamics.

2.5.1 Super twisting guidance laws with discrete sampling and inner loop dynamics

Encouraged by the ability of super twisting control to provide continuous control without requiring the derivative of the sliding variable, the authors of [16, 17, 18, 19, 20, 21] propose the use of STSMC control for aircraft predecessor tracking. Consider the super twisting controller proposed in [30] for tight formation flight. Extending trivially from 2D to 3D and assuming decoupled load factor tracking loops [25], it is given by

$$\hat{n}_p^g = \alpha_p |\sigma_p|^{1/2} \mathrm{sign}(\sigma_p) + \beta_p \int \mathrm{sign}(\sigma_p) dt \tag{2.5.5}$$

where $p = 1...3$ indicates the axes of the guidance frame. We consider the simple LTI sliding surface defined by

$$\sigma(t) = \mathbf{G}\mathbf{x}(t) \tag{2.5.6}$$

$$\mathbf{x}(t) = \begin{pmatrix} \Delta\mathbf{p}^g(t) \\ \Delta\mathbf{v}^g(t) \end{pmatrix} \tag{2.5.7}$$

where $\mathbf{G} \in \mathbb{R}^{3\times 6}$. In the following, for the continuous-time case, the dependence on time is dropped for notational convenience. With

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \mathbf{G}_2 \end{bmatrix} \tag{2.5.8}$$

the position error dynamics in sliding mode ($\sigma = \mathbf{0}$) are

$$\mathbf{0} = \begin{bmatrix} \mathbf{G}_1 \mathbf{G}_2 \end{bmatrix} \begin{pmatrix} \Delta\mathbf{p}^g \\ \Delta\mathbf{v}^g \end{pmatrix} \tag{2.5.9}$$

$$\boldsymbol{\Delta}\mathbf{v}^g = -\mathbf{G}_2^{-1}\mathbf{G}_1\boldsymbol{\Delta}\mathbf{p}^g \tag{2.5.10}$$

Selecting $-\mathbf{G}_2^{-1}\mathbf{G}_1$ as Hurwitz ensures that $\Delta\mathbf{p}$ asymptotically converges to zero while in sliding mode.
From equation (2.4.21) we have the open loop σ dynamics of relative vector degree $\mathbf{r} = (1\,1\,1)^T$

$$\dot{\sigma} = \mathbf{G}\dot{\mathbf{x}} \tag{2.5.11}$$

$$= \mathbf{G} \begin{pmatrix} \Delta\mathbf{v}_i^g + \xi_\mathbf{v} \\ g\hat{\mathbf{n}}_{i+1}^g + \xi_\mathbf{a} \end{pmatrix} \tag{2.5.12}$$

The super twisting algorithm requires bounds on the disturbance magnitude and rate. To simplify defining these bounds in this purely academic setting, we select an inertial guidance frame aligned with the local inertial frame, making $\mathbf{R}_{\mathrm{ge}}(t) = \mathbf{I}_3$ and removing the unmatched disturbance $\xi_\mathbf{v}$, a constant relative position command and no atmospheric perturbations, reducing the remaining disturbance term $\xi_\mathbf{a}$ to

$$\xi_a = \dot{\mathbf{v}}_i^e - \mathbf{g}^g + g\mathbf{n}_{w,i+1}^g \tag{2.5.13}$$

Pre-compensating known terms by the equivalent control

$$\hat{\mathbf{n}}_{i+1,eq}^g = g^{-1}(\mathbf{g}^g - \mathbf{G}_2^{-1}\mathbf{G}_1\Delta\mathbf{v}_i^g) \tag{2.5.14}$$

reduces equation (2.5.12) to

$$\dot{\sigma} = \mathbf{G}_2 g\hat{\mathbf{n}}_{i+1,1}^g - \tilde{\xi}_a \tag{2.5.15}$$

where $\hat{\mathbf{n}}_{i,1}^g$ is the SMC control compensating the unknown disturbances

$$\tilde{\xi}_a = \dot{\mathbf{v}}_i^e + g\mathbf{n}_{w,i+1}^g \tag{2.5.16}$$

We can easily enforce bounds on $\dot{\mathbf{v}}_i^e$ by tracking a virtual predecessor following a sinusoidal trajectory satisfying $\dot{v}_{i,p}^e \leq 1$ for $p = 1, 2, 3$.
In the absence of atmospheric perturbations, load factor tracking errors captured by the second term are solely due to unknown input dynamics. Assuming that $\mathbf{n}_{i+1}^g$ is the output of a stable LTI system of order one, representing well behaved inner loop load factor tracking dynamics

$$\dot{\mathbf{n}}_{i+1}^g(t) = \tau^{-1}(\hat{\mathbf{n}}_{i+1}^g(t) - \mathbf{n}_{i+1}^g(t)) \tag{2.5.17}$$

introduces the obvious issue that equation (2.5.17) breaks the matching condition on top of which classic SMC stability and performance proofs are built. When trying to model this dynamics as feedthrough with additive disturbances, consider the system in the Laplace domain

$$\mathbf{n}_{i+1}^g(s) = \frac{\hat{\mathbf{n}}_{i+1}^g(s)}{1 + \tau s} \tag{2.5.18}$$

The additive disturbance is

$$\mathbf{n}_{w,i+1}^g(s) = 1 - \frac{\hat{\mathbf{n}}_{i+1}^g(s)}{1 + \tau s} \tag{2.5.19}$$

It follows immediately that for high input frequencies ($s = j\omega \to \infty$), the disturbance $\mathbf{n}_{w,i+1}^g(s)$ approaches the commanded input $\hat{\mathbf{n}}_{i+1}^g(s)$, reducing the input margin available for disturbance rejection to zero. Existing proposed HOSM tight formation flight laws sometimes circumvent this fundamental issue by neglecting the rotatory UAS dynamics altogether (e.g. [16]).

Illustration: Standard Vehicle Model closed loop response

To demonstrate the effects of neglecting inner loop dynamics and discrete implementation in the control design model, we apply 2.5.5 to a pair of two SVM models. Control gains are computed according to [51]. For implementation, the STCSMC is sampled with a zero-order hold scheme, leading to the Time-sampled Super Twisting (TSST) controller . We evaluate the two performance indices - mean squared error and maximum absolute error - over the cruise flight flight segment of the FFB benchmark mission. Note that since only the vertical channel is considered, the corresponding weighting coefficient of the weighted mean squared error (see equation (2.1.5)) can be set to one. The FFB standard vehicle model (see chapter 1.1) is considered, and simulations are run over a sweep of two parameters: inner loop time constant $\tau \in [\underline{\tau}, \overline{\tau}] = [0.01, 0.5]$

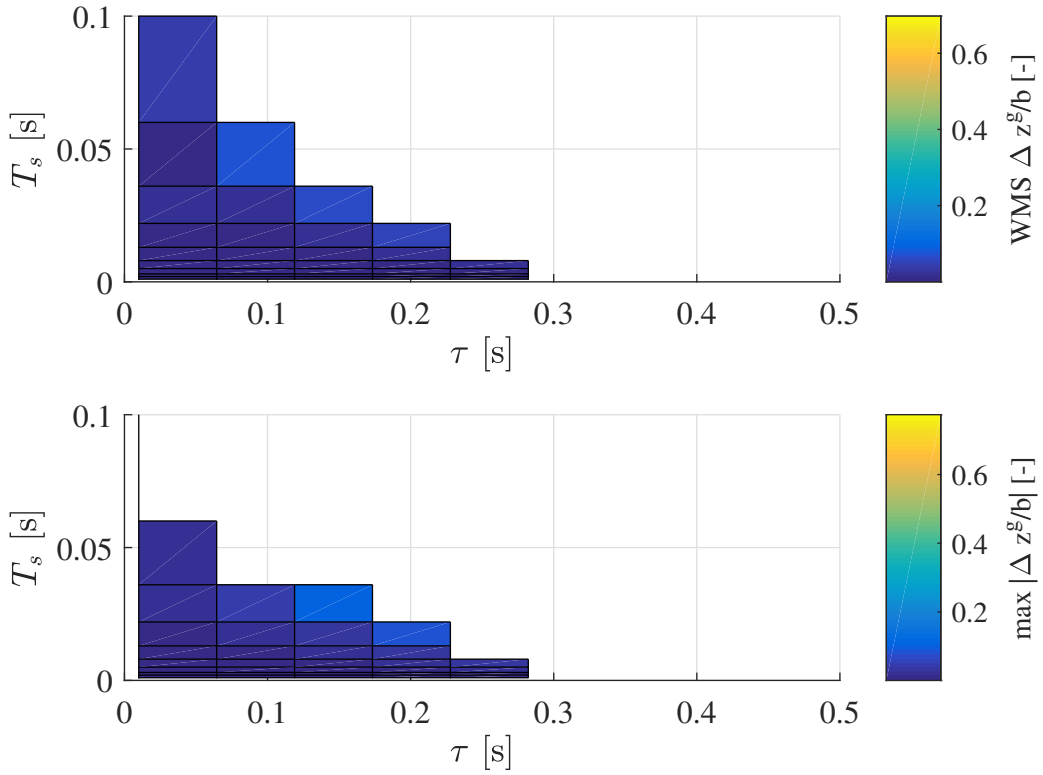

Figure 2.4: Mean squared error and maximum vertical position error under sampled super twisting control over envelope of controller sampling time and inner loop tracking time constant, SVM cruise flight scenario with periodic predecessor motion. White areas mark performance indices exceeding one wingspan.

and controller sampling time $T_s \in [\underline{T}, \overline{T}] = [10^{-3}, 10^{-1}]$. A sinusoidal acceleration of $0.1g$ acts on the predecessor. As seen in figure 2.4, good tracking performance can be achieved for conditions close to the design model. Performance is however very sensitive w.r.t. both parameters and exceeds the defined acceptable interval for a large part of the parameter space. At moderate sampling rates and with realistic inner loop constants, heavy chattering occurs, see time series of normalized position error and the sliding variable in figure 2.5. The depicted four cases represent the corner points of the sampling time and inner loop time constant envelope.
Note that these simulation results have to be considered with caution quantitatively due to the severely simplified nature of the SVM, but qualitatively demonstrate the rapid deterioration of TSST performance in the presence of moderate sampling times and moderately fast inner loop dynamics.

Discontinuity in inner loops

A further, more fundamental limitation arises from the discontinuous first derivative of STSMC control that acts as tracking reference for the inner loops. Consider a load factor tracking law (without direct lift control) with the pitching moment coefficient

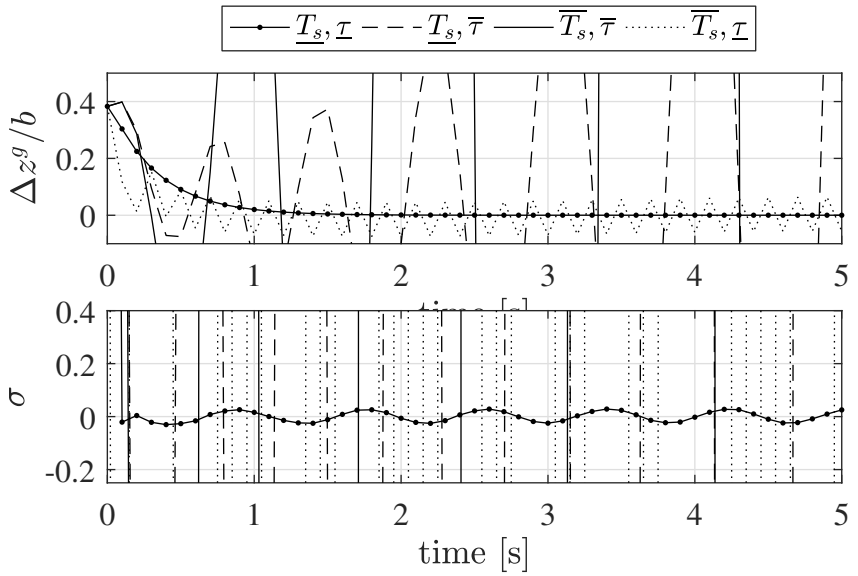

Figure 2.5: Tracking error and sliding variable under sampled super twisting control for corner points of sampling time and inner loop time constant envelope, SVM cruise flight scenario with periodic predecessor motion

about the center of gravity given by [45]

$$C_{m_{cg}} = C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}q + C_{m_{\delta_e}}\delta_e \tag{2.5.20}$$

and assume a hypothetical equivalent control enforcing sliding mode pitch angle command tracking and an LTI first order sliding surface. For clarity of exposition, level flight in calm air and thus $\dot{\theta} = q$ and $\alpha = \theta$ is assumed. The sliding surface with time constant τ_θ is given by

$$q = -\tau_\theta^{-1}(\theta - \theta_c) \tag{2.5.21}$$

$$\sigma = -\tau_\theta^{-1}(\theta - \theta_c) - q \tag{2.5.22}$$

Computing the equivalent control from $\dot{\sigma} = 0$, one obtains:

$$0 = -\tau_\theta^{-1}(q - \dot{\theta}_c) - \dot{q} \tag{2.5.23}$$

$$= -\tau_\theta^{-1}(q - \dot{\theta}_c) - \frac{\bar{q}Sc}{J_{yy}}(C_{m_0} + C_{m_\alpha}\theta + \frac{\bar{c}}{2V_a}C_{m_q}q + C_{m_{\delta_e}}\delta_e) \tag{2.5.24}$$

$$\delta_e = \left(\frac{C_{m_{\delta_e}}\bar{q}Sc}{J_{yy}}\right)^{-1}\left[-\tau_\theta^{-1}(\dot{\theta}_c - q) + \frac{\bar{q}Sc}{J_{yy}}(C_{m_0} + C_{m_\alpha}\theta + \frac{\bar{c}}{2V_a}C_{m_q}q)\right] \tag{2.5.25}$$

Note that the equivalent control is not only inversely proportional to the desired time constant, it is also proportional to the commanded pitch rates $\dot{\theta}_c$. Thus whether magnitude constraints of the elevator actuator are violated depends on the pitch rate reference trajectory, which can be considered as a disturbance in equation (2.5.25). Further dif-

ferentiating, we obtain

$$\dot{\delta}_e = \left(\frac{C_{m_{\delta_e}}\bar{q}Sc}{J_{yy}}\right)^{-1}\left[-\tau_\theta^{-1}(\ddot{\theta}_c - \dot{q}) + \frac{\bar{q}Sc}{J_{yy}}(C_{m_0} + C_{m_\alpha}\dot{\theta} + \frac{\bar{c}}{2V_a}C_{m_q}\dot{q})\right] \tag{2.5.26}$$

Note that to guarantee bounded actuator input rates $\dot{\delta}_e$, the second derivative of the commanded pitch angle, $\ddot{\theta}_c$ needs to be bounded. Applying super twisting sliding mode control, the first derivative of the control input is discontinuous, making the second derivative unbounded. These unbounded signals that appear as virtual inputs propagate to actual actuator commands constitute a further, more fundamental limitation of applying STSMC at the guidance level.

In the sequel we study the application of discrete sliding mode approaches to extend the performance of TSST control at high sampling rates and fast inner loops towards a larger envelope that covers regions compatible with the avionics and inner loop dynamics of more realistic fixed-wing UAS models such as the FFB Cularis UAS (see section 1.2).

Predictive discrete sliding mode control

The detrimental effects of discrete sampling have given rise to discrete variants of continuous time sliding mode techniques.

Discrete Sliding Mode Control

Discrete Sliding Mode Control (DSMC) techniques such as proposed in [52] approach the chattering issue of continuous time SMC by employing discrete-time Lyapunov functions (DTLF) and a discrete-time control design model. The condition on negative definiteness of the time derivative of a continuous time Lyapunov function

$$\dot{V}(\sigma) < 0 \tag{2.5.27}$$

is replaced by a condition on the DTLF decreasing over two successive samples

$$V(\sigma_{k+1}) < V(\sigma_k) \tag{2.5.28}$$

A desired convergence rate can be enforced by requiring

$$V(\sigma_{k+1}) = \Psi V(\sigma_k) \tag{2.5.29}$$

with a $\Psi \in (0, 1)$. The choice of Ψ allows to trade off control effort against reaching time and boundary layer size. DSMC control of this type has been mostly applied to electromechanical systems. See for instance experimental results reported in [53], confirming the expected chattering alleviation in a hydraulic piston position control appli-

cation. Intuitively, the advantages of DSMC when it comes to chattering are due to its being based on a control objective that is closer to our actual objective, which is confining the system states to the sliding surface. Instead of requiring that the derivative of the sliding variable points towards the sliding surface, we require that the state gets closer to it at each sampling instant. This predictive nature allows us to avoid chattering, which is nothing else than a local overshoot over the sliding surface. There are strong parallels between Model Predictive Control and DSMC control and as we will see a DSMC control of the type given by equation (2.5.29) can be cast as a special kind of MPC problem. To the best of the author's knowledge, discrete sliding mode control techniques have not been applied to the problem of aircraft formation flight so far.

The promising properties of DSMC come with a price. To perform the one-step-ahead prediction, a model is required that at least locally well approximates the dynamics of the sliding variable. That being said, as a particular disadvantage, the input gain has to be known, while the continuous time super twisting algorithm only requires its sign and bounds. This property precludes the use of DSMC with inner loops interfaces based on airframe orientation or angle of attack. The particular choice of a load factor interface we make in this work however circumvents this issue, opening the way to apply DSMC techniques.
While heavily mitigating chattering due to discrete sampling, inner loop dynamics negatively affect classic DSMC control by introducing uncertainty into the prediction implicit in the reaching law (2.5.29). In this work we present a extended discrete SMC control scheme that mitigates this chattering effect of inner loop dynamics. We find that established DSMC control laws can be written as equivalent model predictive control problem. This allows us to extend existing algorithms to integrate inner loops states in the prediction step that is solved by established MPC techniques. This also enables us to readily take into account on-line magnitude and rate constraints of inner loops, a further benefit over existing techniques.
In the following, we first derive the discretized predecessor-follower dynamics. We then present an optimization-based procedure for the design of discrete-time sliding surfaces consistent with the constraints of tight formation flight. We then present a predictive discrete-time sliding mode (PDSCM) guidance law and a discrete-time disturbance estimation scheme. Finally, simulations on the FFB benchmark show improved position tracking performance compared to LQ baseline guidance laws.

2.5.2 Discrete predecessor-follower dynamics

The matching condition in discrete time

In real world physical systems, disturbances are always unmatched, due to the inevitable presence of unknown, possibly very fast high order input dynamics. As one

example, the usually neglected mechanical link between a servo actuator and its corresponding control surface represents a fairly complex, very fast and stable nonlinear system in itself. Only by the specific choice of the design model the matching condition can be satisfied. Forward Euler discretization (as selected for instance in an earlier version [25] of the control presented here, not considering inner loop dynamics) preserves the matching conditions of the continuous time dynamics. Under exact discretization however, inputs that are separated from the disturbances by fast stable input dynamics can become matched due to the decay of these parasitic dynamics within one sampling period. Considering the continuous time dynamics of a pair of UAS as given by equation (2.4.21) and assuming that the inner loop tracking behavior is reasonably well approximated by the first order LTI system given by equation (2.5.17) we have

$$\begin{aligned} \Delta\dot{\mathbf{p}}_i^g &= \Delta\mathbf{v}_i^g + \xi_v \\ \Delta\dot{\mathbf{v}}_i^g &= g\mathbf{n}_{i+1}^g + \xi_a \\ \dot{\mathbf{n}}_{i+1}^g &= \tau(\hat{\mathbf{n}}_{i+1}^g - \mathbf{n}_{i+1}^g) \end{aligned} \tag{2.5.30}$$

For greater clarity and ease of notation, we use in the following a generic state space representation of 2.5.30:

$$\dot{\mathbf{x}} = \bar{\mathbf{A}}\mathbf{x} + \begin{bmatrix} \bar{\mathbf{B}}_u & \bar{\mathbf{B}}_d \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{d} \end{pmatrix} \tag{2.5.31}$$

Discretizing 2.5.31, some kind of assumption has to be made about the behavior of control and disturbance inputs between sampling instants. The zero-order hold assumption, most common in MPC algorithms, assumes that inputs keep a constant value between samples. While this can be enforced for control inputs, disturbances generally evolve in between samples and can furthermore only be estimated with some uncertainty. To capture this additional degree of freedom, we split disturbances into a known (correctly estimated) part, constant over the next sampling period, and a bounded but unknown part

$$\mathbf{d} = \tilde{\mathbf{d}} + \Delta\mathbf{d} \tag{2.5.32}$$

so that

$$\dot{\mathbf{x}} = \bar{\mathbf{A}}\mathbf{x} + \begin{bmatrix} \bar{\mathbf{B}}_u & \bar{\mathbf{B}}_d & \bar{\mathbf{B}}_d \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \tilde{\mathbf{d}} \\ \Delta\mathbf{d} \end{pmatrix} \tag{2.5.33}$$

with

$$\mathbf{x} = \begin{pmatrix} \Delta\mathbf{p}_i^g & \Delta\mathbf{v}_i^g & \mathbf{n}_{i+1}^g \end{pmatrix}^T \tag{2.5.34}$$

$$\mathbf{u} = \hat{\mathbf{n}}_{i+1}^{g} \tag{2.5.35}$$

$$\mathbf{d} = \begin{pmatrix} \xi_v & \xi_a \end{pmatrix}^T \tag{2.5.36}$$

and

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & g\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\tau\mathbf{I}_3 \end{bmatrix} \tag{2.5.37}$$

$$\bar{\mathbf{B}}_u = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \tau\mathbf{I}_3 \end{bmatrix} \tag{2.5.38}$$

$$\bar{\mathbf{B}}_d = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{2.5.39}$$

Note that both disturbances in 2.5.33 are unmatched.
Provided the zero-order hold assumption holds, the exact discrete counterpart of 2.5.31 is given by

$$\mathbf{x}_{k+1} = e^{\bar{\mathbf{A}}T_s}\mathbf{x}_k + e^{\bar{\mathbf{A}}T_s}\int_0^{T_s} e^{-\bar{\mathbf{A}}t}\bar{\mathbf{B}}dt \begin{pmatrix} \mathbf{u}_k \\ \tilde{\mathbf{d}}_k \\ \Delta\mathbf{d}_k \end{pmatrix} \tag{2.5.40}$$

$$= \mathbf{A}\mathbf{x}_k + \mathbf{B}\begin{pmatrix} \mathbf{u}_k \\ \tilde{\mathbf{d}}_k \\ \Delta\mathbf{d}_k \end{pmatrix} \tag{2.5.41}$$

where $\mathbf{A}, \mathbf{B}$ are the discrete state transition matrix and input matrix respectively and T_s is the sampling time.

2.5.3 Constrained discrete integral sliding surface

In quasi sliding mode, the sliding surface with its boundary layer constrains the dynamics of the system states. During the reaching phase, apart from a stability condition that ensures contraction of the sliding variable, no guarantees concerning the evolution of the system state are available. More particularly, additional effort would be required to obtain bounds on the state space region of the system stays within during the reaching phase, i.e. to ensure collision avoidance or envelope protection (e.g. $V_a > V_{a_{min}}$).
Integral sliding mode approaches avoid the reaching phase altogether by designing a dynamic sliding surface that starts at the initial state. This dynamic sliding surface can

be designed by applying any standard control technique such as LQR, H_∞ , pole placement etc. to a nominal design model.
In formation flight, state constraints such as collision avoidance regions and minimum and maximum airspeed need to be respected not only during the reaching phase but over the whole mission time. An example that allows to deal with velocity constraints is the nonlinear sliding surface proposed in [54]. However, the equivalent control necessary to evolve along the sliding surface additionally needs to allow for a sufficient margin to reject future disturbances unknown at the time of surface computation.
Very recently, combinations of MPC and sliding mode control have been proposed that accommodate this set of requirements appearing in many control problems beyond aircraft guidance in a very natural manner. In [55, 56] an MPC control law is augmented with a first order switching SMC term and good performance in the presence of matched disturbances on a test bench is demonstrated while showing the usual input chattering. Here we apply this idea to the problem of tight fixed-wing formation flight, replacing the switched input of classic SMC by a predictive discrete sliding mode input to reduce chattering.

Problem statement

Starting from an initial relative state $\mathbf{x}_0$, we ask for a sliding surface (SSF) that satisfies

$$\sigma(\mathbf{x}_0) = 0 \tag{2.5.42}$$

i.e. there is no reaching phase, and

$$\mathbf{u}_{eq,k} \in \mathbb{U}_{eq} \quad \forall k \geq 0 \tag{2.5.43}$$

where $\mathbf{u}_{eq,k}$ is the equivalent control, i.e. the commanded load factors required to have the nominal, disturbance-free system follow the SSF, $\mathbb{U}_{eq}$ is an appropriately tightened load factor constraint set that leaves sufficient margins to reject future disturbances while following the sliding surface. Note that the sequence $\mathbf{u}_{eq,0}...\mathbf{u}_{eq,N-1}$ directly provides the equivalent control necessary to follow the SSF in the nominal disturbance free case. This constraint tightening approach is known from robust MPC, see e.g. [57]. Furthermore we require

$$\mathbf{x}_k^* \in \mathbb{X} \quad \forall k \geq 0 \tag{2.5.44}$$

$$\mathbf{x}_N^* \in \mathbb{X}_N \tag{2.5.45}$$

i.e. the state trajectory $\mathbf{x}_k^*$ representing the integral SSF is required to respect constraints embodied by the state constraint set $\mathbb{X}$ e.g. the velocity must not undercut a given value related to the stall speed and the position must not coincide with the space occupied by other members of the formation. The state constraint set can furthermore be tightened,

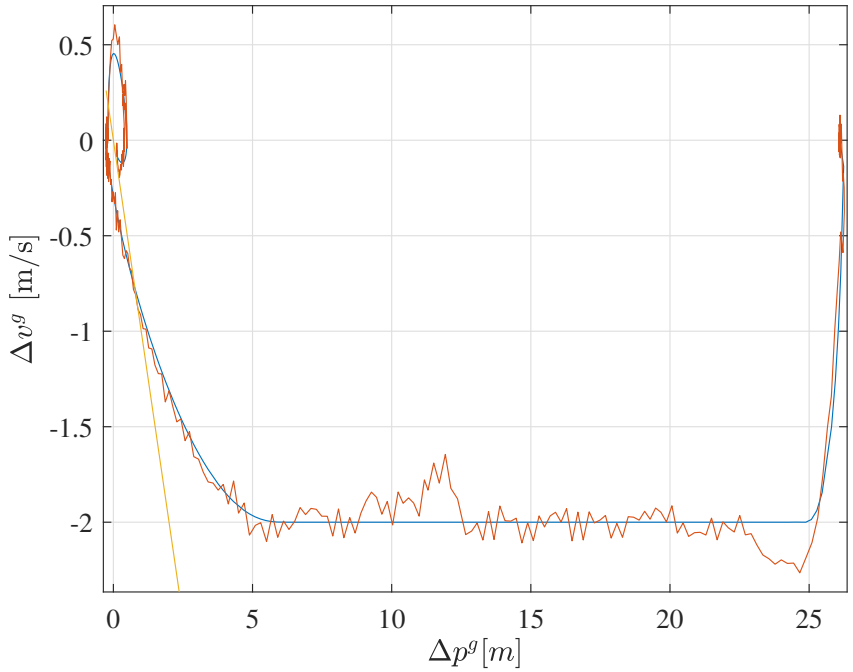

Figure 2.6: FFB benchmark, state convergence phase with large initial relative position error in vertical channel, turbulent air, nominal state constraints $\Delta v_z^g \in [-2, 2]$

taking into account maximum control errors of the sliding mode control that drives the system along the SSF, to guarantee that the original constraint set is not violated under disturbed feedback control. As an example, consider figure 2.6 where the SSF respects constraints on vertical velocity, while a control error of about $0.3m/s$ causes constraint violation in flight in turbulent air. By tightening state constraints, robust constraint compliance can be enforced, see figure 2.7.

Furthermore, the state is supposed to be driven into a neighborhood of the sweet spot in the predecessor's wake, represented by the terminal state constraint set $\mathbb{X}_N$. The size of $\mathbb{X}_N$ depends on the region of attraction of the peak seeking algorithm employed to converge to the position of maximum upwash gain and the amount of uncertainty w.r.t to it.

The problem can be posed as a standard MPC problem with terminal constraint set and prediction horizon N^*

$$\begin{aligned} \min_{\substack{\mathbf{u}_{eq,0}\dots\mathbf{u}_{eq,N^*-1} \\ \mathbf{x}_1^*\dots\mathbf{x}_{N^*}^*}} \quad & \sum_{k=0}^{N^*-1} \mathbf{u}_{eq,k}^T \mathbf{R} \mathbf{u}_{eq,k} + \mathbf{x}^{*T}_{k+1} \mathbf{Q} \mathbf{x}^*_{k+1} \\ \text{subject to } & \mathbf{u}_{eq,k} \in \mathbb{U}_{eq} \\ & \mathbf{x}_k^* \in \mathbb{X} \\ & \mathbf{x}_{N^*}^* \in \mathbb{X}_N \\ & \mathbf{x}_{k+1}^* = \mathbf{A}\mathbf{x}_k^* + \mathbf{B}_u \mathbf{u}_{eq,k} \quad \forall k = 1 \dots N^* \end{aligned} \tag{2.5.46}$$

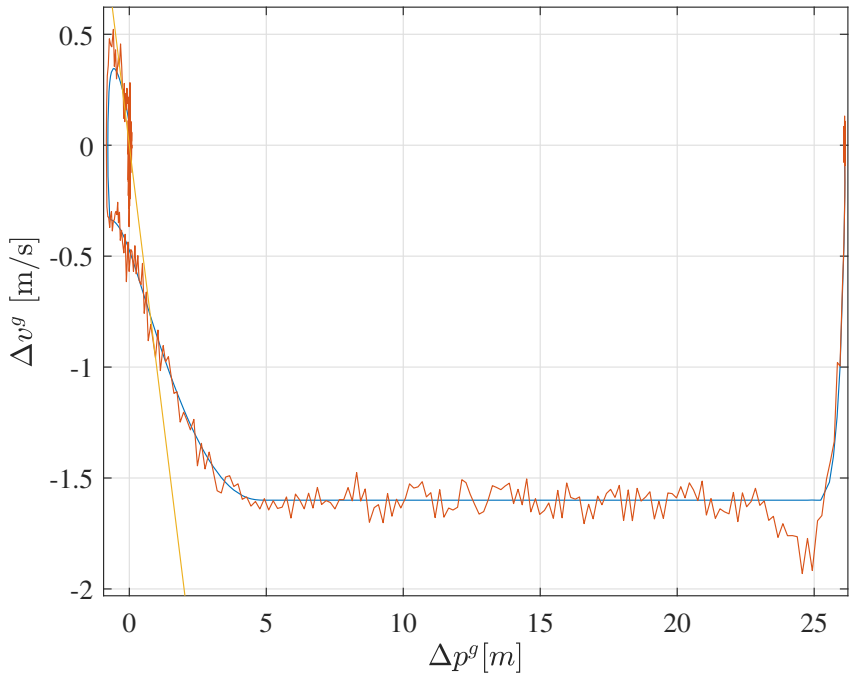

Figure 2.7: FFB benchmark, state convergence phase with large initial relative position error in vertical channel, turbulent air, tightened state constraints

and solved as a quadratic program using fast embeddable QP solvers such as [58].

Horizon length It is desirable to minimize the prediction horizon length for computational reasons. Secondly, while moving along the SSF to the sweet spot, no energy gains can be enjoyed by the follower, adding an additional motivation to find a minimum-time solution. One answer to the minimum time horizon length is given by the notion of reachable sets. If the reachable set corresponding to N contains $\mathbb{X}_N$, $\mathbb{X}_N$ is reachable for an horizon of N. Computing reachable sets is computationally involved even for LTI systems. The minimum horizon length has however to be determined on line, since $\mathbb{X}$ is partly unknown and time-varying due to a-priori unknown positions of other formation members.

In their recent contribution [59], the authors propose finding the minimum horizon N^* for time-optimal model predictive motion control by sequential feasibility checks of the corresponding quadratic program. An implementation example achieving solution update rates of 200 Hz on a standard PC is given, suggesting that this approach is a promising candidate for the minimum time constrained integral sliding surface (CISSF) problem, given the powerful computational resources of today's UAS. It has been implemented in our simulations to identify N^*.

2.5.4 From standard to generalized PDSMC

We proceed now to designing predictive discrete sliding mode laws that have the UAS follow the constrained sliding surface. Employing the CISSF defined in 2.5.3, the open loop sliding variable dynamics follow to

$$\sigma_{k+1} = \mathbf{G}(\mathbf{x}_{k+1} - \mathbf{x}^*_{k+1}) \tag{2.5.47}$$

$$= \mathbf{G}(\mathbf{A}\mathbf{x}_k + \begin{bmatrix}\mathbf{B}_u & \mathbf{B}_d & \mathbf{B}_d\end{bmatrix} \begin{pmatrix}\mathbf{u}_k \\ \tilde{\mathbf{d}}_k \\ \Delta\mathbf{d}_k\end{pmatrix} - \mathbf{A}\mathbf{x}^*_k - \mathbf{B}_u\mathbf{u}_{eq,k}) \tag{2.5.48}$$

Applying the equivalent control sequence resulting from solving 2.5.46

$$\sigma_{k+1} = \mathbf{G}\left(\mathbf{A}\left(\mathbf{x}_k - \mathbf{x}^*_k\right)\begin{bmatrix}\mathbf{B}_u & \mathbf{B}_d & \mathbf{B}_d\end{bmatrix}\begin{pmatrix}\bar{\mathbf{u}}_k \\ \tilde{\mathbf{d}}_k \\ \Delta\mathbf{d}_k\end{pmatrix}\right) \tag{2.5.49}$$

where

$$\bar{\mathbf{u}}_k = \mathbf{u}_k - \mathbf{u}_{eq,k} \in \mathbb{U}' \quad \forall k \geq 0 \tag{2.5.50}$$

is the control available to the sliding mode control for disturbance rejection.

Existing DSMC algorithms such as [52] seek to enforce a prescribed decay rate of σ. Due to the unknown disturbance component $\tilde{\mathbf{d}}$, ideal sliding mode is not achievable for finite sampling rates. It is however possible to drive the system into quasi-sliding mode, defined by

$$|\sigma_k| \leq \epsilon \quad \forall k \geq 0 \tag{2.5.51}$$

and it can be shown that the boundary layer thickness depends quadratically on the sampling time [25, 60] and linearly on the prescribed decay rate. Thus in order to minimize the boundary layer thickness, a small $\mathbf{\Psi}$ is desirable. At the same time, for static sliding surfaces, the initial system state may be far off the sliding surface, limiting the choice of $\mathbf{\Psi}$ to avoid control input saturations.

PDSMC

To meet these conflicting requirements, a time-varying, less restrictive reaching law is proposed in this work, leading to what we denote Predictive Discrete Sliding Mode Control (PDSMC). It is conceptually similar to what the authors of [61, 62] proposed in the context of chemical processing. In contrast to these existing approaches though,

we take hard constraints on input magnitude and rate saturations into account and focus on very short prediction horizons. As the linear reaching laws of type (2.5.29), we enforce a contraction of σ towards the sliding surface at each time step. In contrast to the reaching law (2.5.29), or the optimization based control proposed in [62] the step towards the sliding surface is maximized by solving at each time step the minimization problem

$$\begin{aligned} \min_{\bar{\mathbf{u}}_k} \quad & V(\sigma_{k+1}) \\ \text{subject to} \quad & \bar{\mathbf{u}} \in \mathbb{U}' \subset \mathbb{R}^p \quad \forall k \geq 0 \end{aligned} \tag{2.5.52}$$

Note that this control law has no free tuning parameters that need to be chosen by the control designer but is defined by hard input constraints dictated by the inner loops . It is equivalent to choosing a smaller reaching matrix $\mathbf{\Psi}$ when closer to the sliding surface, leading to tighter bounds on the boundary layer thickness and thus reducing the maximum tracking error. A simple choice for the discrete Lyapunov function $V(\sigma)$ is $V(\sigma) = \frac{1}{2}\sigma^2$.
Inner loop input rate saturations can be taken into account by requiring

$$\bar{\mathbf{u}}_k - \bar{\mathbf{u}}_{k-1} \in \mathbb{U}_\Delta \tag{2.5.53}$$

which can be enforced by setting

$$\bar{\mathbb{U}}_k = \bar{\mathbb{U}} \cap (\bar{\mathbf{u}}_k \oplus \mathbb{U}_\Delta) \tag{2.5.54}$$

where $\oplus$ is the Minkowski sum of two sets given by $A \oplus B = \{\mathbf{a} + \mathbf{b} | \mathbf{A} \in A, \mathbf{b} \in B\}$, i.e. we limit the magnitude constraint set to its part that is consistent with rate constraints. The optimization problem 2.5.52 can efficiently be solved as a small quadratic program, just like linear MPC problems and the same available fast QP solvers can be employed. PDSMC control shares more similarities with MPC and can be written as a special MPC problem. This property is quite remarkable, taking into account PDSMC originally being based on a discrete version of sliding mode control. As a convenient benefit, established ideas and techniques from the field of MPC can readily be exploited in the design of PDSMC control, such as stability constraints.

Stability constraints

Terminal state constraint sets are the dominant way to encode stability into MPC problems, (where usually $N > 1$) either as dual mode MPC or based on contraction. In dual mode MPC the existence of an infinite-horizon unconstrained controller is assumed, e.g. an LQR whose region of attraction (RoA) can be computed. A terminal state constraint set that is an exact representation or outer approximation of this RoA ensures that the last state of the prediction horizon lies within this set. At the end of the prediction horizon the unconstrained controller can thus take over and drive the system

to the origin. In contraction based MPC the terminal state constraint set enforces that a Lyapunov function of the state has decreased. Contraction based stability constraints generally allow for shorter prediction horizons especially when the system is farther away from the origin. The same principle could be applied to PDSMC by including a linear equivalent of the contraction constraint

$$\sigma_{k+1}^2 < \sigma_k^2 \tag{2.5.55}$$

as

$$\sigma_{k+1} \begin{cases} \in [0, \sigma_k), \text{ if } \sigma_k \geq 0 \\ \in (\sigma_k, 0], \text{ if } \sigma_k < 0 \end{cases} \tag{2.5.56}$$

This constraint can be used to check for the feasibility of the optimization problem (OP) 2.5.52. However, if feasible, the constraint 2.5.55 is implicitly satisfied by solving 2.5.52 and does not need to be implicitly included into the QP, reducing its complexity and speeding up solution times.

It is worthwhile at this point to take a step back and put this control as well as classic DSMC approaches of type (2.5.29) into the context of model predictive control to appreciate its particular properties. From an MPC perspective, the PDSMC control solves a state trajectory tracking problem with an extremely short prediction horizon of $N = 1$ (simulations suggest that sampling times of the prediction model of the order of the inner loop time constant provide good performance). This is a particular case of contraction-based MPC, brought forward e.g. in [63, 64]. Contraction based stability in MPC typically allows for much lower prediction horizons than other classic approaches to ensure stability of MPC algorithms, such as dual mode MPC However, minimum prediction horizons for stability reported in the MPC literature typically exceed one.
It is insightful to consider that the controlled system, in spite of the discrete-time prediction model, evolves in continuous time. A minimum horizon required for stability by contraction exist therefore as a minimum prediction time horizon. By choosing a sufficiently large sampling time exceeding this minimum prediction time horizon, contraction over one sampling time step becomes possible. This is an assumption implicitly made in existing DSMC approaches such as [52]. Considering a long sampling time under zero-order input hold is equivalent to an extreme case of what is denoted as "move blocking" in MPC. That being said, posing the PDSMC control problem in an optimization framework also allows for future interesting modifications such as prediction horizons $N > 1$ in conjunction with shorter sampling times. The OP 2.5.52 minimizes the maximum absolute value of the sliding variable at each sampling step, in effect minimizing the boundary layer. By computing an upper bound on σ_{k+1}^2, the maximum boundary layer thickness can readily be obtained.

This is an optimal, i.e. minimum achievable boundary layer thickness under the given input constraints and not a design parameter, in contrast to SMC approaches based on smoothing the CSMC discontinuity by e.g. saturation or sigmoid functions. This sets PDSMC control apart from existing DSMC schemes where the boundary layer is function of the decay rate and as such needs to be chosen by the control designer.
Extending the prediction horizon further than one, the question arises of how to maintain this property. The LQ type costs used in most MPC algorithms form a weighted sum of control errors and as such are not suited. H_∞ MPC control approaches based on Linear Matrix Inequalities have been proposed [65] and adapting this strategy for PDSMC control seems the natural way to approach this problem. However, while mature and fast embedded QP solvers based on automatic C code generation are readily available, this appears not to be the case for LMI solvers [66].

We present in the following a different approach based on contraction constraints that can be solved as a standard QP.

Generalized PDSMC

Consider the following optimization problem $\mathbb{P}_\sigma$ with $\sigma_k = \mathbf{G}\mathbf{x}_k$

$$\begin{aligned}
&\min_{\substack{\bar{\mathbf{u}}_0 \dots \bar{\mathbf{u}}_{N-1} \\ \mathbf{x}_1 \dots \mathbf{x}_N}} \left[\mathbf{x}_1^T (\mathbf{G}^T \mathbf{P} \mathbf{G}) \mathbf{x}_1 + \sum_{k=0}^{N-1} \bar{\mathbf{u}}_k^T \mathbf{R} \bar{\mathbf{u}}_k \right] \\
&\text{subject to } \bar{\mathbf{u}}_k \in \mathbb{U}' \\
&\qquad \mathbf{G}\mathbf{x}_1 \begin{cases} \in [0, \mathbf{G}\mathbf{x}_0), \text{ if } \mathbf{G}\mathbf{x}_0 \geq 0 \\ \in (\mathbf{G}\mathbf{x}_0, 0], \text{ if } \mathbf{G}\mathbf{x}_0 < 0 \end{cases} \\
&\qquad \mathbf{G}\mathbf{x}_k \begin{cases} \in [0, \mathbf{G}\mathbf{x}_1], \text{ if } \mathbf{G}\mathbf{x}_0 \geq 0 \\ \in [\mathbf{G}\mathbf{x}_1, 0], \text{ if } \mathbf{G}\mathbf{x}_0 < 0 \end{cases} \qquad \forall k = 2 \dots N \\
&\qquad \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}_u \bar{\mathbf{u}}_k + \mathbf{B}_d \tilde{\mathbf{d}}_0 \quad \forall k = 0 \dots N-1
\end{aligned}$$

Assuming feasibility, $\mathbb{P}_\sigma$ enforces contraction of the first predicted sliding variable σ_1. It enforces furthermore that the state trajectory does not cross the sliding surface and that all following σ_k do not exceed σ_1, thus in effect minimizing the predicted boundary layer. The novelty of this approach consists in optimizing over the next predicted state, and using this predicted state to constrain the following ones. The matrix $\mathbf{R}$ provides, as in standard MPC problems, a degree of freedom to the control designer to trade off control effort against boundary layer thickness.

Note that the previously considered DSMC reaching law can be stated in the same optimization framework by replacing contraction inequality constraints by equality con-

straints [62]. Various formulations are possible, one example is

$$\begin{aligned} \min_{\mathbf{u}_k, \mathbf{x}_{k+1}} \quad & \bar{\mathbf{u}}_k^T \bar{\mathbf{u}}_k \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}_u \bar{\mathbf{u}}_k + \mathbf{B}_d \tilde{\mathbf{d}}_k \\ & \mathbf{G}\mathbf{x}_{k+1} = \mathbf{\Psi}\mathbf{G}\mathbf{x}_k \end{aligned} \tag{2.5.57}$$

In the following we direct our attention towards the minimum horizon single input PDSMC control 2.5.52 which is a special case of $\mathbb{P}_\sigma$ with $N = 1$, $p = 1$ and $\mathbf{R} = \mathbf{0}$. Due to the very small problem size it can actually easily be solved analytically by what amounts to a minimal version of an active set QP algorithm. Considering

$$\sigma_{k+1} = \mathbf{G}(\mathbf{A}\mathbf{x}_k + \mathbf{B}u_k + \mathbf{B}_d\tilde{\mathbf{d}}_k + \mathbf{B}_d\Delta\mathbf{d}_k) \tag{2.5.58}$$

$$V = \frac{1}{2}\sigma_{k+1}^2 \tag{2.5.59}$$

At an unconstrained minimum we have $\frac{\delta V}{\delta u} = 0$, thus for the nominal control ($\Delta\mathbf{d} = \mathbf{0}$)

$$\frac{\delta V}{\delta u} = (\mathbf{G}(\mathbf{A}\mathbf{x}_k\mathbf{B}_d\tilde{\mathbf{d}}_k) + \mathbf{G}\mathbf{B}\mathbf{u}_k)\mathbf{G}\mathbf{B} \tag{2.5.60}$$

$$= 0 \tag{2.5.61}$$

and the unconstrained PDSMC control is

$$u^* = -(\mathbf{G}\mathbf{B})^{-1}(\mathbf{G}\mathbf{A}\mathbf{x}_k + \mathbf{G}\mathbf{B}_d\tilde{\mathbf{d}}_k) \tag{2.5.62}$$

If the unconstrained control $u^* \in \mathbb{U}'$, it is identical to the constrained control. Otherwise, we compute V at the boundaries of $\mathbb{U}'$ and select the smallest one, e.g. for a box constraint $\mathbb{U}' = [\underline{u}, \bar{u}]$, we obtain two values of $V(\sigma)$ corresponding to applying $\underline{u}$ or $\bar{u}$. Note that, interestingly enough, for a linear sliding surface of type $\sigma = \mathbf{G}\mathbf{x}$ and a linear system, in a neighborhood of the sliding surface where input constraint are inactive, the single input PDSMC control with $N = 1$ results in a linear state feedback plus a disturbance rejection term.
The non-iterative computation of u^* enables fixed-time implementation, an interesting general property for flight control, where computing time guarantees need to be given. We have not commented so far on how to obtain the disturbance estimate $\tilde{\mathbf{d}}_k$. In the following we investigate suitable real-time estimators.

2.5.5 Disturbance estimation

Prediction based PDSMC control builds on the assumption that some kind of estimate of unknown disturbances due to predecessor motion, the rotating guidance frame etc. is available. In this section we present a continuous time sliding mode observer and a

discrete disturbance observer and weigh their respective merits.
Consider the error dynamics of a pair of SVMs

$$\begin{aligned}
\Delta\dot{\mathbf{p}}_i^g(t) =& \Delta\mathbf{v}_i^g(t) + \xi_v && (2.5.63)\\
=& \Delta\mathbf{v}_i^g(t) + \Omega^g(t)\Delta\mathbf{p}^g && (2.5.64)\\
\Delta\dot{\mathbf{v}}_i^g(t) =& g\mathbf{n}_{i+1}^g + \xi_a && (2.5.65)\\
=& g\mathbf{n}_{i+1}^g + \mathbf{R}_{\mathrm{ge}}(t)\dot{\mathbf{v}}_i^e - \Omega^g(t)(\Delta\mathbf{v}_i^g + \Delta\dot{\mathbf{p}}_{c,i}^g(t))\\
& - \Omega^g(t)\Delta\dot{\mathbf{p}}^g - \dot{\Omega}^g(t)\Delta\mathbf{p}^g - \Delta\ddot{\mathbf{p}}_{c,i}^g(t) + \mathbf{g}^g + g\mathbf{n}_{w,i+1}^g && (2.5.66)
\end{aligned}$$

Disturbance elements such as those due to gravity and imperfect load factor tracking acting on each vehicle due to atmospheric turbulence and predecessor wake vortices can in principle be estimated using on board accelerometers. Disturbances due to predecessor motion, require high-rate communication between vehicles.

Super twisting sliding mode disturbance observer

A different approach can be taken by estimating the sum of disturbances with a sliding mode observer (SMO). The problem tackled here is basically a problem of estimating unknown acceleration inputs by real time signal differentiation, in the presence of discrete sampling and measurement noise. Sliding mode observers, just like sliding mode controllers, have very attractive properties in a continuous time framework. SMO techniques have been widely applied to disturbance estimation problems in the context of fault detection and real time signal differentiation. The basic idea is to treat the estimation problem as a control problem and to drive the error between the velocity of a model of a pair of vehicles internal to the observer and the estimated actual relative velocity to zero by means of a virtual control input. Once the error has converged to zero, the virtual control input generated by the SMO is an estimate of the accelerations acting on the system. Consider the vertical channel of 2.5.66

$$\Delta\dot{\hat{v}}_{i,z}^g = a_z \tag{2.5.67}$$

where $\Delta\dot{\hat{v}}_{i,z}^g$ is the relative velocity of the observer model, and a_z is the virtual control input. Defining the sliding variable σ and its first time derivative

$$\begin{aligned}
\sigma &= \Delta v_{i,z}^g - \Delta\hat{v}_{i,z}^g && (2.5.68)\\
\dot{\sigma} &= g n_{i+1,z}^g + \xi_a - a_z && (2.5.69)
\end{aligned}$$

we see that σ is of relative degree one and both classic first order SMC or second or-

der super twisting control can be applied. First order SMC generates high-frequency switching in the virtual control $\tilde{n}$ and low-pass filtering needs to be applied to obtain a smooth estimate, introducing additional lag in the observer response. On the other hand, the super-twisting algorithm readily generates continuous virtual inputs, it is therefore the natural choice

$$a_z = \alpha|\sigma|^{1/2}\text{sign}(\sigma) + \beta \int \text{sign}(\sigma)dt \tag{2.5.70}$$

where α, β have to be tuned according to bounds on $\xi_a, \dot{\xi}_a$, e.g. to satisfy conditions given in [51]. Applying this observer to the relative velocity between two UAS in the guidance frame, we obtain an estimate of its total time derivative. To recover the unknown disturbance, consider equation (2.5.66)

$$\tilde{\xi}_{a_{smo},k} = a_{z,k-1} - g\mathbf{n}^g_{i+1,k-1} \tag{2.5.71}$$

Discrete disturbance observer

The finite difference differential is given by

$$\Delta\dot{\tilde{v}}^g_{i,z,k} = \frac{\Delta v^g_{i,z,k} - \Delta v^g_{i,z,k-1}}{T_s} \tag{2.5.72}$$

and an estimate of the unknown disturbance is recovered by the discrete disturbance observer (DDO) as

$$\tilde{\xi}_{a_{ddo},k} = \Delta\dot{\tilde{v}}^g_{i,z,k} - g\mathbf{n}^g_{i+1,k-1} \tag{2.5.73}$$

The sensitivity of 2.5.72 w.r.t. additive noise w_k as function of sample time can easily be analyzed by considering

$$\Delta\dot{\tilde{v}}^g_{i,z,k} + e = \frac{\Delta v^g_{i,z,k} + w_k - \Delta v^g_{i,z,k-1} - w_{k-1}}{T_s} \tag{2.5.74}$$

$$= \Delta\dot{\tilde{v}}^g_{i,z,k} + \frac{w_k - w_{k-1}}{T_s} \tag{2.5.75}$$

thus the differentiation error due to noise is inversely proportional to the sampling time, confirming intuition, and bounded by $e \in [-2\bar{w}/T_s, 2\bar{w}/T_s]$ with $w \in [-\bar{w}, \bar{w}]$.

Performance under moderate sampling rates

For tight formation flight under GNSS-based relative localization, disturbance observers have to deal with measurement noise and low observation rates of low-cost GNSS receivers (roughly $f_{GNSS} \leq 10\,Hz$ [67]). To evaluate the performance of both observers in

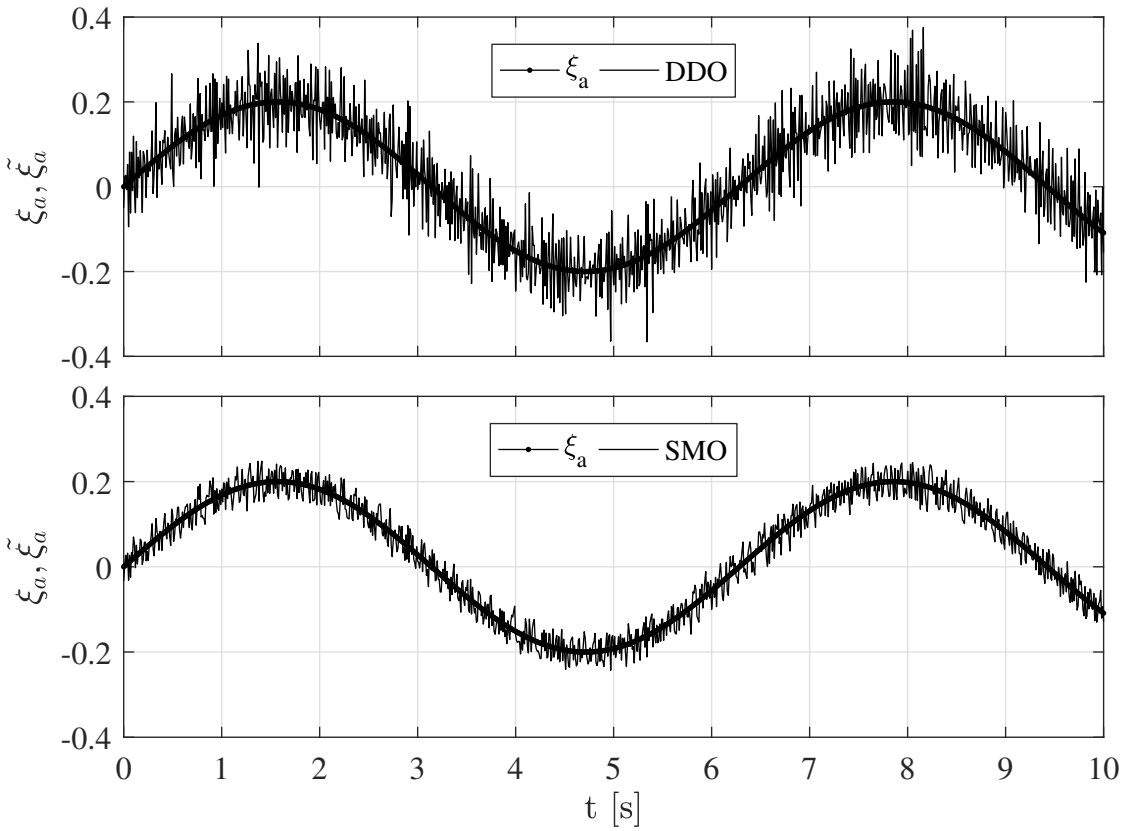

Figure 2.8: Disturbance estimation, $T_s = 10^{-2}s$

simulation, a synthetic disturbance ξ_a is injected. This way magnitude and derivative bounds are perfectly known and the gains of the SMO can be computed without any assumptions. Measurement noise corresponding to estimation quality achievable with GNSS carrier phase time differencing [68] is added to the velocity signal.
See [69] for a recent application of SMO for the piston position control of an electro-pneumatic system. The choice of SMO over finite-difference differentiation is backed in [69] by a simulation example showing smaller errors in the estimated signal derivative in the presence of noise. Our simulations confirm this superiority for fast sampling. Taking differentiator sampling time into account, the picture becomes more involved. While at high sampling rates the SMO handles even noisy velocity observations well, derivation performance deteriorates with lower sampling rates, similar to sampled continuous time sliding mode control performance. See figures 2.8, 2.9 for snapshots of observation performance at sampling rates of $10^{-2}s$ and $10^{-1}s$ respectively. In the first case, closer to its continuous time design conditions, the SMO shows significantly smaller estimation errors.
For a slower sampling rate of $10^{-1}s$, the picture reverses, the disturbance estimate of the SMO is dominated by numerical chattering, while the DDO estimate follows well the true disturbance, with the phase lag inherent to finite-difference differentiation, see figure 2.9. This small case study illustrates how sliding mode observer performance, just like its SMC dual, is tied to its discrete implementation and can be outperformed

by very simple schemes under unfavorable conditions.
See figure 2.10 for a sweep of the rms and maximum absolute error of the SMO and the DDO over a range of sampling rates. The DDO differentiation error in the noisy case evolve as indicated by equation (2.5.75) and converges to the noise-free error for lower sampling rates. For low sampling rates, the error due to phase lag dominates, and the error in case of noisy signal converges to that corresponding to the noise-free case. The sliding mode observer shows superior performance only for fast sampling rates down to about 50 Hz.

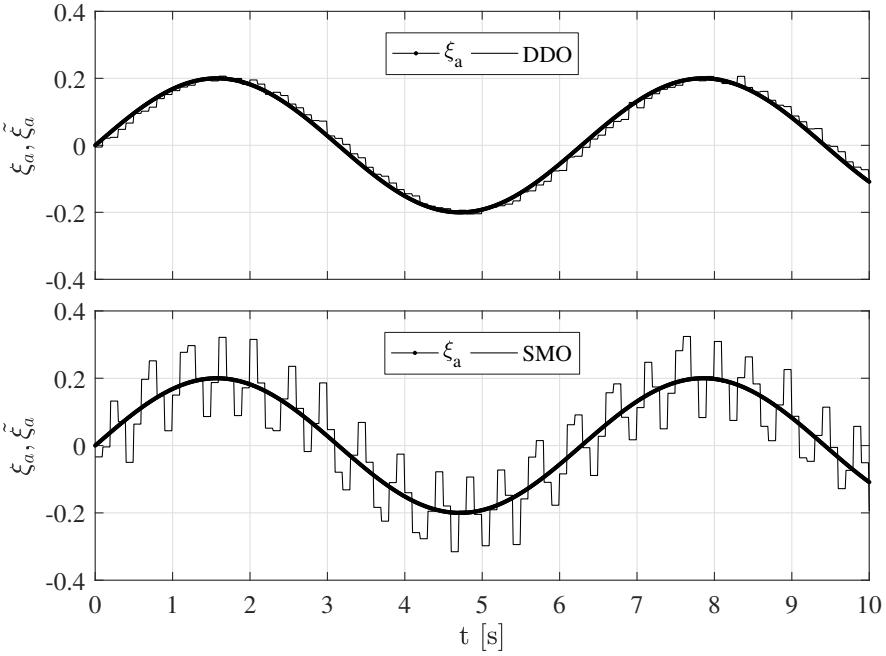

Figure 2.9: Disturbance estimation, $T_s = 10^{-1}s$

2.5.6 A preliminary closed-loop evaluation

We apply PDSMC control and the discrete disturbance observer 2.5.5 to the same SVM tracking problem employed to evaluate TSST performance, see figure 2.4. No measurement noise is considered. PDSMC control shows improvements w.r.t. both performance indices for small sampling times and fast inner loops as well as the desired graceful performance degradation for slower inner loops and lower sampling times. See figure 2.13 for a closeup of the reaching phase and a short period of the quasi-sliding phase under super twisting control and PDSM control under discrete sampling. We have selected for this illustrative simulation a setting where super twisting control starts to exhibit significant, but still acceptable chattering. Under PDSM control, on the other hand, after a moderate overshoot over the sliding surface no visible chattering is encountered, see also figure 2.12. Encouraged by these preliminary results, we apply the disturbance observer based PDSMC guidance law to the vertical channel of the

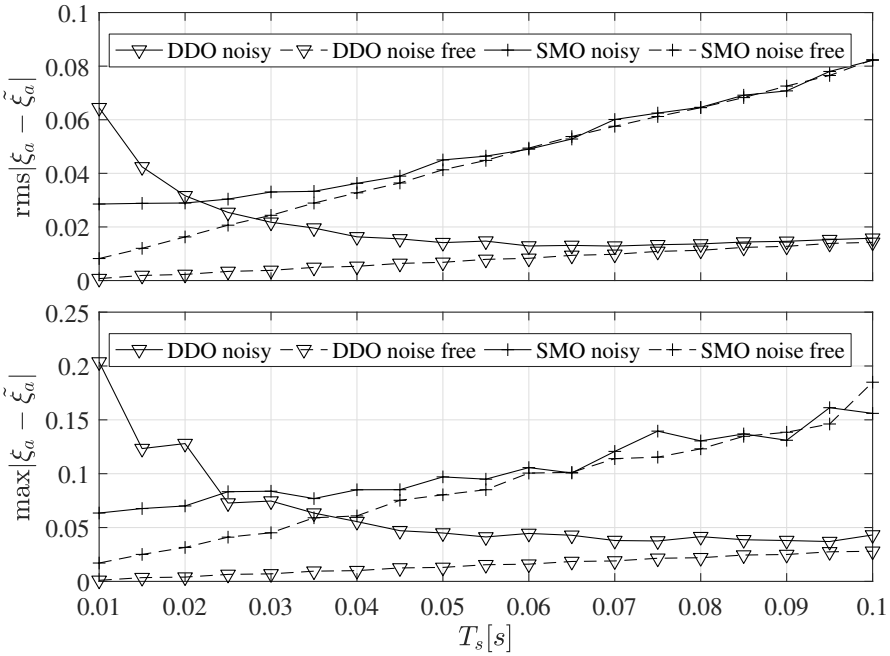

Figure 2.10: Sensitivity of disturbance estimation error w.r.t observation sampling time

more realistic FFB benchmark UAS model (see chapter 1) in combination with the LQ vertical load factor tracking control using direct lift control developed in section 2.4.4.

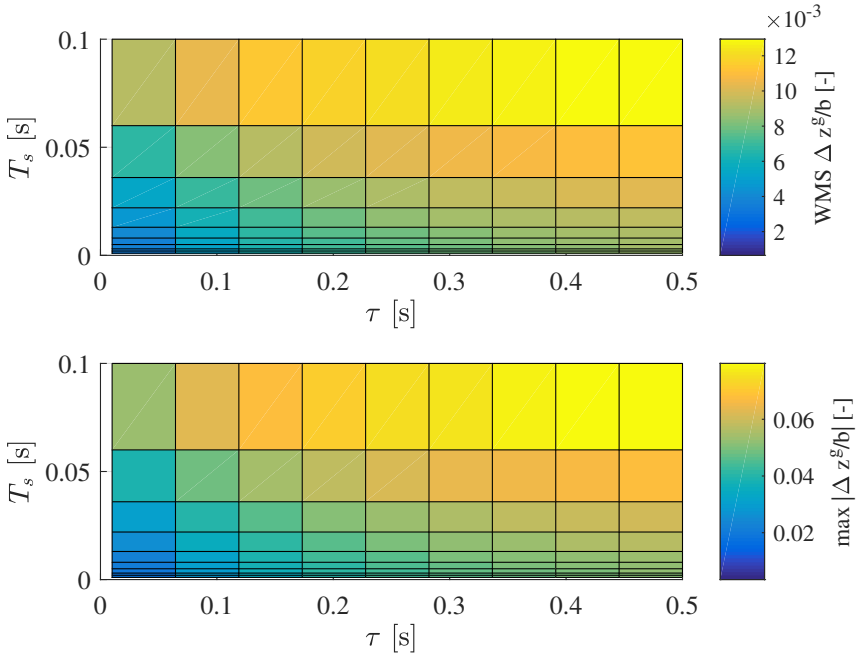

Figure 2.11: Mean squared error and maximum vertical position error under PDSMC control over grid of controller sampling time and inner loop tracking time constant, SVM cruise flight scenario with periodic predecessor motion

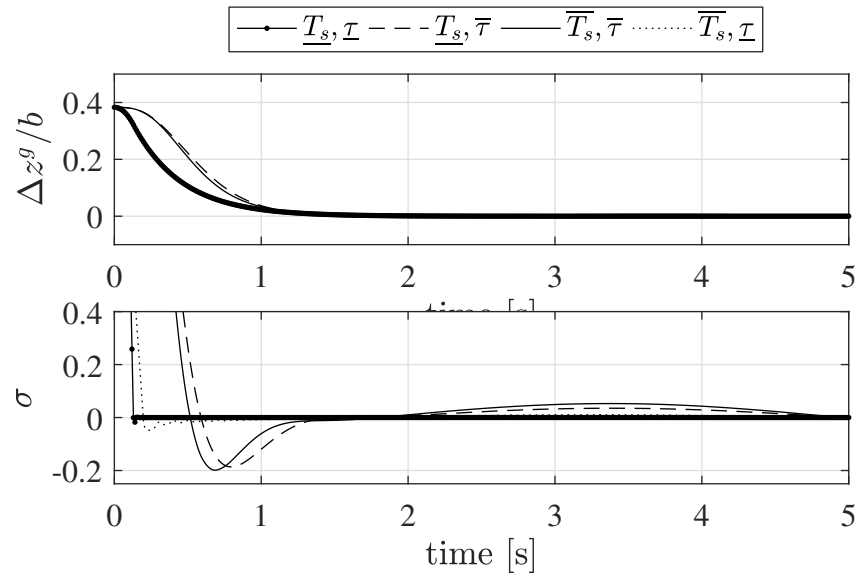

Figure 2.12: Tracking error and sliding variable under PDSMC control for corner points of sampling time and inner loop time constant envelope, SVM cruise flight scenario with periodic predecessor motion

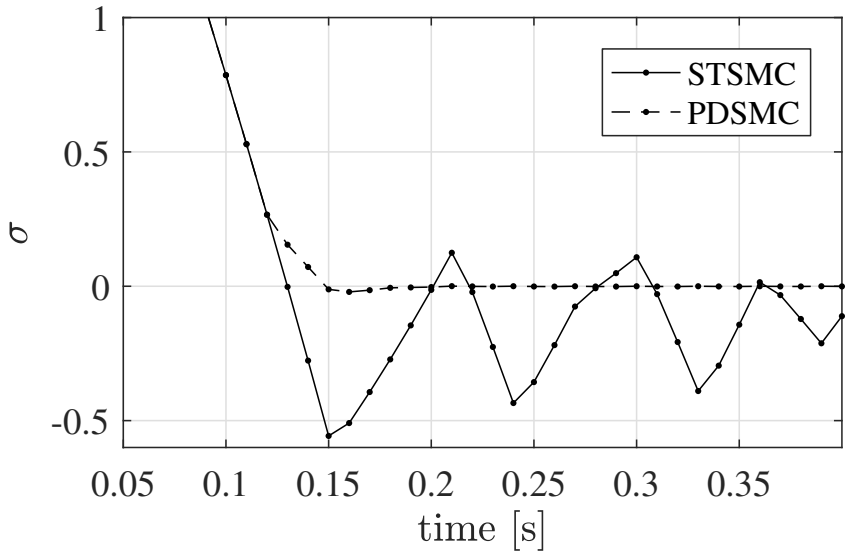

Figure 2.13: Closeup of reaching under sampled super twisting and PDSM control, $T_s = 3 \cdot 10^{-2} s, \tau = 10^{-3} s$

2.6 FFB simulations

We first consider one individual UAS tracking a virtual leader on the benchmark trajectory. Longitudinal and lateral spacing is maintained by baseline LQ control. The trajectory speed in all scenarios is $15\,m/s$. Note that position tracking errors are normalized with span width and commanded load factors as well as elevator and flap commands with their maximum admissible values. A low controller sampling rate of $10\,\mathrm{Hz}$ is selected, compatible with observation rates of low-cost GNSS receivers. Running guidance laws at the same rate as GNSS observations become available, no high-rate exchange of accelerometer observations and no IMU/GNSS data fusion is required, lowering computational load.

2.6.1 Turbulent air

In this scenario, planar turbulent wind of $||\mathbf{v}^e_{w,a}|| = 3\,m/s$ aligned with the cruise flight leg of the FFB trajectory is simulated. Tracking errors of the PDSMC control without disturbance estimation, displayed in figure 2.15) are slightly worse than those achieved by the LQ baseline laws, see figure 2.14). Introducing the disturbance estimation scheme presented in section 2.5.5 appreciably reduces especially peak errors by about $50\,\%$ during predecessor maneuvers.

Large initial errors

Large initial position errors are readily accommodated by the predictive integral sliding surface, see figure 2.17. Peak load factors are of similar magnitude as in the case of small

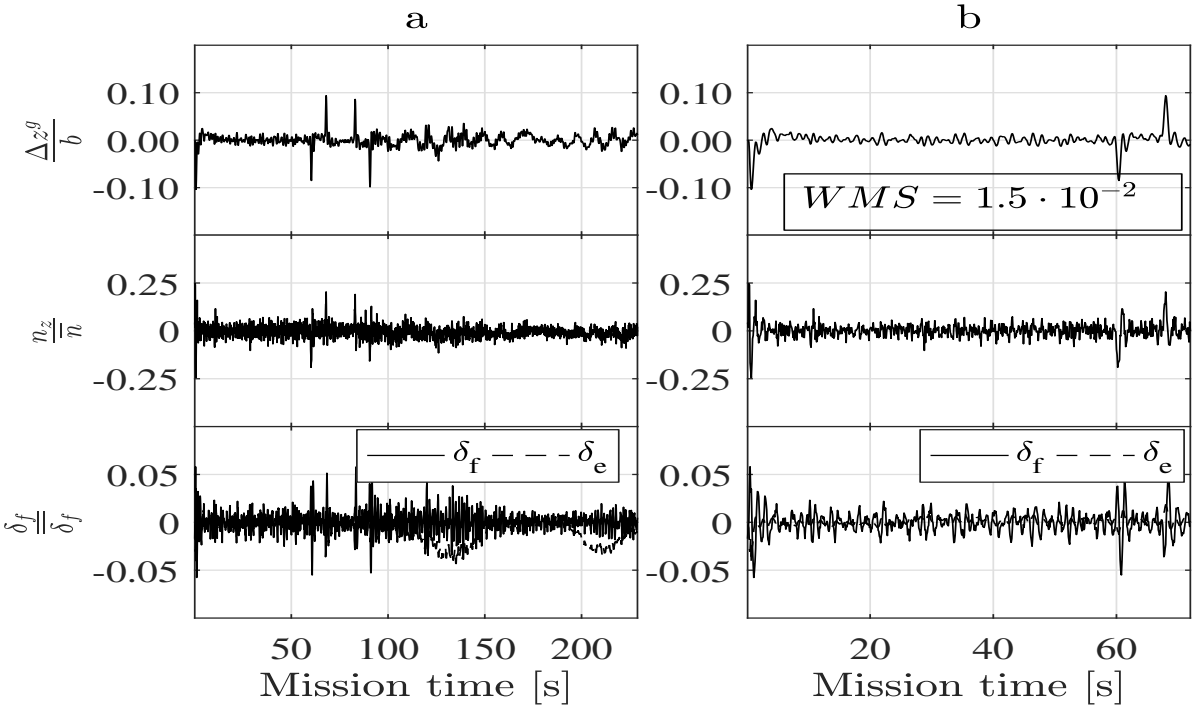

Figure 2.14: Baseline control, $T_s = 10^{-1}s$, turbulent air, (a) entire trajectory, (b) closeup of cruise flight phase and first maneuver. The WMS error applies to the cruise flight segment between $15\,s$ and $55\,s$

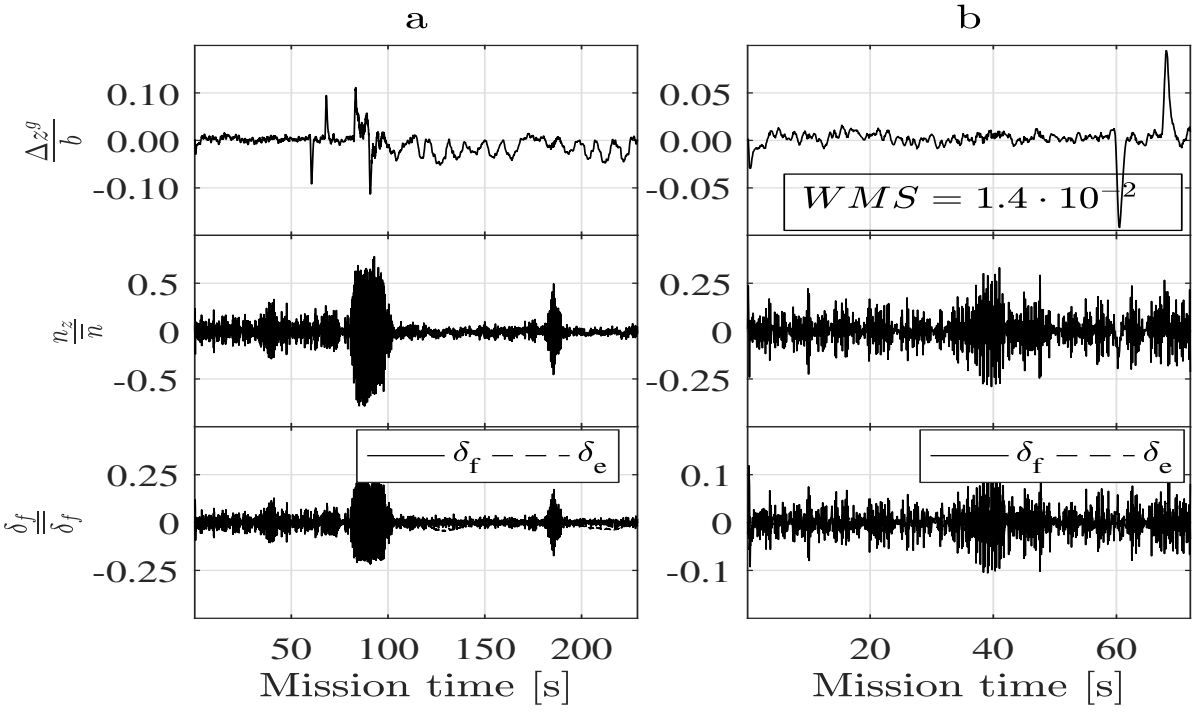

Figure 2.15: PDSMC, $T_s = 10^{-1}s$, turbulent air, no disturbance estimation, (a) entire trajectory, (b) closeup of cruise flight phase and first maneuver. The WMS error applies to the cruise flight segment between $15\,s$ and $55\,s$

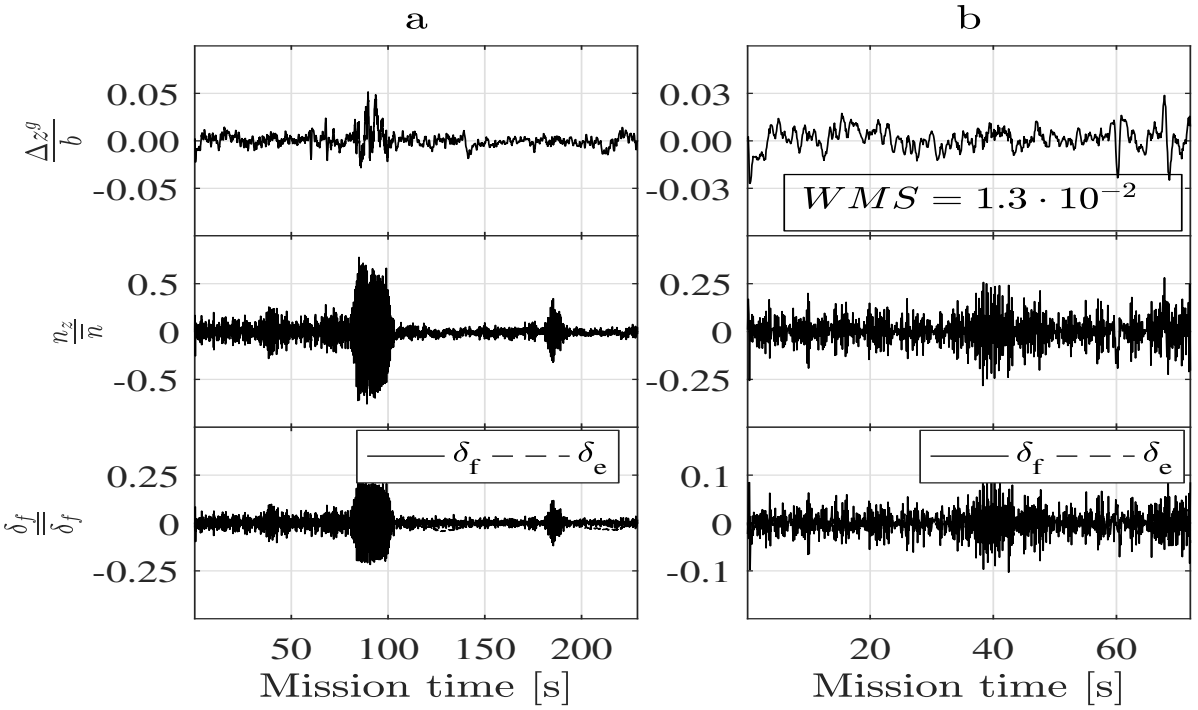

Figure 2.16: PDSMC, $T_s = 10^{-1}s$, turbulent air, disturbance estimation, (a) entire trajectory, (b) closeup of cruise flight phase and first maneuver. The WMS error applies to the cruise flight segment between $15\,s$ and $55\,s$

initial errors despite an initial relative position error of five wingspans, illustrating one of the principal benefits of the constrained optimization based sliding surface.

2.6.2 Calm air

Flying the benchmark in calm air confirms the importance of disturbance estimation to suppress tracking errors due to predecessor motion. The basic PDSMC scheme (see figure 2.19) again displays maximum tracking errors inferior to the baseline scheme (see figure 2.18). Disturbance estimation improves performance even more visibly than under turbulence, see. figure 2.20.
Overall, the tracking errors observed, if maintained in flight tests, are sufficiently small to enable considerable upwash gains. The performance gain is more clearly visible by a direct overlay, see figure 2.21.

2.6.3 Scalability

When adding more members to the formation, linear control with local state feedback is known to lead to position errors growing with vehicle index, i.e. mesh instability [70]. As shown in chapter 2, the PDSMC for $N = 1$ boils down to linear control, leading us to expect mesh instability when applying it to larger formations. This expectation is confirmed by simulations of three vehicles, see figure 2.22. Peak vertical position

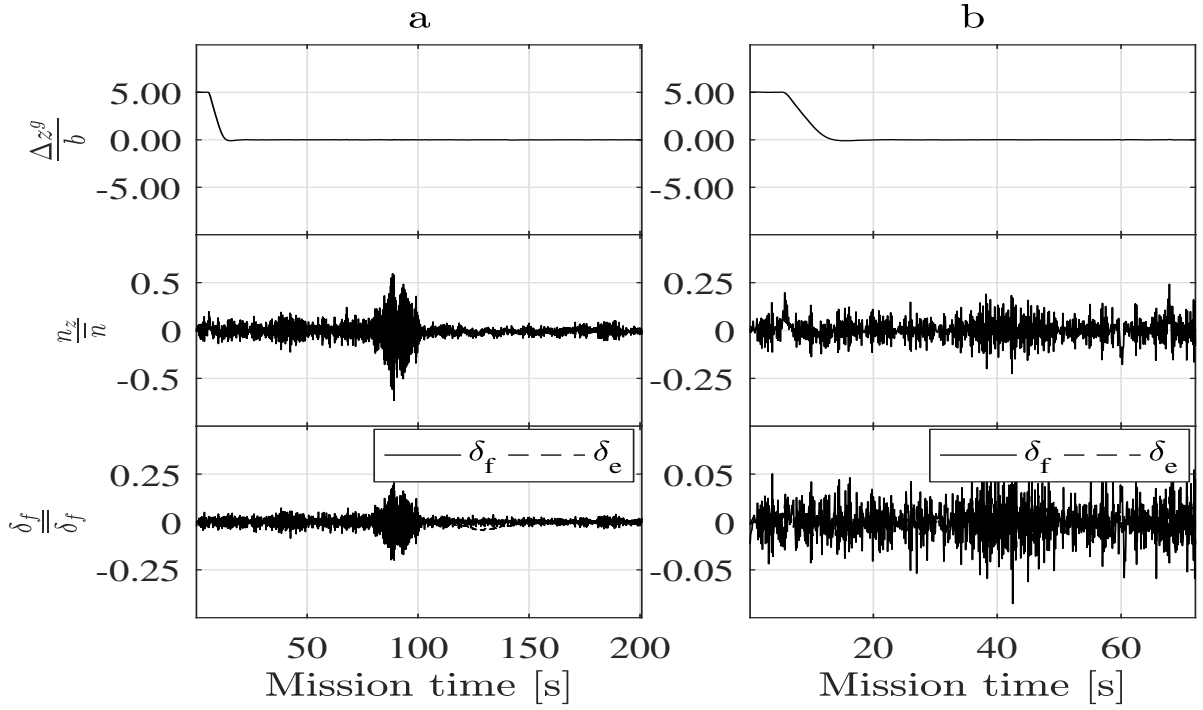

Figure 2.17: PDSMC, $T_s = 10^{-1}s$, turbulent air, disturbance estimation, large initial deviation, (a) entire trajectory, (b) closeup of cruise flight phase and first maneuver

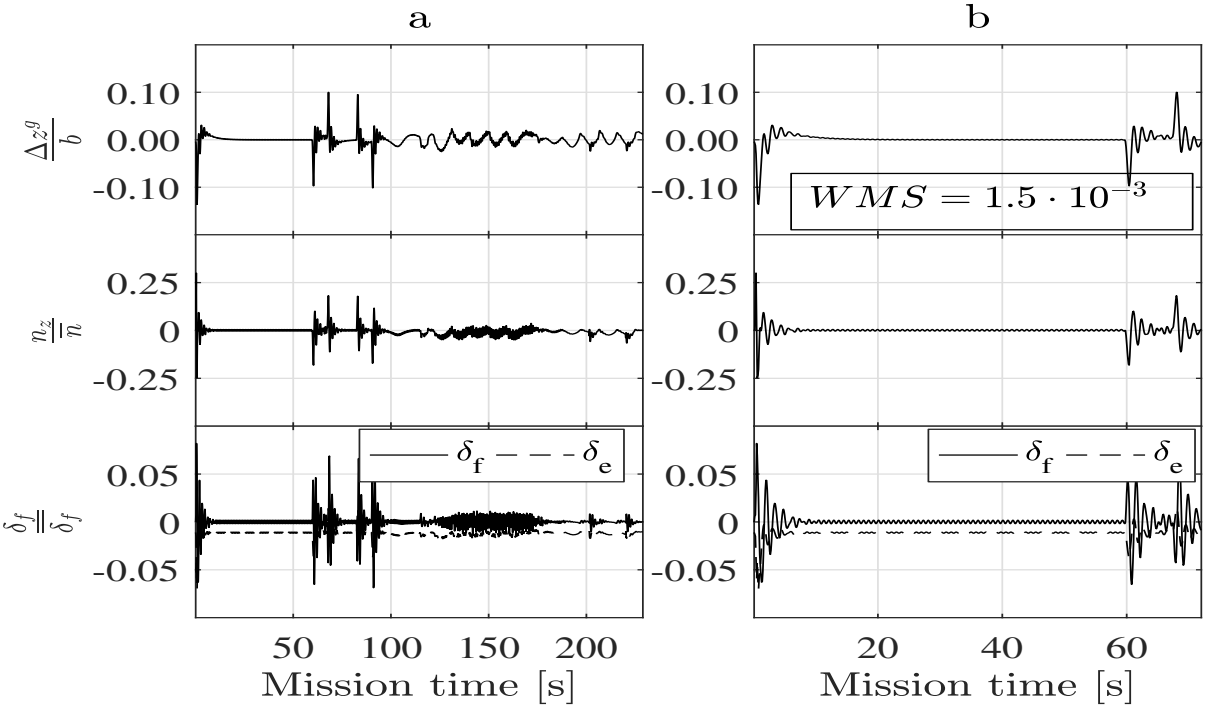

Figure 2.18: Baseline control, $T_s = 10^{-1}s$, calm air, (a) entire trajectory, (b) closeup of cruise flight phase and first maneuver. The WMS error applies to the cruise flight segment between $15\ s$ and $55\ s$

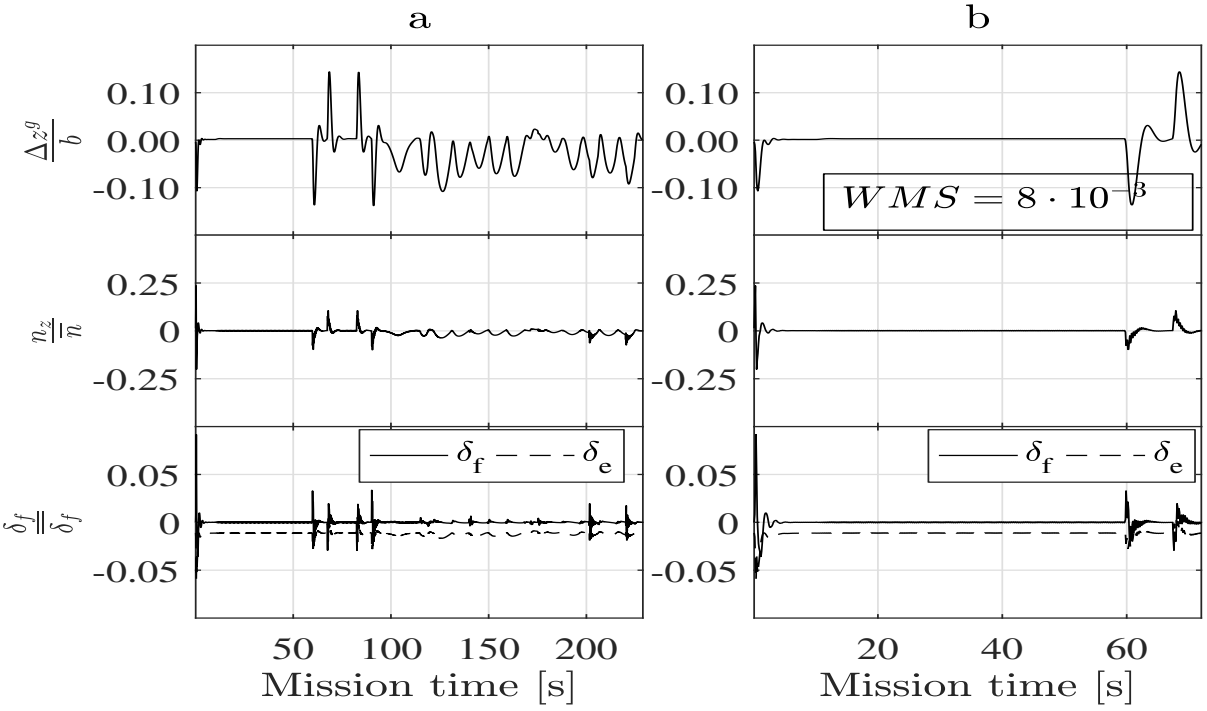

Figure 2.19: PDSMC, $T_s = 10^{-1}s$, calm air, no disturbance estimation, (a) entire trajectory, (b) closeup of cruise flight phase and first maneuver. The WMS error applies to the cruise flight segment between $15\,s$ and $55\,s$

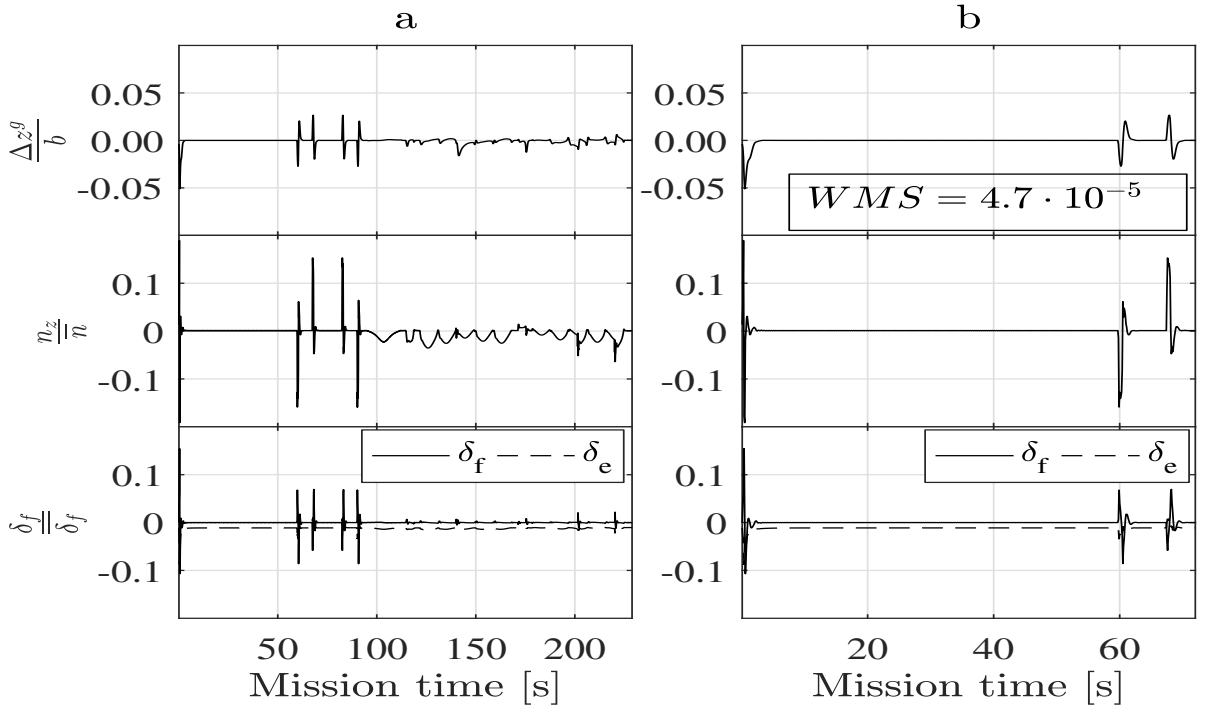

Figure 2.20: PDSMC, $T_s = 10^{-1}s$, calm air, disturbance estimation, (a) entire trajectory, (b) closeup of cruise flight phase and first maneuver. The WMS applies to the cruise flight segment between $15\,s$ and $55\,s$

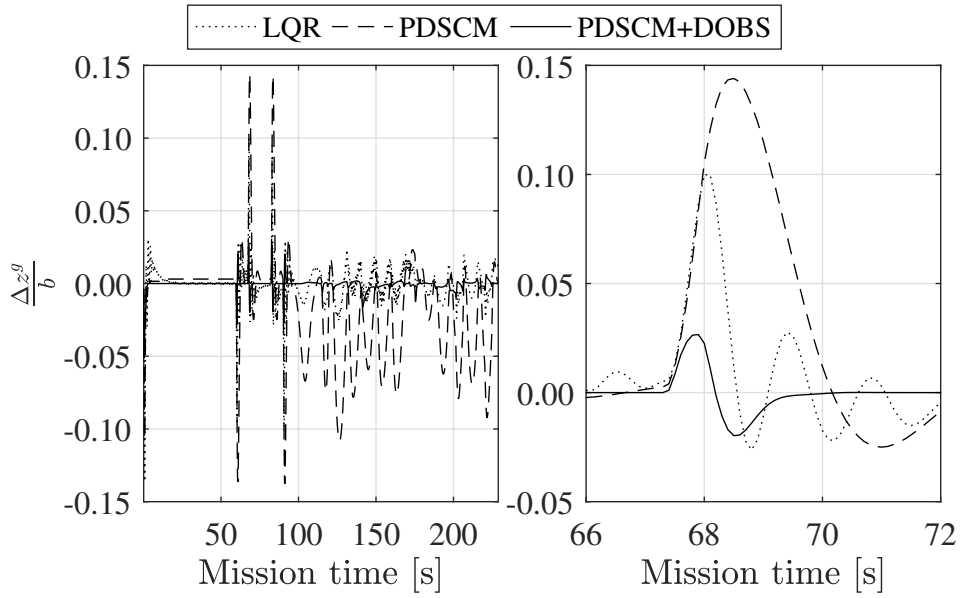

Figure 2.21: Vertical tracking error under baseline, PDSMC, PDSMC with disturbance observation, entire mission (left) and closeup of first maneuver (right)

control errors grow consistently from the leading to the trailing vehicle, suggesting that, while promising excellent performance for applications such as aerial refueling, the proposed control accommodates only small formations. Remarkably, scalability issues are encountered with local feedback continuous time super twisting control under fast sampling as well, see appendix B.

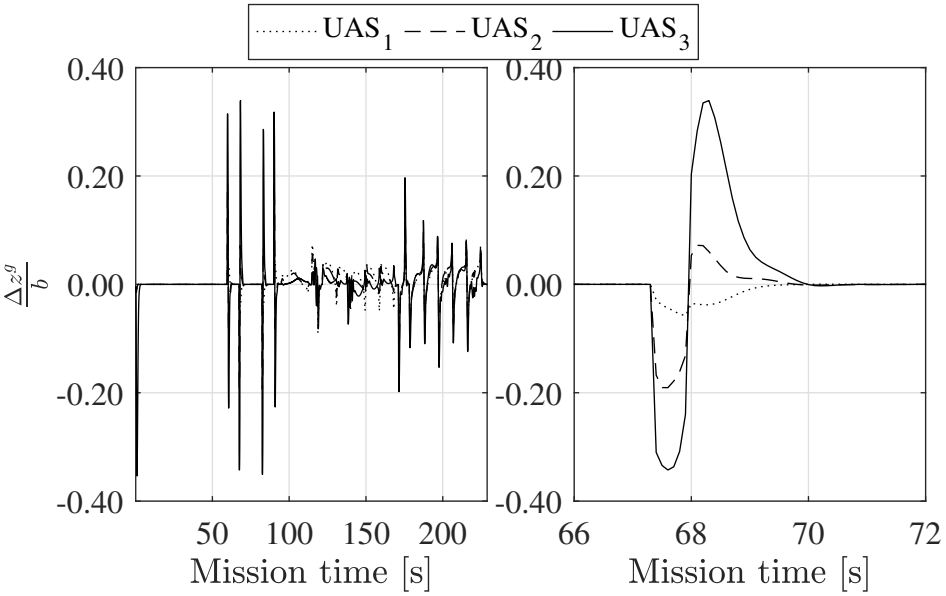

Figure 2.22: N=3, PDSMC, $T_s = 10^{-1}s$, calm air, disturbance estimation, entire mission (left) and closeup of first maneuver (right)

Chapter 3

Set membership localization

The position of the sweet spot in the wake of a preceding UAS allowing for maximum range enhancement is airframe-dependent and only approximate models of the complex flow field forming the wake are available. Peak-seeking algorithms, see e.g. [71], have been proposed to deal with this uncertainty. Peak seeking algorithms attempt to minimize on-line some cost function related to the wake energy gain to converge to the point of minimum power consumption without precise knowledge of its position. Note however that their gradient based nature relies on convexity of the cost, which can exhibit local minima, see section 1.2. That being said, some exploration of the wake might be necessary to identify a suitable starting point for peak seeking that has a high probability of leading to convergence to the global maximum of wake energy gain.
During this exploratory phase, while peak seeking is active and while flying in a static tight formation, collision avoidance needs to be ensured under localization uncertainty as well as bounded control errors.

Accuracy and safety

Localization errors generally have a greater detrimental effect on formations of small UAS than larger ones, since admissible guidance errors scale with airframe size. Using COTS consumer grade GNSS hardware, relative localization errors in the centimeter range are possible using readily available Real Time Kinematic (RTK) algorithms, see e.g. [22], or commercial low cost RTK systems such as [72], corresponding to the high precision of GNSS carrier phase observations. This accuracy falls well within the bounds required for TFF even for sub-meter class UAS.
Alternative GNSS-independent approaches such as machine vision have been considered but generally provide less accuracy, among other reasons owing to attitude estimation errors of low-cost AHRS, affecting the critical transformation of observations in the camera frame to the local inertial frame, see e.g. [14].

Availability is a major concern when it comes to using GNSS for guidance, and can be prohibitive in close-to ground operations due to multipath and signal obstruction. However, in energy-efficient planar formation flight, excellent availability conditions can be expected.
For civil UAS applications, where furthermore no intentional GNSS jamming or spoofing is to be expected, localization based on single frequency RTK systems appears to be well suited, and localization accuracy is thus not a fundamental issue when it comes to tight formation flight of small UAS. However, to ensure collision avoidance, the unavoidable error of relative position estimates needs to be taken into account.

3.1 Problem statement

To this end, not only point estimates of relative position vectors are required, but rather an outer estimate of the regions that can be guaranteed to contain the other members of the formation. We want furthermore this estimate to be useful, i.e. only as large as necessary and relying only on sensors realistically available on board a typical small low cost UAS. What is more, the employed algorithm needs to be compatible with the still somewhat limited computational resources of today's small UAS.

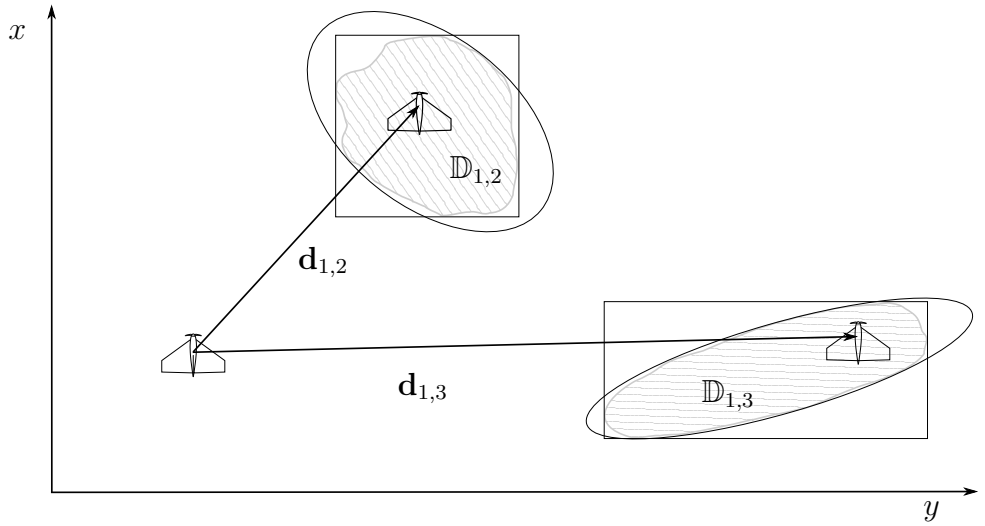

Figure 3.1: Relative localization problem in 2D: guaranteed relative position sets and possible ellipsoidal and interval outer approximations

Assumptions and constraints

Hardware resources We assume that in view of cost and mass constraints affecting in particular small UAS for civil applications, the set of observations and state estimates available for localization is limited to

- The image stream of a single camera, fixed in the body frame

- Attitude estimates and observations of angular rates, accelerations and the earth's magnetic field in the body frame provided by a MEMS based AHRS
- Single-frequency GNSS carrier phase, Doppler and pseudo range observations
- Ultra-Wide Band (UWB) inter-UAS ranging observations

In this chapter we shortly review existing relative localization approaches. We then present a modified Extended Set Membership Filter (ESMF) compatible with the assumed limited set sensor hardware. Care has been taken to replace iterative operations such as numerical optimization by closed-form approximations to enable fixed-time implementation.
We compare the filter to a standard Extended Kalman Filter to illustrate the benefits arising from the deterministic nature of set membership filtering.

3.2 Airborne relative localization: state of the art

Existing relative localization approaches rely mostly on Real-Time-Kinematics (RTK) GNSS positioning, machine vision or a combination of both.

Differential GNSS Using differential GNSS observations, Real Time Kinematics (RTK) systems provide, after ambiguity resolution, estimates of the position vector between two GNSS receivers with the centimeter-level accuracy of carrier phase positioning. GNSS RTK systems usually estimate the unknown carrier phase ambiguities employing pseudo range double differences in what is denoted the "float solution". The estimator of choice is usually a Kalman filter for its ability to estimate the covariance of its estimation error. Once the covariance of the float solution falls within a predefined threshold, a search algorithm is employed that finds with high probability the correct set of carrier phase ambiguities. In that case, the position error is reduced to centimeter level. Two properties of this standard approach are problematic when it comes to flight safety. First, before ambiguities are fixed, only probabilistic bounds on the estimation error of the float solution are available. Second, once ambiguities are fixed, a small nonzero probability that the fixed set of ambiguities is not the actual one remains, due to the statistical nature of the fixing algorithm.

UWB assisted RTK Very cost effective UWB ranging modules based on time of flight observations have emerged on the consumer market over the last years. Their decimeter level accuracy makes them a natural candidate to enhance differential GNSS localization both on the pseudo range level and for RTK solutions. After early experiments for static surveying applications [73], the authors of [22] report robustness enhancements

of a conventional RTK solution based on the LAMBDA [74] algorithm for relative localization of two UAS in flight. In this approach, UWB range observations are employed in forming the float ambiguity estimate.
UWB ranging has as well seen a rising interest for local ground-based localization systems, see e.g. [75, 76].

Machine vision Both techniques purely based on machine vision [77] as well as combinations with differential GNSS [78] have been employed in the community for accurate relative localization of UAS and centimeter level precision is reported in experimental settings, again employing an Extended Kalman Filter.
Machine vision approaches suffer from the fundamental limitation that tracked UASs need to be within the field of view of the tracking UAS for each filter update, necessitating multiple of gimbaled cameras in larger formations as well as trajectory planning algorithms that take into account maintaining continuous visual contact. Maintaining visual contact to multiple neighbors becomes more difficult with decreasing separation, as in tight formation flight close-by UAS might obstruct the view towards other ones.

Algorithms Relative positioning typically involves nonlinear observation equations. This is the case also for GNSS double difference observations, due to small linearization errors usually assumed to be linear in the separation vector (see section 3.4.1 where we derive upper bounds on the linearization error and consider also the impact of standalone position and ephemeris uncertainty). The Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) are the predominant state estimation algorithms employed. The Kalman Filter can indeed provide an ellipsoidal confidence set that is guaranteed to contain the true state with a selected probability given that its underlying assumptions of normally distributed, zero mean modeling errors and measurement errors are met. However, although the normal distribution in many cases provides a reasonable model (see figure 3.5), as is well known, in reality both error sources can only be approximated by it and filter consistency needs to be ensured by careful tuning. Often no attempt to identify the probabilistic properties of the linearization error is made, and the filter is tuned in a grey-box fashion.
What is more, Kalman filters, being based on probabilistic error models, provide estimates of probabilistic state estimation error bounds. Even given a consistent estimate, probabilistic bounds can however only guarantee probabilistic collision avoidance. Guaranteed bounds of estimation errors are however of crucial importance for flight safety. Set membership filtering is concerned with the problem of providing tight outer approximations of these bounds.

3.3 Set Membership Filters for State Estimation

Set membership state estimation filters are the deterministic counterpart to probabilistic filtering frameworks such as the various flavors of the Kalman filter and have received considerable interest for applications that require guaranteed bounds on state estimation errors. They deal with the problem of finding mathematically tractable outer approximations of the set of states compatible with all past observations of the state with their respective error bounds and the constraints imposed by a state transition model with bounded prediction error. An exact representation of this set is generally difficult or impossible to compute and can take on very complex shapes over time.

Assume an unknown state $\mathbf{x}_k \in \mathbb{R}^n$ contained in the **guaranteed state set** $\mathbb{D}_k$. The states in $\mathbb{D}$ evolve according to

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{3.3.1}$$

where $\mathbf{w}_k \in \mathbb{W}_k$ is some bounded prediction error. Using equation (3.3.1), at each discrete sampling instant the predicted guaranteed set $\mathbb{D}_k^+$ can be computed, usually expanding $\mathbb{D}_{k-1}$. This is analogous to the state estimate covariance prediction step of a Kalman Filter. Assume now that some sensor provides observations $\mathbf{y}_k \in \mathbb{R}^p$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{e}_k) \tag{3.3.2}$$

with a bounded measurement error $\mathbf{e}_k \in \mathbb{E}_k$.
From $\mathbf{y}$ and $\mathbb{E}$ an observation set $\mathbb{Y}$ can be computed, given by the set-valued observation equation

$$\mathbb{Y}_k = H(\mathbb{D}_{\mathbb{Y},k}) \tag{3.3.3}$$

where the **consistent state set** is

$$\mathbb{D}_{\mathbb{Y},k} = \{\mathbf{x} | h(\mathbf{x}) \in \mathbb{Y}_k\} \tag{3.3.4}$$

i.e. the set of all the states that yield an observation inside the observation set $\mathbb{Y}$. Whenever new measurements $\mathbf{y}$ come in, the intersection of the current predicted state set $\mathbb{D}^+$ with $\mathbb{D}_\mathbb{Y}$ provides an updated state set, analogous to the Kalman filter measurement update. This is the standard set membership filtering problem.

As exact representations of $\mathbb{D}_k$ are usually computationally intractable, numerous approaches have emerged that employ various outer set approximations of lower complexity, rendering the set sum and intersection operations that form the basis of set membership filtering actually implementable.
Application of set membership filtering to the problem of relative localization of UAS

does not appear to have received a lot of attention so far, especially in contrast to localization schemes involving Kalman filtering. Two examples of related problems are given in [79], where GPS standalone position measurements are considered, comparing interval and zonotope set representations, and [80], studying a cooperative ellipsoid set membership filter for joint target observation by a group of UAS. Interval membership localization filtering based on standalone pseudo range observations is presented in [81] for ground vehicle localization. An interval representation is considered as well in [82] for set membership localization filtering of underwater vehicles. The authors of [83] present a cooperative set membership localization algorithm for ground vehicles based on shared pseudo range observations.
Differential GNSS observations as well as inter-UAS ranging observations generally lead to arbitrarily oriented guaranteed state sets that can be largely overapproximated by intervals. Thus set representations with additional degrees of freedom are desirable to obtain less conservative constraints for trajectory planning, freeing up more maneuvering space. What is more, tight set approximations can be decisive for the success of a mission relying on tight formation flight - if relative position set estimates are too large, the members of the formation possibly cannot reach their assigned energy-efficient position in the wake.
Set membership filters employing ellipsoids [84, 85] have been proposed, and even more flexibility at the cost of computational complexity can be obtained by bounding the guaranteed state set by zonotopes [86, 87], a special class of polyhedra. In this work we present an adaptation of the Extended Set Membership Filter [85] based on ellipsoidal set representations for the relative localization problem. The ESMF is particularly interesting in this context for its direct comparability with the widely popular Extended Kalman Filter.

Remark The major benefit of set membership filtering is the absolute nature of the given guarantees. It relies on our assumed ability to guarantee bounds on observation errors and system modeling errors. In reality, these error bounds are often obtained from a finite amount of observation samples. Assuming even a bounded - thus staying in a deterministic framework - non-uniform probability distribution of the error, this finite observation window might be too small to detect large errors of low probability. Thus, in a strict sense no guarantee, but a high confidence can be given that all future samples will respect the bounds thus obtained, interestingly giving a probabilistic aspect to this so decidedly deterministic class of filtering.

In what follows, after providing some essential notation, we derive bounded-error linearizations of differential GNSS code phase and carrier phase observations as well as of inter-UAS range observations. We then present a modified Extended Set Membership filter that exploits these three observations to provide an ellipsoidal outer approxima-

tion of the set containing the relative position vector between UAS. We compare the modified ESMF to a standard EKF in simulation.

Definitions and Notation

In this chapter we make use of interval as well as ellipsoid representations of sets. Some relevant notions and notations are given in the following for ease of exposition.

Intervals Apart from regular scalar real valued intervals we encounter interval vectors as well as interval matrices in this chapter. We use a general interval matrix notation $[\mathbf{x}]$ for the family of matrices $\mathbf{x} \in \mathbb{R}^{n\times m}$ where $x_{i,j} \in [\underline{x}_{i,j}, \overline{x}_{i,j}]$ for $i = 1...n, j = 1...m$ to cover all three cases. For $n > 1, m = 1$, $[\mathbf{x}]$ is an interval column vector, for $n = 1, m = 1$, $[\mathbf{x}]$ is an interval scalar.
Wherever additive errors are assumed, it can be more convenient to write intervals in center-range notation as $[\mathbf{x}] = \mathbf{x} + [\mathbf{w}_\mathbf{x}]$ where $\mathbf{x}$ is the center point of $[\mathbf{x}]$ and $[\mathbf{w}_\mathbf{x}] = [\mathbf{x}] - [\mathbf{x}, \mathbf{x}]$ where $[\mathbf{x}, \mathbf{x}]$ denotes a degenerate interval containing only the point $\mathbf{x}$.
Wherever regular terms and interval terms appear in the same expression, such as $[c] = a + [b]$ it is assumed that the regular terms (a) represent degenerate intervals ($[a, a]$) and all operations in the expression are interval operations.

Interval extensions Evaluating the result of a function over interval arguments amounts to constructing an *interval extension* of this function. Interval extensions of various basic operators and functions ($+, -$, exp, log etc.) are known to be sharp inclusions of the real result sets. When constructing interval extensions of more complex functions, care needs to be taken, as alternative but equivalent formulations of the same function can lead to different, possibly heavily conservative result intervals. Take as a purely illustrative example the functions

$$f_1(x) = x - x \tag{3.3.5}$$
$$f_2(x) = x(1 - 1) \tag{3.3.6}$$

Whereas f_1, f_2 are equivalent, evaluating both over the interval $[x] = [-1, 1]$ leads to vastly different results:

$$[f_1([x])] = [2, 2] \tag{3.3.7}$$
$$[f_2([x])] = [0, 0] \tag{3.3.8}$$

This fundamental issue is linked to a variable appearing more than once in an expression. When evaluating the function using interval arithmetic, the dependency between both occurrences - the fact that both represent the same variable - cannot be taken

into account, resulting in inflated result intervals. There is no systematic way to avoid dependency issues for general functions, except reducing their impact by skillful rearrangement of terms.
As a useful exception, for (piecewise) *monotonic* functions dependency issues can be circumvented altogether. Monotonic functions are either increasing or decreasing over the interval range of their argument and it suffices to evaluate the corner points of an argument interval and to select the results that form the widest interval. For functions with vector argument $\{y = f(\mathbf{x}) | \mathbf{x} \in \mathbb{R}^n, y \in \mathbb{R}\}$, monotonicity can be verified on a by-element basis [88]. If the elements of the gradient of $f(\mathbf{x})$ do not change sign over a given input interval vector range, the function is monotonic w.r.t. $\mathbf{x}$ and it suffices to evaluate the 2^n corner points of $[\mathbf{x}]$ to find a sharp inclusion of $[f(\mathbf{x})]$. Monotonicity is exploited in section 3.5 to find sharp inclusions of linearization errors.

Ellipsoids The ESMF relies on ellipsoidal set approximations, and the probability level sets of the KF covariance matrices form ellipsoids as well. An ellipsoid in $\mathbb{R}^n$ is given by

$$\Omega(\mathbf{x}_c, \mathbf{P}) = \{\mathbf{x} | (\mathbf{x} - \mathbf{x}_c)^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{x}_c) \leq 1\} \tag{3.3.9}$$

with $\mathbf{x} \in \mathbb{R}^n$ some point in space, $\mathbf{x_c} \in \mathbb{R}^n$ the ellipsoid center and a positive definite matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ defining half axes and orientation. Useful for analyzing ellipsoid size metrics and displaying the ellipsoid surface in 3D, equation (3.3.9) can be decomposed into a diagonal matrix $\mathbf{P}_0 \in \mathbb{R}^{n \times n}$ (corresponding to an axes-aligned ellipsoid) carrying the squares of the half axes on its diagonal, and a matrix $\mathbf{R} \in \mathbb{R}^{n \times n}$ performing the rotation to its original attitude

$$(\mathbf{x} - \mathbf{x}_c)^T (\mathbf{R}^T \mathbf{P}_0 \mathbf{R})^{-1} (\mathbf{x} - \mathbf{x}_c) \leq 1 \tag{3.3.10}$$

The columns of $\mathbf{R}$ are the eigenvectors of $\mathbf{P}^{-1}$, while the diagonal of $\mathbf{P}_0$ can be computed from the eigenvalues of $\mathbf{P}^{-1}$. For $n = 3$, $\mathbf{R}$ is the DCM defined by the half axis unit vectors of the ellipsoid forming a right-handed Cartesian frame.

An alternative representation employed for ellipsoid intersections with a strip in section 3.5 is given by describing the ellipsoid surface as a unit ball $\mathbf{w}$ under a linear transform $\mathbf{V}$

$$E(\mathbf{x}_c, \mathbf{V}) = \{\mathbf{x} | \mathbf{x} = \mathbf{x}_c + \mathbf{V}\mathbf{w}, ||\mathbf{w}|| \leq 1\} \tag{3.3.11}$$

See [89] for useful transformations between different ellipsoid representations.

3.4 Observation models

For both the ESMF and the EKF we consider in a first step GNSS code phase and carrier phase observations. We then integrate UWB ranging observations as a first step to robustify the localization scheme against GNSS outages and investigate how this additional information affects filter performance. In the following GNSS and inter-UAS ranging observation models and their respective linearizations required for the ESMF and the EKF are given. Observation error intervals are derived including parametric uncertainty for set membership estimation. All operations are performed in a local NED frame and satellite positions are assumed to have been transformed into this frame.

3.4.1 GNSS differential code phase observations

GNSS receivers are most popularly used for standalone absolute positioning. However, internally they measure the phase shift between an internal copy of the signals emitted by satellites in space and the received signals. This principle is applied to the sinusoidal carrier phase signal as well the pseudo-random number (PRN) binary code modulated on top of it (phase shifts are usually converted to units of distance for convenience). Due to the principles employed to measure both phase shifts, the measurement noise of code phase observations exceeds the millimeter-level carrier phase noise by several levels of magnitude. Carrier phase observations, on the other hand, are inherently ambiguous due to their periodicity and short wavelength.
From both phase shifts, range information can be derived, making assumptions about the speed of signal propagation through the atmosphere. Variations of the speed of propagation are at once the major source of error (see e.g. [90] for a broader treatment of GNSS ranging principles) and highly correlated for closeby receivers. This spatial correlation is exploited by differential GNSS positioning techniques. By subtracting observations of two receivers, exogenous errors can be almost completely eliminated. By differencing, absolute position information is lost, and instead the relative position between receivers can be computed. In the following, we employ both differential code phase and carrier phase observations, each to benefit from its respective complementary qualities.
GNSS code phase observations, often denoted as pseudo ranges, of UAS i w.r.t. satellite p in units of distance are given by

$$\begin{aligned}\rho_{p,i} =& r_{p,i} + c(\Delta T_i - \Delta T_p) + \Delta r_{I,p,i} + \Delta r_{T,p,i} \\ & \Delta r_{E,p,i} + \Delta r_{M,p,i} + \Delta r_{DLL,p,i}\end{aligned} \tag{3.4.1}$$

with $r_{p,i} = ||\mathbf{p}_i - \mathbf{p}_p||$ the slant range between receiver i and satellite p, c the speed of light, ΔT_i the receiver clock bias, ΔT_p the satellite clock bias, $\Delta r_{I,p,i}$ the ionospheric

range error, $\Delta r_{T,p,i}$ the tropospheric range error, $\Delta r_{E,p,i}$ the ephemeris error, $\Delta r_{M,p,i}$ the range error due to multipath and $\Delta r_{DLL,p,i}$ the code phase tracking tracking noise. Most of these code phase observation errors are highly correlated spatially and can largely be eliminated by forming single differences at epoch k between two receivers on board UAS i and UAS j, leading to

$$\begin{aligned}\nabla\rho_{p,i,j} =& \rho_{p,i} - \rho_{p,j} && (3.4.2)\\ =& (r_{p,i} - r_{p,j}) + c(\Delta T_i - \Delta T_j) \\ &+ (\Delta r_{M,p,i} - \Delta r_{M,p,j+} + \Delta r_{DLL,p,i} - \Delta r_{DLL,p,j}) && (3.4.3)\\ =& \nabla r_{p,i,j} + c\nabla\Delta T_{i,j} + \nabla\epsilon_{p,i,j} && (3.4.4)\end{aligned}$$

where $\nabla\epsilon_{p,i,j}$ lumps together residual differential observation errors due to uncorrelated hardware noise and multipath. Note in particular that since UAS flying in formation are typically close in altitude, tropospheric propagation delays are highly correlated between receivers and thus largely cancel out. It is convenient to then form double differences w.r.t. to a reference satellite to remove the receiver clock bias common to all observations, thereby later reducing the localization filter state dimension by one. Double differencing single differences w.r.t. a reference satellite r (usually the maximum elevation satellite in view to minimize noise and multipath contamination of double difference observations), leads to

$$\nabla^r\rho_{p,i,j} = ||\mathbf{p}_i - \mathbf{p_s}|| - ||\mathbf{p}_i + \mathbf{d}_{i,j} - \mathbf{p_s}|| - ||\mathbf{p}_i - \mathbf{p_r}|| + ||\mathbf{p}_i + \mathbf{d}_{i,j} - \mathbf{p_r}|| + \nabla^r\epsilon_{p,i,j} \quad (3.4.5)$$

with the relative position vector $\mathbf{d}_{i,j} = \mathbf{p}_j - \mathbf{p}_i$ between receivers.

Parametric uncertainty

In reality, the first receiver position $\mathbf{p}_i$ is only known with meter-level standalone precision. Furthermore, broadcast ephemeris are only local approximations of the true satellite orbits, and RMS deviations between predicted and true satellite positions up to tens of meters are recorded [91, 92], adding additional parametric uncertainty to the observation equation 3.4.5.

This parametric uncertainty is traditionally neglected in the GNSS literature [90] due to their submillimeter impact owing to high GNSS satellite orbits (roughly $2 \cdot 10^4\,\mathrm{km}$), i.e its effect tends to "disappear" in the always present receiver hardware observation noise. The first order Taylor expansion of the observation equation (3.4.5) w.r.t. to $\mathbf{d}$ about $\mathbf{d} = \mathbf{0}$ leads then to the familiar linear double difference pseudo range observation equation

$$\nabla^r\rho_{p,i,j} = \mathbf{H}_{p,r}\mathbf{d}_{i,j} + R_\rho + \nabla^r\epsilon_{p,i,j} \quad (3.4.6)$$

with the geometry matrix $\mathbf{H}_{p,r} = \frac{\partial \nabla^r \rho}{\partial \mathbf{d}}\big|_{\mathbf{d}=\mathbf{0}}$ The small remainder R_ρ is again commonly neglected in the differential GNSS literature. When applying probabilistic filters such as the EKF, neglecting the ensemble of these small effects can be a reasonable simplification due to their millimeter-level magnitude compared to the typically meter-level code phase hardware noise.

Interval inclusion of linearization errors

The deterministic nature of the ESMF requires however guaranteed bounds on the observation error, forcing us to actually quantify these small error terms. To obtain these bounds, we follow in principle the approach proposed in [85] and perform an interval evaluation of the Taylor expansion's remainder term. In the following we consider without loss of generality one element y_i of an output vector $\mathbf{y}$. The Lagrange remainder being bounded by the interval inclusion

$$[R_i([\mathbf{x}])] = ([\mathbf{x}] - \hat{\mathbf{x}})^T \frac{\partial^2 h_i([\mathbf{x}])}{\partial \mathbf{x}^2} ([\mathbf{x}] - \hat{\mathbf{x}}) \tag{3.4.7}$$

interval bounds are obtained by evaluating R_i over $[\mathbf{x}]$, the interval containing the state. In [85], no parametric uncertainty is considered, i.e. it is assumed that the output

$$y_i = h_i(\mathbf{x}) \tag{3.4.8}$$

depends only on the state. To deal with observation equations subject to parametric uncertainty of type

$$\mathbf{y}_i = h_i(\mathbf{x}, \theta) \tag{3.4.9}$$

where interval vector bounds on the parameters θ are known, we proceed as follows. By adopting an augmented state

$$\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x}^T & \theta^T \end{pmatrix} \tag{3.4.10}$$

we can write

$$\mathbf{y} = h(\tilde{\mathbf{x}}) \tag{3.4.11}$$

The state as well as the parameters are bounded by an interval vector $[\tilde{\mathbf{x}}]$ with the interval center $\tilde{\mathbf{x}}_0$ corresponding to the current state point estimate and nominal parameter values. The first order interval Taylor expansion about $\tilde{\mathbf{x}}_0$ is then given by

$$[y_i] = h(\tilde{\mathbf{x}}_0) + \left.\frac{\partial h}{\partial \tilde{\mathbf{x}}}\right|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}}_0} ([\tilde{\mathbf{x}}] - \tilde{\mathbf{x}}_0) + [R_i([\tilde{\mathbf{x}}])] \tag{3.4.12}$$

We split the gradient

$$[y_i] = h(\tilde{\mathbf{x}}_0) + \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} (\mathbf{x} - \mathbf{x}_0) + \left.\frac{\partial h}{\partial \theta}\right|_{\theta=\theta_0} ([\theta] - \theta_0) + [R_i([\tilde{\mathbf{x}}])] \tag{3.4.13}$$

Evaluating the last two terms of equation (3.4.13) over $[\theta]$ and $[\tilde{\mathbf{x}}]$ respectively provides then interval bounds on the error of the combined parametric and linearization error. This procedure to incorporate parametric uncertainty is a general one, but depending on the structure of the observation equation, may not provide very tight interval enclosures as we will see in the following. In some cases the degree of overapproximation is acceptable. in other cases, as will be the case for inter UAS range observations, interval evaluations of the parametric error tailored to the problem enable sharper enclosures.

Application to pseudo range double differences

Before applying this method to the double difference observation equation (3.4.5), it is convenient to write it as

$$\nabla\rho = \sum_{i=1}^{4} ||\mathbf{E}_i\tilde{\mathbf{x}}|| \tag{3.4.14}$$

$$= \sum_{i=1}^{4} (\tilde{\mathbf{x}}^T\mathbf{E}_i^T\mathbf{E}_i\tilde{\mathbf{x}})^{1/2} \tag{3.4.15}$$

$$= \sum_{i=1}^{4} (\tilde{\mathbf{x}}^T\mathbf{M}_i\tilde{\mathbf{x}})^{1/2} \tag{3.4.16}$$

with

$$\tilde{\mathbf{x}} = (\mathbf{d}^T\ \mathbf{p}^T\ \mathbf{p}_s^T\ \mathbf{p}_r^T)^T \tag{3.4.17}$$

$$\tilde{\mathbf{x}} = (\mathbf{d}^T\ \theta^T)^T \tag{3.4.18}$$

and where the $\mathbf{E}_i$ matrices form the vector argument of each vector norm term. For instance, for the first term of 3.4.5

$$||\mathbf{p} - \mathbf{p_s}|| = ||\mathbf{E}_1\tilde{\mathbf{x}}|| \tag{3.4.19}$$

$$= || \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & -\mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} (\mathbf{d}^T\ \mathbf{p}^T\ \mathbf{p}_s^T\ \mathbf{p}_r^T)^T || \tag{3.4.20}$$

In this compact form the first and second derivative can be formed conveniently as

$$\frac{\partial \nabla\rho}{\partial \tilde{\mathbf{x}}} = \sum_{i=1}^{4} (\tilde{\mathbf{x}}^T \mathbf{M}_i \tilde{\mathbf{x}})^{-1/2} \tilde{\mathbf{x}}^T \mathbf{M}_i \tag{3.4.21}$$

$$\frac{\partial^2 \nabla\rho}{\partial \tilde{\mathbf{x}}^2} = \sum_{i=1}^{4} (\tilde{\mathbf{x}}^T \mathbf{M}_i \tilde{\mathbf{x}})^{-1/2} \mathbf{M}_i^T - \mathbf{M}_i^T \tilde{\mathbf{x}} (\tilde{\mathbf{x}}^T \mathbf{M}_i \tilde{\mathbf{x}})^{-3/2} \tilde{\mathbf{x}}^T \mathbf{M}_i \tag{3.4.22}$$

Interval bounds on the observation model error due to linearization and parametric uncertainty can thus be obtained by evaluating

$$[R_{\nabla\rho}] = \frac{\partial \nabla\rho([\tilde{\mathbf{x}}])}{\partial \theta}\bigg|_{\theta=\theta_0} ([\theta] - \theta_0) + ([\tilde{\mathbf{x}}] - \tilde{\mathbf{x}}_0) \frac{\partial^2 \nabla\rho([\tilde{\mathbf{x}}])}{\partial \tilde{\mathbf{x}}^2} ([\tilde{\mathbf{x}}] - \tilde{\mathbf{x}}_0)^T \tag{3.4.23}$$

A sufficiently small degree of overapproximation can thus be obtained, see figure 3.2 for an example. A large number ($> 10^3$) of samples is drawn from an uniform distribution over $\tilde{\mathbf{x}}$ to approximate the true shape of the set containing the error terms. Note that the obtained intervals are quite expectedly not sharp due to dependency. The order of magnitude and center of the error term is however well (outer) approximated. Note furthermore that due to the small magnitude of the error term, the achieved tightness of the interval is fully sufficient.
Regarding implementation, intervals bounding the receiver standalone position can be found by again applying set membership filtering to the standalone positioning problem, using algorithms such as proposed by [81] or by again applying an ESMF. Note that there is a correlation between satellite ephemeris error bounds and receiver standalone position bounds, since ephemeris errors directly affect receiver positioning errors.
No published assessment of maximum broadcast ephemeris errors is known to the author. This constitutes an open problem for set membership filtering. As a workaround, we assume a pessimistic satellite position error interval of $100m$ in each coordinate.

3.4.2 Time differenced differential carrier phase observations

When basing localization only on pseudorange observations, assumptions need to be made about the maximum displacement of a UAS between samples for the filter propagation step. This in turn requires an accurate model permitting to formulate upper bounds on motion variables such as maximum accelerations, minimum turn radii etc. In the following we describe how time differenced carrier phase observations can be integrated in a relative position estimation filter that provides accurate tracking by treating position increments as virtual system inputs and decouples the filter propagation step from aircraft motion. Note however that this argument is only valid at the discrete sampling instants, i.e. whenever carrier phase observations are available. In between

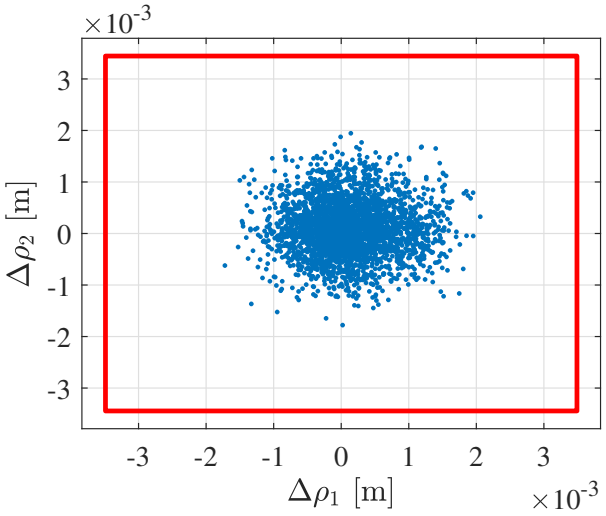

Figure 3.2: Interval bounds on pseudorange double difference linearization error including parametric error for two satellites, Monte-Carlo sampling approximating true error set

samples, assumptions about aircraft motion constraints would still be required to compute a guaranteed relative position set. This aspect is an important future extension of the work presented here.

While the noise level of pseudo range double differences translates into position errors of the order of meters, carrier phase observations have complimentary properties as they feature a very low, i.e. millimeter level noise component (see figures 3.5 and 3.6), but are biased by unknown integer phase ambiguities due to the periodicity of the carrier signal.

GNSS carrier phase observations of UAS i w.r.t. satellite p in units of distance are given by

$$\begin{aligned}\Phi_{p,i} =& r_{p,i} + c(\Delta T_i - \Delta T_p) - \Delta r_{I,p,i} + \Delta r_{T,p,i} \\ & \lambda N_{p,i} + \Delta r_{E,p,i} + \Delta r_{M,p,i} + \Delta r_{PLL,p,i}\end{aligned} \tag{3.4.24}$$

with $r_{p,i}$ the slant range between receiver i and satellite p c the speed of light, ΔT_i the receiver clock bias, ΔT_p the satellite clock bias, $\Delta r_{I,p,i}$ the ionospheric range error, $\Delta r_{T,p,i}$ the tropospheric range error, $N_{p,i}$ the integer ambiguities in carrier signal cycles, $\Delta r_{E,p,i}$ the ephemeris error, $\Delta r_{M,p,i}$ the range error due to multipath, $\Delta r_{PLL,p,i}$ the PLL carrier tracking noise and λ_p the carrier signal frequency. Note that the effect of ionospheric delay is of inverse sign compared to pseudo ranges. Just as is the case for code phase observations, forming the difference between observations of two closeby receivers eliminates common range errors and forming double differences removes the differential receiver clock error. Double difference carrier phase observations with the reference

satellite r and the UAS i and j are given by

$$\begin{aligned}\nabla^r\Phi_{p,i,j} =&\Phi_{p,i} - \Phi_{p,j} - \Phi_{r,i} + \Phi_{r,j} && (3.4.25)\\ =&r_{p,i} - r_{p,j} - r_{r,i} + r_{r,j} + \lambda(N_{p,i} - N_{p,j} - N_{r,i} + N_{r,j}) && (3.4.26)\\ &+ \nabla^r\epsilon^{\Phi}_{p,i,j}\end{aligned}$$

where $\nabla^r\epsilon^{\Phi}_{p,i,j}$ lumps together residual differential observation errors due to uncorrelated hardware noise and multipath. Using the linearized double difference range observation equation (3.4.5), we can write carrier phase observations as

$$\begin{aligned}\nabla^r\Phi_{p,i,j} =&\mathbf{H}_{p,r}\mathbf{d}_{i,j} + R_\Phi + \lambda(N_{p,i} - N_{p,j} - N_{r,i} + N_{r,j}) + \nabla^r\epsilon^{\Phi}_{p,i,j} && (3.4.27)\\ =&\mathbf{H}_{p,r}\mathbf{d}_{i,j} + R_\Phi + \lambda\nabla^r N_{p,i,j} + \nabla^r\epsilon^{\Phi}_{p,i,j} && (3.4.28)\end{aligned}$$

where R_Φ combines the linearization remainder term and parametric observation model uncertainty due to uncertainty in satellite and receiver position. With 3.4.28 the vector of double difference observations to n satellites is

$$\begin{pmatrix}\nabla^r\Phi_{1,i,j}\\ \nabla^r\Phi_{2,i,j}\\ \vdots\\ \nabla^r\Phi_{n,i,j}\end{pmatrix} = \begin{bmatrix}\mathbf{H}_{1,r}\\ \mathbf{H}_{2,r}\\ \vdots\\ \mathbf{H}_{n,r}\end{bmatrix}\mathbf{d}_{i,j} + \begin{bmatrix}R_{\Phi,1}\\ R_{\Phi,2}\\ \vdots\\ R_{\Phi,n}\end{bmatrix} + \begin{pmatrix}\lambda\nabla^r N_{1,i,j}\\ \lambda\nabla^r N_{2,i,j}\\ \vdots\\ \lambda\nabla^r N_{2,i,j}\end{pmatrix} + \begin{pmatrix}\nabla^r\epsilon^{\Phi}_{1,i,j}\\ \nabla^r\epsilon^{\Phi}_{2,i,j}\\ \vdots\\ \nabla^r\epsilon^{\Phi}_{n,i,j}\end{pmatrix} \quad (3.4.29)$$

$$\nabla\mathbf{\Phi} = \mathbf{H}\mathbf{d} + \mathbf{R}_\Phi + \nabla\mathbf{N} + \nabla\epsilon^{\Phi} \quad (3.4.30)$$

Note that from here on we drop unambiguous indices for the sake of readability. Computing $\mathbf{d}_{i,j}$ from equation (3.4.30) requires first fixing the unknown integer ambiguities e.g. by applying some standard method such as LAMBDA [93].
Time-differencing observations locally provides another approach that has originally been introduced for precise offline short term trajectory reconstruction [94]. Forming time differences between observations of one receiver removes, in the absence of cycle slips, carrier phase cycle ambiguities or their respective single or double differences. Furthermore, atmospheric delays largely cancel out due to their slow variation over time and resulting high correlation over consecutive epochs. Variations of atmospheric delays over multiple epochs however unavoidably introduce drift in the standalone time-differenced carrier phase solution. In double differences, these error sources are already canceled. We form thus time differenced double difference (T3D) observations at epoch k w.r.t. an epoch k_0

$$\Delta\nabla\mathbf{\Phi} = \mathbf{H}_k\nabla\mathbf{d}_k - \mathbf{H}_{k_0}\mathbf{d}_{k_0} + \Delta\mathbf{R}_{\Phi,k} + \Delta\nabla\epsilon^{\Phi} \quad (3.4.31)$$

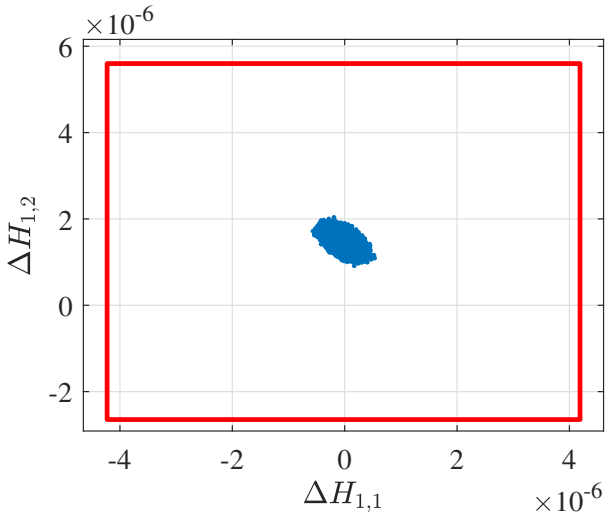

Figure 3.3: Interval range of two matrix elements of $\mathbf{H}$, approximation of true result set

By choosing $k_0 = k - 1$, satellite-receiver geometry varies only very little due to the low airspeeds of small UAS, and we can approximate

$$\Delta\nabla\mathbf{\Phi} \approx \mathbf{H}_k\Delta\mathbf{d} + \Delta\mathbf{R}_{\Phi,k} + \Delta\nabla\epsilon^{\Phi} \tag{3.4.32}$$

and compute the least squares solution of $\Delta\mathbf{d} = \mathbf{d}_k - \mathbf{d}_{k0}$. To find a bound on the approximation error we consider bounds on the variation $\Delta\mathbf{H}$ of the geometry matrix $\mathbf{H}$ between two consecutive epochs by first writing

$$\Delta\nabla\mathbf{\Phi} = \mathbf{H}_k\mathbf{d}_k - (\mathbf{H}_k + \Delta\mathbf{H})(\mathbf{d}_k + \Delta\mathbf{d}) + \Delta\mathbf{R}_{\Phi,k} + \Delta\nabla\epsilon^{\Phi} \tag{3.4.33}$$

$$= \mathbf{H}_k\Delta\mathbf{d} + \Delta\mathbf{H}\Delta\mathbf{d}_k + \Delta\mathbf{H}_k\Delta\mathbf{d} + \Delta\mathbf{R}_{\Phi,k} + \Delta\nabla\epsilon^{\Phi} \tag{3.4.34}$$

By interval arithmetic we can then find interval bounds on the second and third term of equation (3.4.34), see figure 3.3 for an example of the obtained intermediate interval of two elements of $\Delta\mathbf{H}$. Note that the inclusion is not tight, but predicts well the non-zero center and order of magnitude of the true range, approximated by a large number of point samples. The interval sum of the last four terms of equation (3.4.34) provides then an outer bound $[\mathbf{w}_\Phi]$ for the overall observation error of linearized time differenced carrier phase double differences and the observation interval is

$$[\Delta\nabla\mathbf{\Phi}] = \mathbf{H}_k\Delta\mathbf{d} + [\mathbf{w}_\Phi] \tag{3.4.35}$$

The interval least-squares solution of 3.4.35 is then given by

$$[\Delta\tilde{\mathbf{d}}] = (\mathbf{H}_k^T\mathbf{H}_k)^{-1}\mathbf{H}_k^T[\Delta\nabla\mathbf{\Phi}] \tag{3.4.36}$$

Note that this is a modification of the accumulation strategy studied in [95] and inherits the same attractive properties. Most importantly, the property that only a set of at least

four common observations is required over any two consecutive epochs. As a consequence, the availability of the solution 3.4.36 is higher than when using a fixed reference epoch, especially in the dynamic environment of a small UAS. Flight test results with a GPS L1 low-cost receiver [96] suggest that although loss of lock occurs frequently, e.g. in sharp turns, a set of > 4 observations can always be maintained[1]. Note that when using a fixed reference epoch, losing lock to a sufficient majority of the satellites in view between the current and the reference epoch even only once, or a change in the set of satellites in view due to orbital motion can actually break availability of the carrier phase position increment altogether. For any pair of two UAS, the relative position vector interval can thus be propagated in time by

$$[\mathbf{d}_k] = [\mathbf{d}_{k-1}] + [\Delta\tilde{\mathbf{d}}_k] \tag{3.4.37}$$

or alternatively in center-range notation (see 3.3)

$$[\mathbf{d}_k] = [\mathbf{d}_{k-1}] + \Delta\tilde{\mathbf{d}}_k + [\mathbf{w}_{\Delta\mathbf{d}_k}] \tag{3.4.38}$$

3.4.3 Inter-UAS range observations

The range between the antenna centers of UWB ranging devices on board UAS i and j is given by

$$r_{i,j,k} = ||\mathbf{d}_{i,j,k} + \mathbf{R}_{\mathrm{eb,j}}\mathbf{r}_j^b - \mathbf{R}_{\mathrm{eb,i}}\mathbf{r}_i^b|| + w_{r,k} \tag{3.4.39}$$

with the positions $\mathbf{r}_i^b, \mathbf{r}_j^b$ of the UWB antenna centers w.r.t. the GNSS antenna centers in the respective body frame and the device hardware measurement noise $w_r \in [\underline{w}_r, \overline{w}_r]$. Equation (3.4.39) is subject to parametric uncertainty due to uncertainty in the attitudes $\mathbf{R}_{\mathrm{eb,i}}, \mathbf{R}_{\mathrm{eb,j}}$ of both UAS. In the following we assume that $\mathbf{r}_i^b$, $\mathbf{r}_j^b$ are small, i.e. the ranging antennas are close to the GNSS antennas, and that interval bounds of their additive effect on the range observation have been determined. We can then lump their effect on the range observation into a combined range error $\hat{w}_{r,k}$:

$$r_{i,j,k} = ||\mathbf{d}_{i,j,k}|| + \hat{w}_{r,k} \tag{3.4.40}$$

with $\hat{w}_r \in [\underline{\hat{w}}_r, \overline{\hat{w}}_r]$

[1] What is more, the ongoing incorporation of concurrent GNSS systems (GPS, Glonass, Compass, Galileo) in low-cost receivers can be expected to improve availability drastically in the near future.

Linearization

To linearize equation (3.4.39) for the ESMF and EKF, we first write range observations as

$$r = (\mathbf{d}^T\mathbf{d})^{1/2} \tag{3.4.41}$$

dropping indices and subscripts since we always consider UAS i and j at sampling instant k. The first order interval Taylor expansion of equation (3.4.41) follows in the form familiar from pseudo range observations

$$[r] = (\mathbf{d}^T\mathbf{d})^{1/2} + \left.\frac{\partial r}{\partial \mathbf{d}}\right|_{\mathbf{d}=\mathbf{d}} ([\mathbf{d}] - \mathbf{d}) + ([\mathbf{d}] - \mathbf{d})\frac{\partial^2 r([\mathbf{d}])}{\partial \mathbf{d}^2}([\mathbf{d}] - \mathbf{d})^T + [\hat{w}] \tag{3.4.42}$$

with the gradient and Hessian

$$\frac{\partial r}{\partial \mathbf{d}} = (\mathbf{d}^T\mathbf{d})^{-1/2}\mathbf{d}^T \tag{3.4.43}$$

$$\frac{\partial^2 r}{\partial \mathbf{d}^2} = (\mathbf{d}^T\mathbf{d})^{-1/2}\mathbf{I}_3 - \mathbf{d}(\mathbf{d}^T\mathbf{d})^{-3/2}\mathbf{d}^T \tag{3.4.44}$$

Geometrically, the linearized range observation is the distance from a plane tangent to the sphere centered at UAS i at the point of linearization. While a good approximation at large distances, the increasing curvature of the sphere introduces larger linearization errors as UAS get closer to one another, see figure 3.4.
Evaluating the remainder term of (3.4.42) and forming the interval sum with $[\hat{w}_r]$ provides an interval bound on the total error of the linearized observation equation.
Implementing this generic bounding scheme as introduced in equation (3.4.42) leads however to some overapproximation due to dependency in the involved expressions. The larger the linearization error, the less likely it is that the linearized observations can contribute to contracting the guaranteed state set. That being said, it is desirable to find tight bounds on the linearization error to exploit the maximum amount of information captured by range observations. Explicitly evaluating the error between the nonlinear observation equation (3.4.40) and its linearization, exploiting monotonicity, can however provide tight bounds. The explicit linearization error interval is given by

$$[\Delta r] = (\mathbf{d}^T\mathbf{d})^{1/2} + \tilde{\mathbf{g}}[\mathbf{d}] - \tilde{\mathbf{g}}\tilde{\mathbf{d}} - ([\mathbf{d}]^T[\mathbf{d}])^{1/2} \tag{3.4.45}$$

with $\tilde{\mathbf{g}} = \left.\frac{\partial r}{\partial \mathbf{d}}\right|_{\mathbf{d}=\tilde{\mathbf{d}}}$. Evaluating equation (3.4.45) at the 8 vertices of $[\mathbf{d}]$ provides a sharp inclusion of $[\Delta r]$.

To verify monotonicity of equation (3.4.45) w.r.t. $[\mathbf{d}]$, consider the interval-valued gra-

dient

$$\left[\frac{\partial \Delta r}{\partial \mathbf{d}}\right] = \tilde{\mathbf{g}} - ([\mathbf{d}]^T[\mathbf{d}])^{-1/2}[\mathbf{d}]^T \tag{3.4.46}$$

$$= \frac{\tilde{\mathbf{d}}^T}{||\tilde{\mathbf{d}}||} - \frac{[\mathbf{d}^T]}{|||[\mathbf{d}]|||} \tag{3.4.47}$$

For any $\tilde{\mathbf{d}}, \mathbf{d}$, the elements of the gradient can be made positive over the range of $[\mathbf{d}]$ by a change of coordinates. The fact that monotonicity can be enforced by a simple translation of the frame we are operating in, implies that it suffices to evaluate the vertices of $[\mathbf{d}]$ in the original frame, since a translation does not affect Δr.
See figure 3.4 for two examples. The first one, a separation of about $100\,m$ is representative of the situation when the UAS is approaching the formation from the far field during the rendezvous phase. The second one is representative of keeping a position right at the border of the relative position interval, i.e. as close as possible to the predecessor. Note the drastically increasing linearization error for smaller separations.

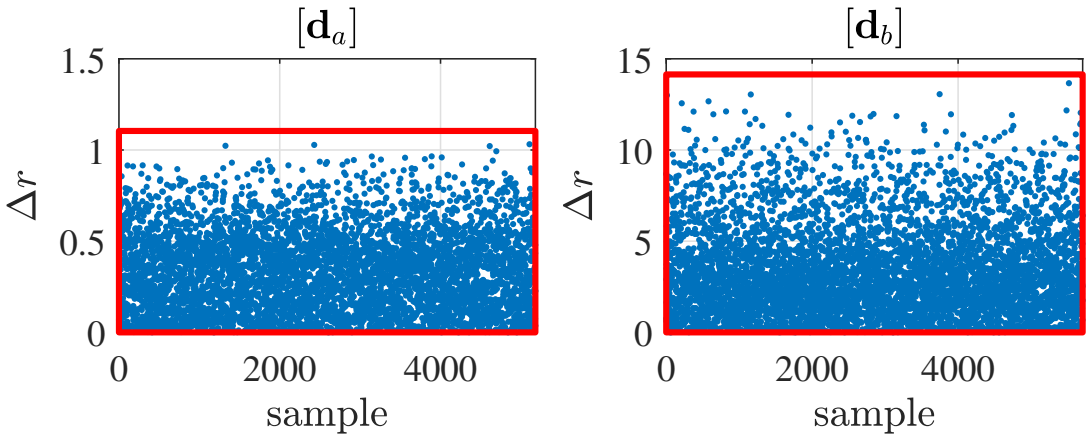

Figure 3.4: Range observation linearization error interval bounds and Monte Carlo samples for two separation vector intervals: $[\mathbf{d}_a] = \begin{pmatrix}[90, 110] & [-10, 10] & [-10, 10]\end{pmatrix}^T$ and $[\mathbf{d}_b] = \begin{pmatrix}[0, 20] & [-10, 10] & [-10, 10]\end{pmatrix}^T$

Using the linearized observation equations and their interval error bounds developed in the previous section, we are now ready to apply linear set membership filtering techniques.

3.5 Modified Extended Set Membership Filter

The various existing set membership filters employing ellipsoidal set representations (see e.g. [85, 97]) follow the common recursive structure introduced in section 3.3, each one having a convenient intuitive geometrical interpretation. In a first step, the guaranteed state set $\Omega(\hat{\mathbf{x}}_{k,k}, \mathbf{\Sigma}_{k,k})$ of the preceding filter step k is propagated using the dynamics model and an ellipsoidal bound $\Omega(\mathbf{0}, \mathbf{Q}_k)$ on the modeling error. This amounts

to forming the Minkowski sum of $\Omega(\hat{\mathbf{x}}_{k,k}, \mathbf{\Sigma}_{k,k})$ and $\Omega(\mathbf{0}, \mathbf{Q}_k)$. This sum generally not being an ellipsoid, an outer ellipsoidal approximation $\Omega(\hat{\mathbf{x}}_{k+1,k}, \mathbf{\Sigma}_{k+1,k})$ is formed minimizing some measure of size.

Algorithms primarily differ in the selection of this measure. The volume, or equivalently the determinant (see [98] for a survey of minimum volume algorithms) as well as the trace of $\mathbf{\Sigma}_{k+,k}$ have been considered. The determinant criterion is reported to potentially produce very thin, long ellipsoids, while the trace criterion, proportional to the sum of squared half axes, tends to lead to better behaved ellipsoids [97].

In a second step, the intersection of $\Omega(\hat{\mathbf{x}}_{k+1,k}, \mathbf{\Sigma}_{k+1,k})$ and an ellipsoidal approximation of the consistent state set is approximated by a bounding ellipsoid $\Omega(\hat{\mathbf{x}}_{k+1,k+1}, \mathbf{\Sigma}_{k+1,k+1})$, again minimizing the trace or determinant of $\mathbf{\Sigma}_{k+1,k+1}$.

These two steps are particularly efficient to implement for linear systems, see eg. [97], exploiting firstly the fact that ellipsoids stay ellipsoids under linear transformations and secondly that the set inversion involved in computing the consistent state set can be performed without approximation by matrix inversion.

In the spirit of the Extended Kalman Filter, the Extended Set Membership Filter proposed in [85] extends ellipsoidal set membership filtering to nonlinear systems by local linearization. Ellipsoidal bounds on the resulting linearization errors are employed and lumped together with the propagation error bounds and measurement error bounds of the nonlinear model. The algorithm is rather generic as the propagation and update step provide sets of approximating ellipsoids, each parametrized by a scalar parameter. In [84, 85] the authors, while mentioning the existence of closed form minimum trace solutions, propose applying numerical optimization to determine the parameter corresponding to the minimum size ellipsoid in each set. Note that the utilization of numerical optimization does not preclude real-time application, since every member of the given sets is guaranteed to contain the propagated set or intersection respectively, so optimization can be stopped anytime. Closed-form solutions are however preferable due to improved efficiency (especially taking into account the still somewhat limited embedded computing resources of UAS) and guaranteed minimization of the selected size criterion. As can be shown, the minimum trace solution presented in [97] can directly be applied to the propagation step. On the other hand, no minimum trace or minimum volume closed form exist solutions for the update step in its form given by [85]. In the following we present a modified, fixed time version of the ESMF, adopting the closed form minimum trace ellipsoid approximations proposed in [97]. A further modification we make to the ESMF is the addition of an input vector acting on both the state transition and outputs. This will prove useful for propagating the relative position by time differenced carrier phase observations and to take the position offset of UWB ranging antennas into account.

3.5.1 Algorithm

We provide first the algorithm in general terms, followed by specifics for relative UAS localization.

Propagation The state transition is given by

$$\mathbf{x}_{k+1,k} = f(\mathbf{x}_{k,k}) + \mathbf{u}_k \tag{3.5.1}$$

$$\mathbf{\Sigma}_{k+1,k} = \mathbf{A}_k \frac{\mathbf{\Sigma}_{k+1,k}}{1-\beta_k} \mathbf{A}_k^T + \frac{\mathbf{Q}_k}{\beta_k} \tag{3.5.2}$$

where $\mathbf{A} = \frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k,k}}$. The parameter β_k is computed following [97] to minimize the trace of $\mathbf{\Sigma}_{k+1,k}$ to

$$\beta_k = \left(\frac{\mathrm{tr}\left(\mathbf{\Sigma}_{k,k}^{1/2}(\mathbf{\Sigma}_{k,k}^{1/2})^T\right)^{1/2} + \mathrm{tr}\left(\mathbf{Q}_{k,k}^{1/2}(\mathbf{Q}_{k,k}^{1/2})^T\right)^{1/2}}{\mathrm{tr}\left(\mathbf{Q}_{k,k}^{1/2}(\mathbf{Q}_{k,k}^{1/2})^T\right)^{1/2}} \right)^{-1} \tag{3.5.3}$$

Update The update of the standard ESMF algorithm is a one-step matrix operation. Although closed-form solutions are mentioned in [85] as an option, numerical optimization has to be employed to minimize the trace of the resulting ellipsoid when implementing the one-step update, see [97]. The alternative update step presented in [97] exploits the fact that a closed form minimum trace solution is possible for the intersection of ellipsoids and strips. Each element of the observation vector with its corresponding error interval defines a pair of hyperplanes - a strip - in state space, in turn defining the consistent space set as all states enclosed by the hyperplanes. By consecutive intersection of the propagated state ellipsoid with each strip, an approximation of the minimum trace ellipsoid intersection is obtained. From the linearized observation equation

$$\begin{pmatrix} y_{k+1,1} \\ y_{k+1,2} \\ \vdots \\ y_{k+1,m} \end{pmatrix} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_m \end{bmatrix} \mathbf{x} + \begin{pmatrix} w_{k+1,1} \\ w_{k+1,2} \\ \vdots \\ w_{k+1,m} \end{pmatrix} \tag{3.5.4}$$

where $\mathbf{C}_i$ are the rows of $\mathbf{C} = \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1,k}}$ and $w_{k+1,i} \in [-W_{k+1,i}, W_{k+1,i}]$[2] we obtain m pairs of hyperplanes defining m strips in $\mathbb{R}^n$.

[2]Note that every asymmetric observation error interval can be converted into a new, centered observation and a corresponding symmetric error interval

If known inputs affect outputs in the known form

$$\tilde{\mathbf{y}}_k = h(\mathbf{x}) + g(\mathbf{u}_k) + \mathbf{w}_k \tag{3.5.5}$$

we define a new output $\mathbf{y}_k = \tilde{\mathbf{y}}_k - g(\mathbf{u}_k)$ allowing us again to write

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}) + \mathbf{w}_k \tag{3.5.6}$$

and to linearize without modifications to obtain equation (3.5.4).
The strips are then recursively intersected with the propagated state ellipsoid $\Omega(\hat{\mathbf{x}}_{k+1,k}, \mathbf{\Sigma}_{k+1,k})$.

Recursive intersection

Define

$$\mathbf{M}_0 = \mathbf{\Sigma}_{k+1,k}^{-1} \tag{3.5.7}$$

$$\mathbf{c}_0 = \hat{\mathbf{x}}_{k+1,k} \tag{3.5.8}$$

For $i = 1...m$

For each element of the observation vector the two hyperplanes are given by

$$\begin{aligned} \text{Plane 1}: \quad \mathbf{C}_i||\mathbf{C}_i||^{-1}\mathbf{x} &= (y_{k+1,i} - W_{k+1,i})||\mathbf{C}_i||^{-1} \\ &= p^+ \\ \text{Plane 2}: \quad \mathbf{C}_i||\mathbf{C}_i||^{-1}\mathbf{x} &= (y_{k+1,i} + W_{k+1,i})||\mathbf{C}_i||^{-1} \\ &= p^- \end{aligned} \tag{3.5.9}$$

with their common unit normal vector $\mathbf{C}_i||\mathbf{C}_i||^{-1}$ and the respective distances to the origin p^+, p^-. As required by the intersection algorithm, we first check for each plane if it intersects the ellipsoid $\Omega(\hat{\mathbf{x}}_{k+1,k}, \mathbf{\Sigma}_{k+1,k})$, and if not, compute a parallel hyperplane that is tangent to the ellipsoid, see[3] appendix A. The resulting tightened hyperplanes being given by

$$\begin{aligned} \text{Tightened plane 1}: \quad \mathbf{C}_i||\mathbf{C}_i||^{-1}\mathbf{x} &= p_t^+ \\ \text{Tightened plane 2}: \quad \mathbf{C}_i||\mathbf{C}_i||^{-1}\mathbf{x} &= p_t^- \end{aligned} \tag{3.5.10}$$

we compute a new tightened observation interval center

$$\hat{y}_{k+1,i} = \frac{1}{2}(p_t^+ + p_t^-)||\mathbf{C}_i|| \tag{3.5.11}$$

[3]Details on computing these tightened hyperplanes might also be given in [99], where tightening has first been introduced. Since [99] appears to be available only in print, we provide a derivation in the appendix for the sake of accessibility

and tightened observation error bounds

$$\overline{W}_{k+1,i} = \frac{1}{2}(p_t^- - p_t^+)||\mathbf{C}_i|| \tag{3.5.12}$$

The corresponding tightened strip has the same intersection with $\Omega(\hat{\mathbf{x}}_{k+1,k}, \mathbf{\Sigma}_{k+1,k})$, but simplifies the intersection algorithm, see [97]. After normalizing with the new error bounds

$$\overline{y}_{k+1,i} = \overline{W}_{k+1,i}^{-1}\hat{y}_{k+1,i} \tag{3.5.13}$$

$$\overline{\mathbf{C}}_i = \overline{W}_{k+1,i}^{-1}\mathbf{C}_i \tag{3.5.14}$$

we obtain a strip

$$\mathbf{S}(\overline{y}_{k+1,i}, \overline{\mathbf{C}}_i) = \{\mathbf{x} : |\overline{y}_{k+1,i} - \overline{\mathbf{C}}_i\mathbf{x}| \leq 1\} \tag{3.5.15}$$

in the form required for the intersection algorithm. The intersection of the ith strip and the intermediate ellipsoid $\Omega(\mathbf{c}_{i-1}, \mathbf{M}_{i-1}^{-1})$ is then computed as follows, adapted from [97]:

$$\begin{aligned}
\mathbf{P} &= \mathbf{M}_{i-1}^{-1} \\
\delta &= \overline{y}_{k+1,i} - \overline{\mathbf{C}}_i\mathbf{c}_{i-1} \\
g &= \overline{\mathbf{C}}_i\mathbf{P}\overline{\mathbf{C}}_i^T \\
\gamma &= \overline{\mathbf{C}}_i\mathbf{P}^2\overline{\mathbf{C}}_i^T \\
\mu &= \mathrm{tr}(\mathbf{P}) \\
\beta_1 &= \frac{3}{g} \\
\beta_2 &= \frac{g(\mu(1-\delta^2)-\gamma) + 2(g\mu - \gamma(1-\delta^2))}{g^2(g\mu-\gamma)} \\
\beta_3 &= \frac{\mu(1-\delta^2)+\gamma}{g^2(g\mu-\gamma)}
\end{aligned}$$

if $\beta_3 > 0$ the intersection does not modify $\Omega(\mathbf{c}_{i-1}, \mathbf{M}_{i-1}^{-1})$, i.e. both tightened hyperplanes are tangent and

$$\begin{aligned}
\mathbf{M}_i &= \mathbf{M}_{i-1} \\
\mathbf{c}_i &= \mathbf{c}_{i-1}
\end{aligned}$$

Otherwise the minimum trace ellipsoid bounding the intersection is computed as

$$
\begin{aligned}
M =& \frac{3\beta_2 - \beta_1^2}{9} \\
N =& \frac{9\beta_1\beta_2 - 27\beta_3 - 2\beta_1^3}{54} \\
\theta =& \operatorname{acos}\left(\frac{N}{(-M^3)^{1/2}}\right) \\
q =& 2(-M)^{1/2}\cos\left(\frac{\theta}{3}\right) - \frac{\beta_1}{3} \\
\alpha^* =& \frac{1}{1+q} \\
\mathbf{M}_{\alpha^*} =& \alpha^*\mathbf{M}_{i-1} + (1-\alpha^*)\overline{\mathbf{C}}_i\overline{\mathbf{C}}_i^T \\
\mathbf{c}_i =& \mathbf{M}_{\alpha^*}^{-1}\left[\alpha^*\mathbf{M}_{i-1}\mathbf{c}_{i-1} + (1-\alpha^*)\overline{\mathbf{C}}_i\overline{y}_{k+1,i}\right] \\
\delta_{\alpha^*} =& \mathbf{c}_{i-1}^T\mathbf{M}_{i-1}\mathbf{c}_{i-1} + (1-\alpha^*)\overline{y}_{k+1,i}^2 - \mathbf{c}_i^T\mathbf{M}_{\alpha^*}\mathbf{c}_i \\
\mathbf{M}_i =& (1-\delta_{\alpha^*})^{-1}\mathbf{M}_{\alpha^*}
\end{aligned}
$$

After the last intersection

$$
\begin{aligned}
\hat{\mathbf{x}}_{k+1,k+1} =& \mathbf{c}_m \\
\mathbf{\Sigma}_{k+1,k+1} =& \mathbf{M}_m^{-1}
\end{aligned}
$$

provide the updated guaranteed state ellipsoid at sample $k+1$.

3.5.2 Application to relative localization

We now apply the modified ESMF to the problem of cooperative relative localization between UAS. In a most basic configuration, each UAS runs a bank of filters, one for each other UAS. Computational effort could further be reduced by exchanging set estimates between UAS. In the following we consider the case of one UAS tracking the relative position of another one.

To initialize the guaranteed state ellipsoid, the maximum error interval of the differential pseudo range position $\tilde{\mathbf{d}}^{\rho,0}$ is computed by interval arithmetic from the least-squares solution of 3.4.6

$$
[\tilde{\mathbf{d}}_\rho] = (\mathbf{H}_0^T\mathbf{H}_0)^{-1}\mathbf{H}_0^T[\nabla^r\rho_0] \tag{3.5.16}
$$

The Minimum Volume Ellipsoid (MVE) enclosing this interval then provides the initial guaranteed state set $\Omega(\hat{\mathbf{x}}_{0,0}, \mathbf{\Sigma}_{0,0})$.

The state comprises the relative position vector between UAS i and UAS j

$$\mathbf{x}_k = \mathbf{d}_{i,j,k} \tag{3.5.17}$$
$$f(\mathbf{x}_k) = \mathbf{d}_{i,j,k} + \Delta\tilde{\mathbf{d}}_k + \mathbf{w}_{\Delta\mathbf{d},k+1} \tag{3.5.18}$$
$$\mathbf{A} = \mathbf{I}_3 \tag{3.5.19}$$

and observations are double differences of pseudo ranges

$$\mathbf{y}_k = \nabla^r \rho_k \tag{3.5.20}$$
$$= \mathbf{H}_{p,r}\mathbf{d}_{i,j,k} + \mathbf{w}_{\nabla^r\rho,k} \tag{3.5.21}$$
$$\mathbf{C} = \mathbf{H}_{p,r} \tag{3.5.22}$$

Note that we use the linearized observation equation here due to the small linearization error. The observation error is bounded by the interval sum of the linearization error interval given by equation (3.4.23) and the interval of double difference pseudo range observation noise obtained from zero-baseline observations. Note that in contrast to the standard ESMF, the interval vector bounds on pseudo range observations (see section 3.4.1) can directly by used and do not need to be approximated by a bounding ellipsoid. The propagation error ellipsoid $\Omega(\mathbf{0}, \mathbf{Q}_k)$ is obtained as the MVE of the interval of carrier phase position propagation errors $\mathbf{w}_{\Delta\mathbf{d},k}$, see equation (3.4.38).

Incorporating UWB ranging

Range observations are integrated into the filter by augmenting the observation vector to

$$\mathbf{y}_k = \begin{pmatrix} \nabla^r \rho_k \\ r_{i,j,k} \end{pmatrix} \tag{3.5.23}$$
$$= \begin{pmatrix} \mathbf{H}_{p,r}\mathbf{d}_k(i,j) + \mathbf{w}_{\nabla^r\rho,k} \\ ||\mathbf{d}_{i,j,k}|| + \bar{w}_{r,k} \end{pmatrix} \tag{3.5.24}$$
$$\mathbf{C} = \begin{bmatrix} \mathbf{H}_{p,r} \\ \frac{\partial r}{\partial \mathbf{d}}\big|_{\mathbf{d}=\hat{\mathbf{x}}_{k,k-1}} \end{bmatrix} \tag{3.5.25}$$

Asynchronous observations If ranging and GNSS observations are available asynchronously at different time instants, ranging observations can be incorporated in a

separate update step using

$$\mathbf{y}_k = r_{i,j,k} \tag{3.5.26}$$

$$= ||\mathbf{d}_{i,j,k}|| + \bar{w}_{r,k} \tag{3.5.27}$$

$$\mathbf{C} = \left.\frac{\partial r}{\partial \mathbf{d}}\right|_{\mathbf{d}=\hat{\mathbf{x}}_{k,k-1}} \tag{3.5.28}$$

It facilitates an informed judgment of the merits of the proposed set membership filter to compare it to its stochastic EKF counterpart, to which we will consequently turn our attention in the following section.

3.6 Extended Kalman Filter

The Extended Kalman Filter appears to be, along with the Unscented Kalman Filter, the most popular state estimator providing a confidence estimate for nonlinear systems. For the sake of comparison, we implemented two EKFs that are the analogue of the ESMF schemes presented in the previous section. The state transition and observation equations for the EKF are identical to those used for the ESMF, except that stochastic instead of deterministic bounds on the prediction errors and observation errors are assumed. Accordingly, $\mathbf{Q}_{KF}$ and $\mathbf{R}_{KF}$ are the corresponding process noise and measurement noise covariance matrices.
For the GNSS-only filter we obtain an estimate of the elements of $\mathbf{R}_{KF}$ by fitting a normal distribution to zero-baseline double difference samples collected with a low-cost GPS receiver [100], see figure 3.5. An approximation of $\mathbf{R}_{KF}$ is then

$$\mathbf{R}_{GNSS} = \sigma^2_{\nabla^r \rho} \mathbf{I}_m \tag{3.6.1}$$

where $\sigma_{\nabla^r \rho}$ is the standard deviation of the fitted normal distribution and m is the number of pseudo range double differences. When incorporating UWB ranging, we approximate the unknown error distribution of the linearized range observation by setting $\sigma_r = 3 * \frac{\bar{w}_{r,k}}{2}$, leading to

$$\mathbf{R}_{GNSSUWB} = \begin{bmatrix} \mathbf{R}_{GNSS} & \mathbf{0} \\ \mathbf{0} & \sigma_r^2 \end{bmatrix} \tag{3.6.2}$$

To approximate $\mathbf{Q}_{KF}$, we first compute an estimate $\mathbf{R}_\Phi$ of the covariance matrix of the carrier-phase TDSD observation noise $\Delta\nabla\epsilon^\Phi_{i,j,k}$ again from zero-baseline observations

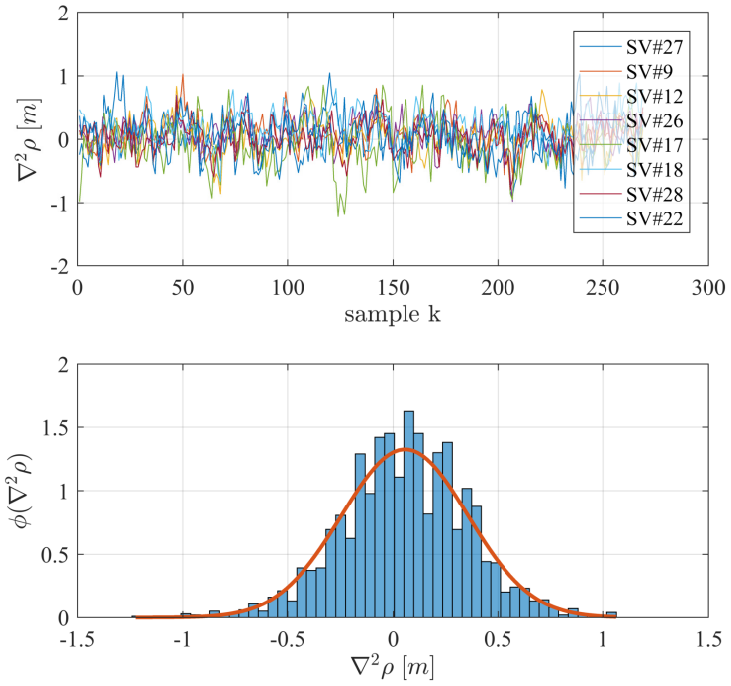

Figure 3.5: Time series and fitted normal distribution of GPS C/A pseudo range double difference noise obtained from zero-baseline measurements (ublox LEA-6T receivers)

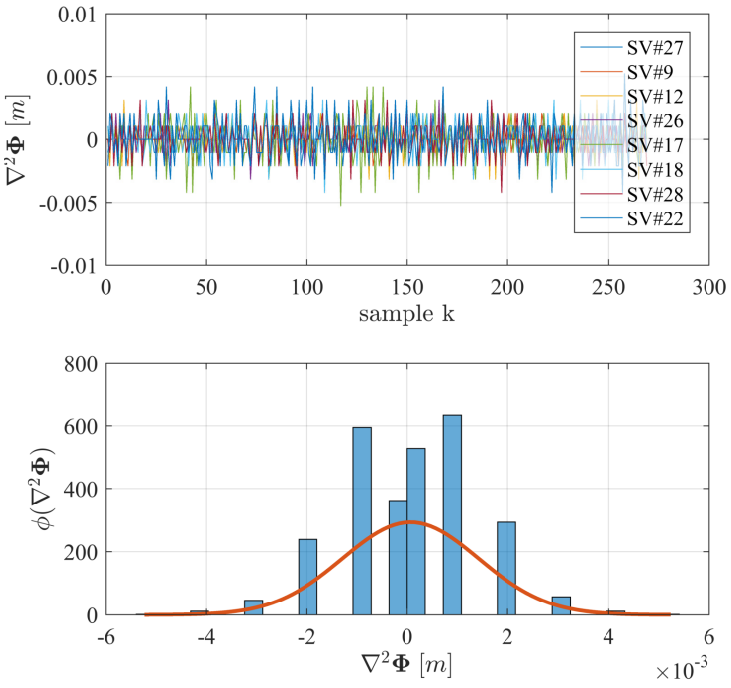

Figure 3.6: Time series and fitted normal distribution of GPS L1 carrier phase time differenced double difference zero-baseline observations (ublox LEA-6T receivers)

(see figure 3.6). The covariance of $\mathbf{w}_k$ can then be computed as

$$(\mathbf{H}_{i,j,k}^T\mathbf{H}_{i,j,k})^{-1}\mathbf{H}_{i,j,k}^T\mathbf{R}_\Phi\left[(\mathbf{H}_{i,j,k}^T\mathbf{H}_{i,j,k})^{-1}\mathbf{H}_{i,j,k}^T\right]^T \tag{3.6.3}$$

Note that the resulting covariance matrix is time-varying due to the time-varying geometry matrix $\mathbf{H}_{i,j}$.

Equipped with these covariance matrices, we follow the standard prediction-correction scheme of the EKF.

3.7 Derivation of trajectory planning constraints

The ellipsoid produced by the modified ESMF constrains the position vector between GNSS antennas. To obtain an ellipsoid that contains all parts of the UAS, we cover the airframe by another ellipsoid $\Omega(\mathbf{0}, \mathbf{P}_{ac})$ centered at the GNSS antenna center and approximate their Minkowski sum by its minimum trace bounding ellipsoid with the same method used in the ESMF propagation step, see equations (3.5.2)-(3.5.3). The result is an ellipsoid

$$\Omega(\hat{\mathbf{x}}, \bar{\mathbf{\Sigma}}) = \Omega(\hat{\mathbf{x}}, \frac{\mathbf{\Sigma}_{k,k}}{1-\beta} + \frac{\mathbf{P}_{ac}}{\beta}) \tag{3.7.1}$$

guaranteed to completely contain the tracked UAS.
The ellipsoid $\Omega(\hat{\mathbf{x}}, \bar{\mathbf{\Sigma}})$ can directly be integrated as a nonlinear constraint into trajectory planning algorithms. It is however convenient to derive a set of constraints linear in the UAS position to enable use of fast LMPC based trajectory planning (see section 2.5.1). We propose computing an approximate convex hull where the number of faces is a design parameter, representing a tradeoff between tightness of fit and complexity of the resulting polyhedron.

Finding an approximate convex hull

We adopt the algorithm proposed in [101] for the computation of an approximate convex hull of a set of points in $\mathbb{R}^n$. As shown below, it can be applied to find an outer polyhedral approximation of ellipsoids as well. The algorithm is briefly outlined here, see [101] for a detailed derivation.
We first define a set of direction unit vectors $\mathbf{n}_1, \mathbf{n}_2 ... \mathbf{n}_N$ that are evenly spread on the unit sphere. One way to achieve this even sampling is by adopting spherical coordinates

$$\mathbf{n}_i^s = \begin{pmatrix} r & \Theta & \phi \end{pmatrix}^T \tag{3.7.2}$$

where r is distance from the origin, Θ the zenith angle and ϕ the azimuth angle. Given a vector of angles $\mathbf{\Theta} = \begin{pmatrix}\theta_1, \theta_2 ... \theta_n\end{pmatrix}$ evenly spaced on the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and $\phi =$

$\left(\phi_1, \phi_2 ... \phi_n\right)$ evenly spaced on $[0, 2\pi]$, a set of unit direction vectors results to

$$\left((1\,\Theta_1\,\phi_1)^T, (1\,\Theta_2\,\phi_2)^T, ...(1\,\Theta_n\,\phi_n)^T\right)$$

Memory consumption can be reduced by sampling one half space of the unit sphere and considering each unit vector and its negative. This is left as an implementation detail, as it messes up notation. After converting back to Cartesian coordinates

$$\mathbf{n}_i = \begin{pmatrix} \sin\Theta_i \cos\phi_i \\ \sin\Theta_i \sin\phi_i \\ \cos\Theta_i \end{pmatrix} \tag{3.7.3}$$

each of these direction vectors $\mathbf{n}_i$ is the normal vector of a plane given by

$$\mathbf{n}_i^T \mathbf{d} = p_i \tag{3.7.4}$$

where $\mathbf{d}$ is a point on the plane.

Application to Ellipsoids Given the guaranteed state ellipsoid $\Omega_k(\hat{\mathbf{x}}_k, \bar{\Sigma}_k)$, for each of the direction unit vectors $\mathbf{n}_i$ we find the $p_{i,t}$ that represents a plane tangent to the ellipsoid and normal to $\mathbf{n}_i$ by applying the method given in appendix A. Each tangent plane forms one of the planes constituting the approximate convex hull. The resulting set of planes can be written as a polyhedron of the form

$$\begin{gathered} \mathbf{A}\mathbf{d} \leq b \\ \begin{bmatrix} \mathbf{n}_1^T \\ \mathbf{n}_2^T \\ \vdots \\ \mathbf{n}_n^T \end{bmatrix} \mathbf{d} \leq \begin{pmatrix} p_{1,t} \\ p_{2,t} \\ \vdots \\ p_{n,t} \end{pmatrix} \end{gathered} \tag{3.7.5}$$

Existing algorithms such as [102] can then be used to derive convex position constraints from the polyhedron 3.7.5.

3.8 Evaluation

To evaluate the proposed localization schemes, we consider two UAS in close proximity. Each UAS is assumed to broadcast its carrier phase and code phase observations as well as to take UWB range measurements. We select a half circle trajectory of relative positions of the follower UAS w.r.t. the predecessor, emulating a follower changing stations inside a formation, a maneuver requiring guaranteed but not overly conservative relative position estimation error bounds for safe but fast execution. Recall that

for tight formation flight we want a filter that converges fast to useful, i.e. not overly conservative outer estimates of a set that is guaranteed to contain the relative position.

Figures of Merit

We consider three figures of merit to compare ESMF and EKF as well as to judge the impact of UWB ranging observations. First, the trace of the respective ellipsoid matrices (For the EKF we consider the 3σ level surface ellipsoid), as it gives a scalar measure of the ellipsoid size and thus its relative conservativeness. Second, the distance between the respective ellipsoid center and the relative position vector, representing the usefulness of the ellipsoid center as point state estimate. Third, plugging the true relative position into equation (3.3.9) provides a measure of distance to the ellipsoid surface indicating whether the guaranteed state set indeed contains the true state at all times.

GNSS observation errors When comparing the EKF, based on a normally distributed, i.e. unbounded process and measurement error model, and the ESMF and BPDMF, based on interval noise bounds, care has to be taken when it comes to simulating observation errors, as the assumed distribution inherently favors one or another filter formulation. While a simulated normal distribution obviously unduly favors the EKF, a uniform distribution is even further from realistic. From inspection of real zero baseline code phase double difference samples, it appears that the noise profile could reasonably be approximated by a biased normal distribution. However, even more realism can be introduced by directly using noise samples from zero-baseline experiments in simulation, thereby removing the need for assumptions about the statistical nature of double difference observation errors. This approach has been adopted here to simulate both differenced carrier phase as well as differenced code phase observations.

UWB ranging observation errors Our field experiments [103] with a pair of UWB ranging devices (DecaWave EVK1000 [104]) indicate a range error interval excluding antenna position offsets of $w_r \in [-0.5\,m, 0.5\,m]$ in unobstructed outdoor conditions. Larger error bounds reported in [105] are most likely due to the considered metal-rich industrial indoor environment that is not representative of the conditions expected within a formation of UAS made mostly of composite material. For simulation we draw observation errors from a uniform distribution.

Satellite constellation A static constellation of 7 satellites on a sphere corresponding to GPS orbits, constrained by a minimum elevation of 15° is simulated.

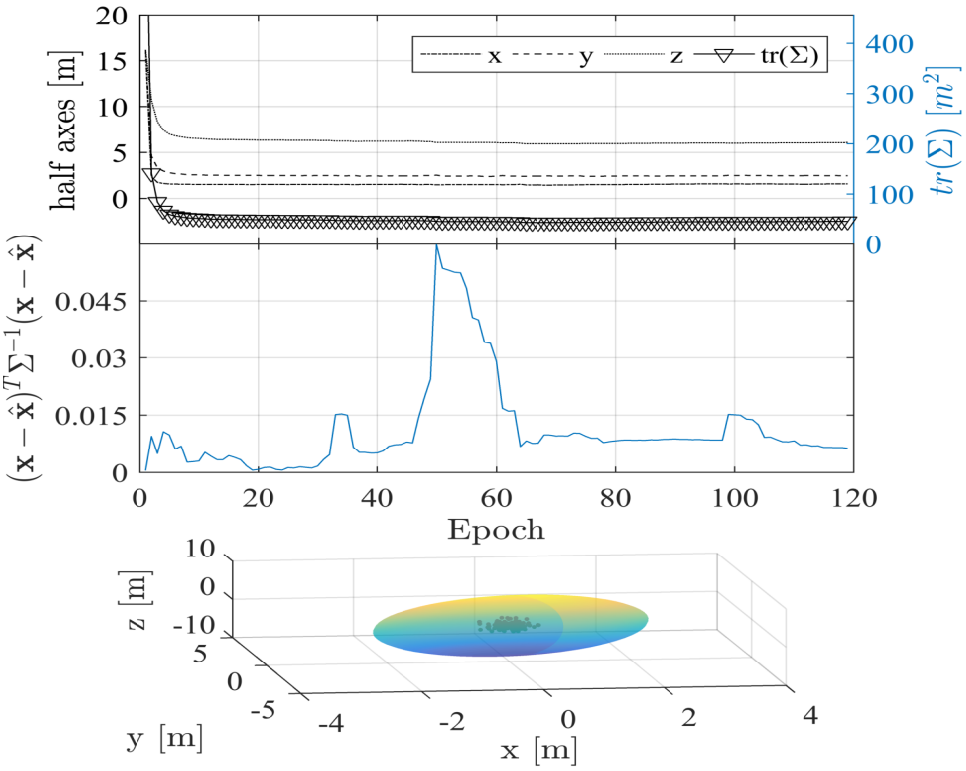

Figure 3.7: ESMF, GNSS only: half axes of guaranteed state set, ellipsoid membership of true state, guaranteed state set at last epoch with pseudo range positions and ESMF position estimates

3.8.1 GNSS only

We first run the ESMF and EKF filter formulations with GNSS observations only. Figure 3.7 shows how the ESMF guaranteed state ellipsoid fulfills its primary property and at all epochs contains the true relative position. The ellipsoid contracts drastically over the first few epochs to a quasi steady state in all three axes with a largest half axis of about 6 m. In comparison, the EKF 3σ level set ellipsoid is by roughly one order of magnitude more optimistic, freeing up considerably more maneuvering space. The ellipsoid contains however from about epoch 45 on no longer the true relative position, see figure 3.9, illustrating the fundamental limitation of Kalman filtering for this application. Note that the EKF estimates an inconsistent confidence ellipsoid in spite of differential pseudo range noise being well approximated by a zero mean normal distribution, i.e. although operating under rather favorable conditions.

While both the EKF mean state estimate and the ESMF ellipsoid center approximate the true relative position with sub-meter accuracy (see figure 3.9), the EKF provides a smoother estimate.

Note that fluctuations in filter execution times (an interpreted Matlab implementation running on a 4-core i7 CPU on Windows 7 OS) can be attributed to the non-real-time execution environment.

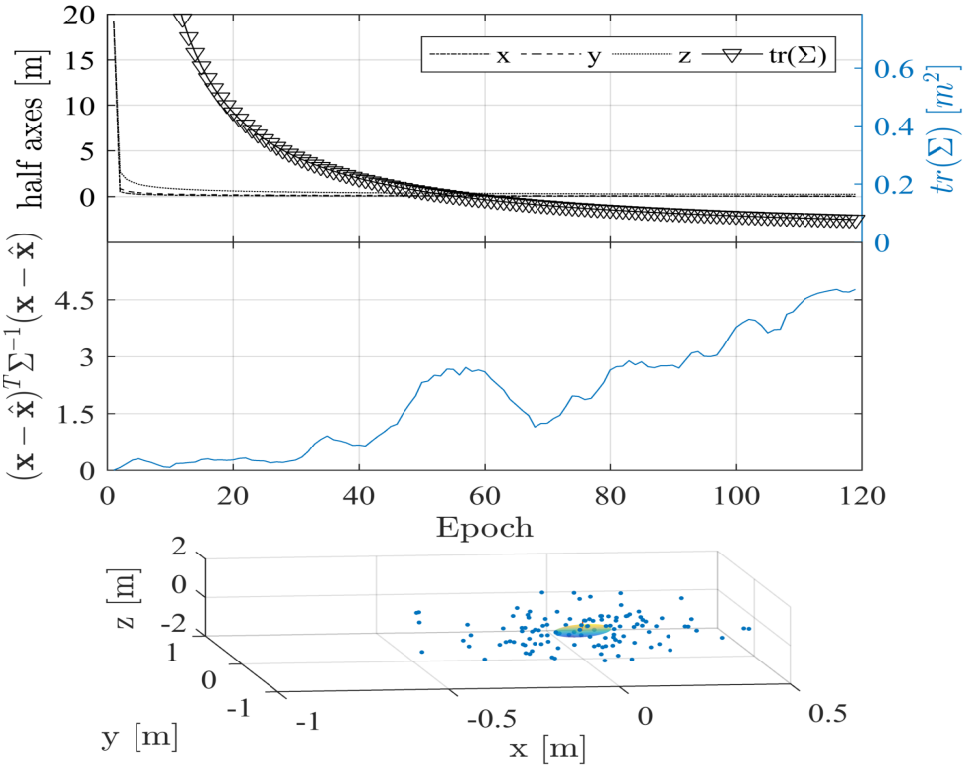

Figure 3.8: EKF, GNSS only: half axes of ellipsoid $\Sigma_{3\sigma}$, ellipsoid membership of true state for $\Sigma_{3\sigma}$, $\Sigma_{3\sigma}$ at last epoch with pseudo range positions and KF position estimates

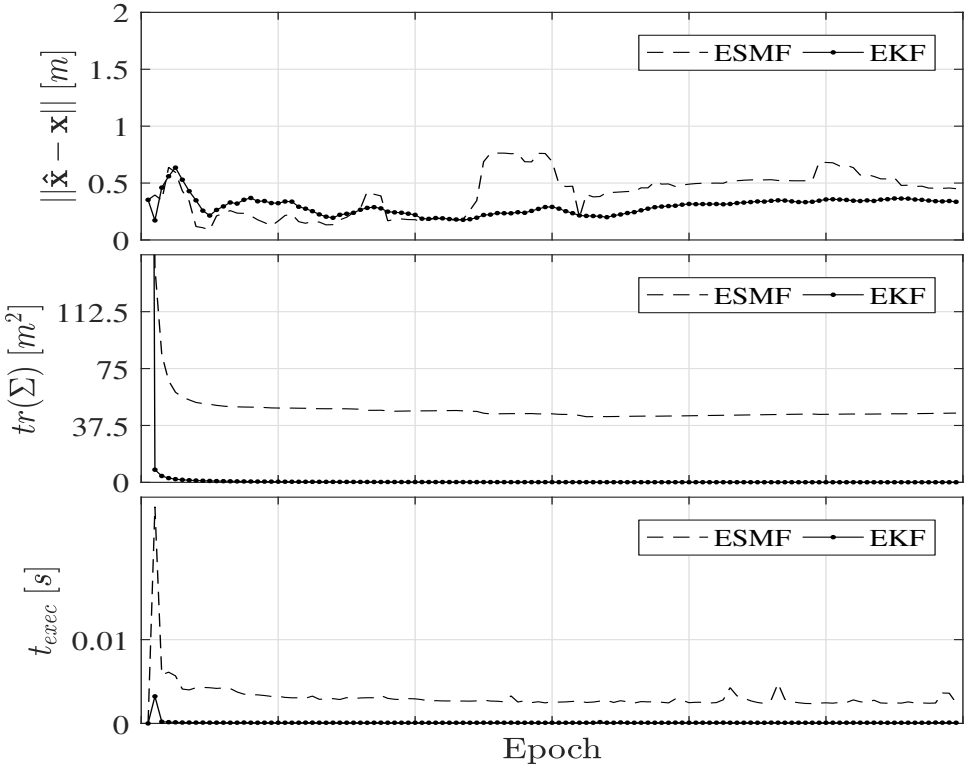

Figure 3.9: GNSS only, comparison of ESMF, EKF: norm of center estimation error, size of guaranteed set (ESMF) or 3σ level set (EKF) respectively as measured by matrix trace, filter execution time

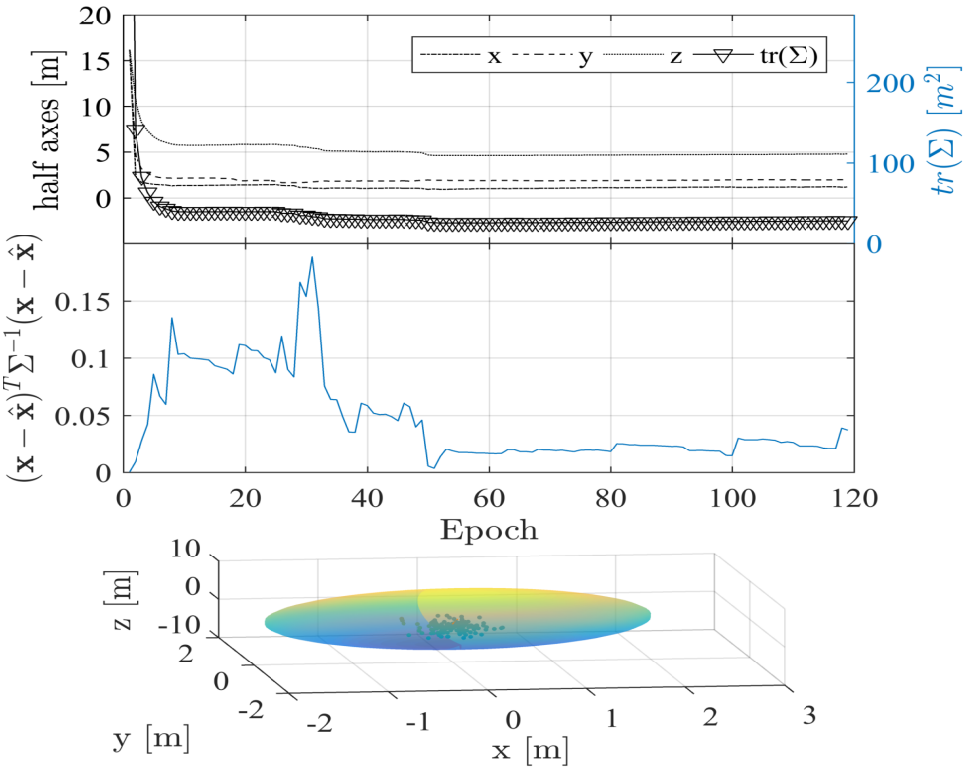

Figure 3.10: ESMF, GNSS+UWB ranging: half axes of guaranteed state set, ellipsoid membership of true state, guaranteed state set at last epoch with pseudo range positions and ESMF position estimates

3.8.2 Benefits of inter-UAS ranging

Incorporating inter UAS range observations leads to tighter guaranteed state ellipsoids as well as reduced EKF covariances, see figs. 3.10, 3.11. Both point position estimates are improved after larger initial transient errors, see figure 3.12. The superior point estimation accuracy of the EKF could be exploited by using it for guidance purposes while relying on the confidence ellipsoid of the ESMF for guaranteed collision avoidance.
The benefit of incorporating ranging observations is more clearly illustrated by figure 3.13, displaying the ratio of the ESMF *a posteriori* half axes and trace without and with ranging observations. A reduction in size as measured by the ellipsoid trace of roughly 30% can be observed compared to the GNSS only case.

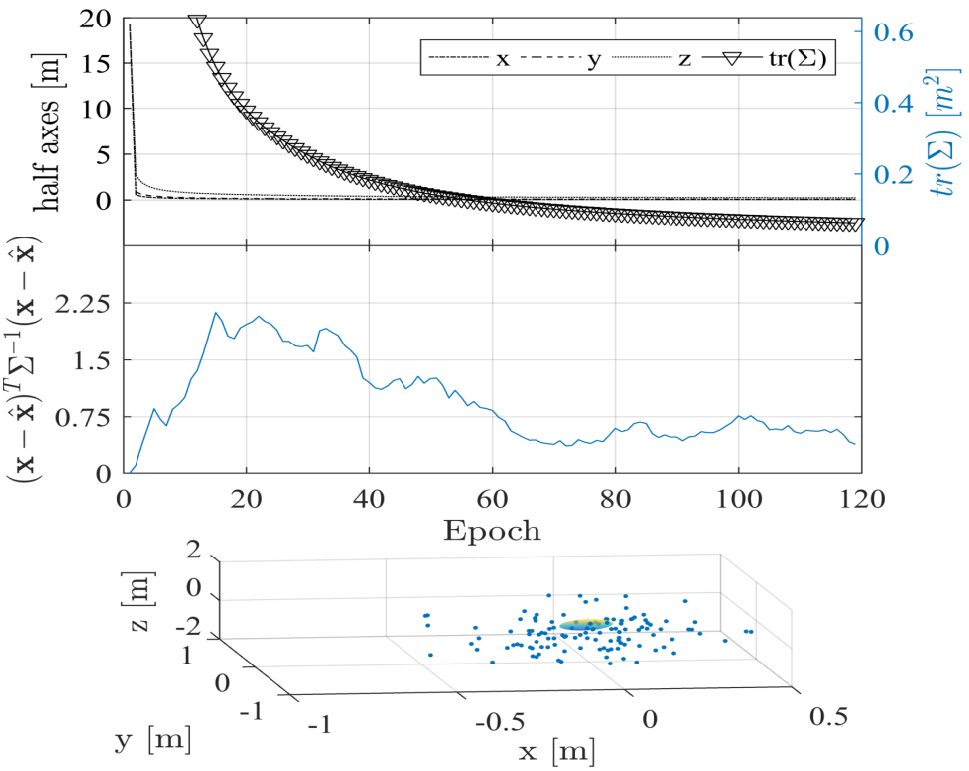

Figure 3.11: EKF, GNSS + UWB ranging: half axes and trace of ellipsoid $\Sigma_{3\sigma}$, membership of true state in $\Omega(\hat{\mathbf{x}}_{k,k}, \Sigma_{3\sigma,k,k}, \Sigma_{3\sigma}$ at last epoch with pseudo range positions (blue dots) and EKF position estimates (red line)

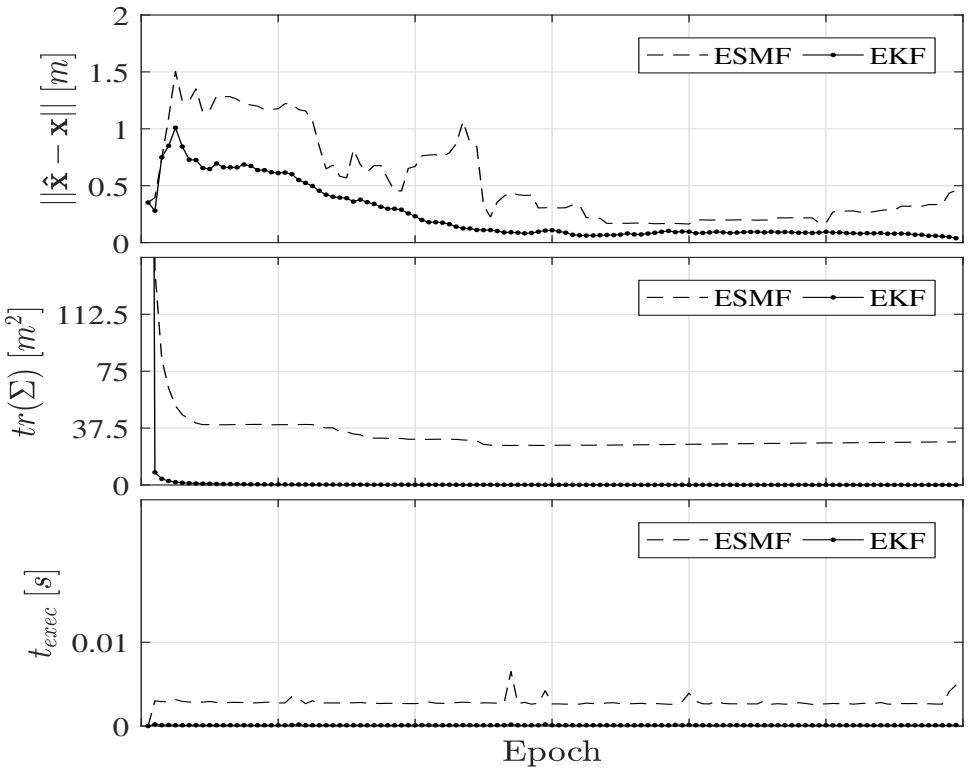

Figure 3.12: GNSS + UWB, comparison of ESMF, EKF: norm of state estimation error, volume of guaranteed set (ESMF) or 3σ set (EKF) respectively, filter execution time

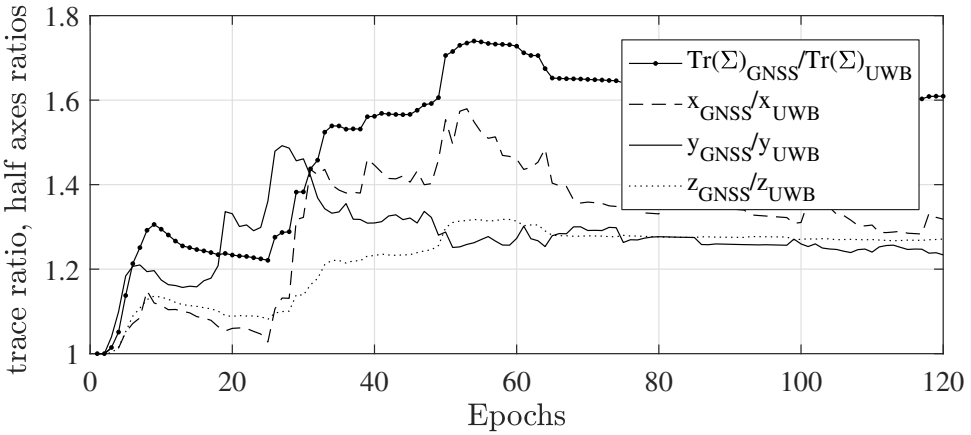

Figure 3.13: ESMF: Ratios of ellipsoid volume and half axes with and without UWB ranging

3.8.3 Fixed-time intersection suboptimality

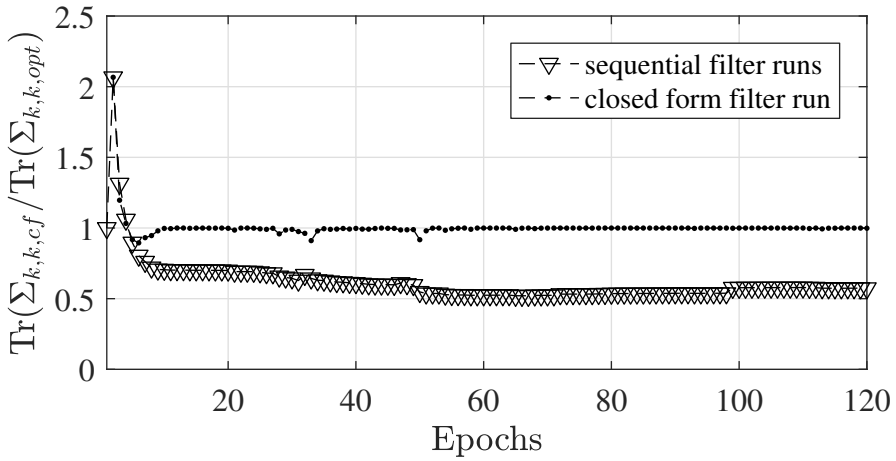

Figure 3.14: Suboptimality of element-wise ESMF update

As mentioned in [97], incorporating observations recursively in the update step of the modified ESMF can increase conservativeness of the resulting *a posteriori* ellipsoid. On the other hand, overbounding the observation error interval by an ellipsoid is likely to increase the conservativeness of the standard ESMF update step compared to the recursive intersection that directly operates on the observation error interval.
We consider two measures to study these opposing effects quantitatively. Firstly, when running the modified ESMF, we compute the minimum trace *a posteriori* ellipsoid by numerical optimization in parallel with the closed-form algorithm, see figure 3.14. Interestingly, during the initial contraction both an improvement in *a posteriori* ellipsoid size and increased conservativeness can be observed. Differences diminish over time and after filter convergence both methods are roughly on par.
Of more consequence is the size of the converged *a posteriori* ellipsoid. Running the ESMF in both configurations and forming the ratio of the trace of the resulting ellipsoid trajectories, we observe steady state ellipsoids roughly twice as large (as measured

by the trace criterion) when using numerical optimization. Note however that a factor of two in the trace size criterion translates into a factor of $\sqrt{2}$ considering lengths of half axes.
Still, this snapshot result suggests that if observation errors are defined by intervals, the overapproximation by ellipsoids when applying the standard ESMF leads to significantly more conservative set estimates, on top of the performance penalty of numerical optimization. It suggests also that the effect of ellipsoid bounding of observation error intervals outweighs the effect of recursive strip intersection.

3.8.4 Overview

We have seen in this chapter how ellipsoidal sets that are guaranteed to contain the relative positions of other UAS can be efficiently computed with a modified version of the Extended Set Membership Filter, relying on low-cost GNSS receiver and UWB ranging hardware. The performed simulations suggest that the resulting algorithm is suited for simultaneously tracking multiple UAS in real-time due to its moderate computational cost. We have demonstrated how the resulting ellipsoidal sets can be outer bounded by polyhedrons to render them useful for path planning algorithms requiring linear position constraints.

Conclusion and future work

Tight formation flight is an enabling technology for a large number of range enhancing techniques such as upwash exploitation, aerial recharging and aerial docking between UAS.
The corresponding major technological challenges lie in reliably and accurately maintaining relative position in the wake of another aircraft, while guaranteeing collision avoidance. In this work we address both of these challenges. We present important building blocks that are expected to be of help in taking tight formation flight from the experimental stage and make it a safe and routine technique.
We have investigated the critical sensitivity of previously proposed continuous time sliding mode guidance techniques w.r.t. two often neglected properties of real-world aircraft: discrete sampling and inner control loop dynamics. We show how these implementability issues can be overcome by extending established discrete sliding mode techniques. The proposed laws are entirely defined by hard input constraints, an advantage over existing DMSC laws. We show that both existing DSMC control as well as the proposed PDSMC control can be posed as special cases of model predictive control
To complement the proposed novel guidance laws, we present a design procedure based on model predictive control techniques to generate minimum-time constrained sliding surfaces. By tightening input and position constraints, these sliding surfaces allow for taking dynamic limitations of the UAS and bounded control errors into account simultaneously to guarantee closed-loop constraint satisfaction.
We present a set membership localization algorithm that enables for the first time to give safety guarantees when it comes to relative positioning between aircraft. The presented set membership filtering scheme provides both guaranteed regions and point estimates of the relative position between UAS. It can be employed in a complementary fashion parallel to possible existing point localization algorithms such as the Extended Kalman Filter. We provide a derivation of linearized observation equations and the corresponding interval inclusions of modeling error bounds. In extension of the standard ESMF algorithm, we take not only linearization errors, but also parametric errors into account that are commonly neglected in differential GNSS positioning. The algorithm can readily be applied to manned aircraft, where safety guarantees are key to making commercial manned tight formation flight a reality in the future. Incorporating UWB

ranging observations is a first step towards GNSS-independent positioning. Guaranteed localization is complemented by the predictive sliding surfaces employed in this work, allowing to take bounded position tracking errors into account.
As a distinctive property, all the presented algorithms are readily implementable using the avionics hardware of a typical small low-cost UAS.

Beyond the pressing need for experimental verification, the work presented here has opened up numerous directions for future exciting work.

Future research

Simulation

- Available atmospheric turbulence models such as the Dryden model currently used by the FFB benchmark do not allow taking into account the spatial distribution of turbulence in a formation. Due to the very small distances between UAS in formation flight, turbulence acting on each UAS can be expected to be highly correlated. As a result, the effect of atmospheric turbulence is likely to be overestimated by current simulations. Efforts should be made towards a multiple-aircraft turbulence model.

Heterogeneous formations

- In this work, formations of identical UAS have been considered. Heterogeneous formations have distinct advantages for certain missions. Examples are search and rescue missions in cluttered environments, where a group of low-flying small UAS can explore an area more efficiently than a single larger UAS at high altitude. Tight formation flight holds potential in this context for two reasons. First, due to their stronger upwash, larger UAS could improve the endurance of following smaller UAS, leading to better overall formation endurance. Secondly, larger UAS could carry batteries and recharge smaller UAS in air, again improving overall group endurance. The stronger wake of larger UAS will make avoiding certain regions of the wake necessary. The constrained sliding surface presented in this work can readily accommodate wake region constraints.

Scalability

- As is illustrated in section 2.6, rigid formations face scalability problems when it comes to maneuvering. This issue should be treated in a global manner in a future research effort. Two lines of attack appear promising. Firstly, for a given formation, new constraints could be introduced into trajectory planning to ensure indi-

vidual constraints satisfaction on the level of each formation member. Secondly, dynamically modifying the formation shape during maneuvering could enable even large formations to fly on trajectories designed to be feasible for one UAS. To give one example, by warping a V-formation into a stacked vertical formation, a horizontal turn feasible for one UAS becomes feasible for a formation of arbitrary size.

PDSMC guidance and load factor control

- The PDSMC guidance law has been implemented on the vertical channel of the FFB benchmark. It is the next step to apply PDSMC control to the coupled vertical/lateral channels, followed by experimental evaluation.

- The proposed PDSMC guidance law relies on sufficiently accurate local prediction models of the load factor and bank angle tracking loops. Adaptive control techniques could improve the proposed guidance schemes in two ways. Firstly, indirect adaptive control techniques, i.e. online learning could be employed to learn better models of the inner loop dynamics. Secondly, adaptive control on the inner loop level could be employed to track reference models corresponding to the PDSMC prediction models. The second option appears more promising, since it alleviates at once the need to accurately identify an aerodynamic model currently required for LQ tracking law synthesis.

- Plasma actuation, while still at an early development stage (about TRL 3 as first wind tunnel [106] and experimental results for UAS actuation are reported) allows for very fast flow manipulation [107]. Once this technology has become more mature, it can remove the main obstacle, slow input dynamics, for the application of continuous time higher order sliding mode control, with its distinct advantages outlined in chapter 2, and continous time HOSM should be revisited for the vertical and possibly the lateral channel.

- The PDSMC optimization problem 2.5.52 could be posed employing a prediction model sampled at the sampling time of the inner loops, imposing the same σ contraction constraint on the last state, enlarging the degrees of freedom of the control. It is considered a worthy future study to determine the degree of performance loss due to the limitation to constant control inputs over the contraction horizon.

- In this work estimated load factors are assumed to be available. Accurate 3D load factor estimation with the consumer-grade MEMS sensors found on todays small low-cost UAS is an interesting problem in itself and deserves future study.

Localization

- In evaluating the set membership localization filter presented in chapter 3, no effort has been made to optimize the trajectory of the follower, i.e. by maximizing some observability measure. Including the contraction of the guaranteed state set into the design of the approach trajectory, based on some kind of observability metric (see e.g. [108] considering the Fisher information matrix) contraction time as well as final size of the guaranteed state set could be systematically minimized. Consider also the QP based approach for bearings-only observability optimized approach trajectories [109] that appears to be a good candidate to be adapted to the set membership localization scheme considered in this work.

Appendix A

Computing tangent hyperplanes of ellipsoids

Given an ellipsoid and a plane

$$\begin{aligned} E =& \Omega(\mathbf{x}_c, \Sigma) \\ \mathbf{n}^T\mathbf{x} =& p \end{aligned}$$

with the plane normal vector $\mathbf{n}$, we check first whether the plane intersects the ellipsoid and, if not, find a modified p_t that represents a plane parallel to the original one and tangent to the ellipsoid.
To considerably simplify doing so, we first apply a linear transform to the problem space that distorts the ellipsoid back into a unit sphere, find the tangent plane in the distorted space and transform it back to the original space.
The ellipsoid being a unit ball under a linear transform, the inverse transform is given by

$$T = \Sigma_k^{-1/2} \tag{A.0.1}$$

The transformed ellipsoid becomes the unit sphere

$$E' = \Omega(T\mathbf{x}_c, \mathbf{I}_3) \tag{A.0.2}$$

$$E' = \Omega(\mathbf{x}'_c, \mathbf{I}_3) \tag{A.0.3}$$

where $()'$ denotes a vector in the transformed space, and the plane corresponding to the unit vector $\mathbf{n}$ in transformed space is given by

$$\frac{\mathbf{n}^T T^{-1}}{||\mathbf{n}^T T^{-1}||}\mathbf{x}' = \frac{p}{||\mathbf{n}^T T^{-1}||} \tag{A.0.4}$$

$$(\mathbf{n}')^T\mathbf{x}' = p' \tag{A.0.5}$$

In the transformed space the distance of the initial plane from the surface of the sphere is straightforwardly given by

$$d = \mathbf{n}'\hat{\mathbf{x}}' - p' \tag{A.0.6}$$

By replacing p' by an appropriately modified p'_t the plane is made tangent to the sphere. The tangent plane is then transformed back to the original space by T^{-1}

$$\frac{\mathbf{n}^T T}{||\mathbf{n}^T T||}\mathbf{x} = \frac{p'_t}{||\mathbf{n}^T T||} \tag{A.0.7}$$

$$\mathbf{n}^T\mathbf{x} = p_t \tag{A.0.8}$$

Appendix B

Disturbance propagation under local SMC

String stability and its 3D generalization mesh stability are a feature of a three-dimensional formation of vehicles that allow separation errors to stay locally contained. It is a well known fact ([110]) that formations under linear control with local feedback are mesh unstable.
Very few solutions based on nonlinear techniques such as Port-Hamiltonian systems, [111] and sliding mode control, [25, 30], have been proposed so far to tackle this problem and no experimental results are known to the authors. Furthermore, the existing approaches have been applied to very simplified models, basically point-mass approximations, neglecting a large part of the complex dynamics and constraints of a fixed-wing UAS.
While in ideal sliding mode, the position error dynamics are by definition confined to the sliding surface, independently of adjacent separation errors, implying mesh stability if the system can be kept in ideal sliding mode. It is this insensitivity towards bounded disturbances that makes sliding mode control a very interesting candidate for mesh stable formation control with local state information.
In the following we show that even under conditions allowing for ideal sliding mode (continuous time system, no input dynamics), bounds on control inputs required to maintain motion on the sliding surface grow downstream, making SMC unsuitable for large formations with local feedback.

1D mesh stability in ideal sliding mode In ideal sliding mode after the reaching phase ($t > t_{reach}$), the system states are confined to the sliding surface and exogenous disturbances completely compensated for, given that they remain within the design bounds. Consider the simplified vertical motion of a pair of two UAS i and $i+1$ in the

sliding phase ($\sigma = 0$), where

$$\Delta\dot{p}_i = \Delta v_i \tag{B.0.1}$$

$$\Delta\dot{v}_i = a_{i+1} - a_i \tag{B.0.2}$$

$$\sigma = \sigma((\Delta p_i\, \Delta v_i)^T) \tag{B.0.3}$$

then $\dot{\sigma} = 0$ provides the equivalent control applied by the predecessor $i+1$

$$0 = \frac{\delta\sigma}{\delta\Delta p_i}\Delta v_i + \frac{\delta\sigma}{\delta\Delta v_i}(a_{i+1} - a_i) \tag{B.0.4}$$

$$a_{i+1} = a_i - \frac{\delta\sigma}{\delta\Delta v_i}^{-1}\frac{\delta\sigma}{\delta\Delta p_i}\Delta v_i \tag{B.0.5}$$

By the triangle inequality we obtain an upper bound for the equivalent control (the acceleration of the follower vehicle, a_{i+1} required to completely reject disturbances and maintain sliding motion) as

$$||a_{i+1}||_\infty \leq ||a_i||_\infty + ||\frac{\delta\sigma}{\delta\Delta v_i}^{-1}\frac{\delta\sigma}{\delta\Delta p_i}\Delta v_i||_\infty \tag{B.0.6}$$

where $||v||_\infty$ denotes $\max_{t\geq 0}(||v(t)||)$. The first term on the right side of equation (B.0.5) represents the input component compensating disturbances while the second term drives the compensated dynamics along the sliding surface. The second component represents the control input generated by the nominal controller, since without disturbances integral sliding mode is already induced by the nominal controller. In ideal sliding mode, the system states can be made to reach the origin and stay confined to it in finite time by the right choice of sliding surface , i.e. $\Delta p_i(t) = \Delta v_i(t) = 0$ for $t > t^* > t_{reach}$. We denote the time interval $t^* > t > t_{reach}$ the state convergence phase. Note that, during the state convergence phase, upper bounds on the accelerations the follower has to exert to maintain sliding motion exceed those of the predecessor. Since the acceleration of the follower enters the next system downstream as disturbance, by recursion this implies that norm bounds on the acceleration demands in a chain of vehicles in sliding mode grow with the vehicle index during the state convergence phase. In quasi sliding mode, the system stays in a reaching phase inside the boundary layer at all times, therefore this issue affects quasi sliding mode in an even more severe fashion.
We illustrate this issue by a simple simulation example. Consider a chain of double integrator UAS models. An integral sliding surface designed by applying LQ control results to

$$\sigma = \mathbf{G}\left(\mathbf{x} - \int_0^t \mathbf{A}_r\mathbf{x}_r(\tau)d\tau\right) \tag{B.0.7}$$

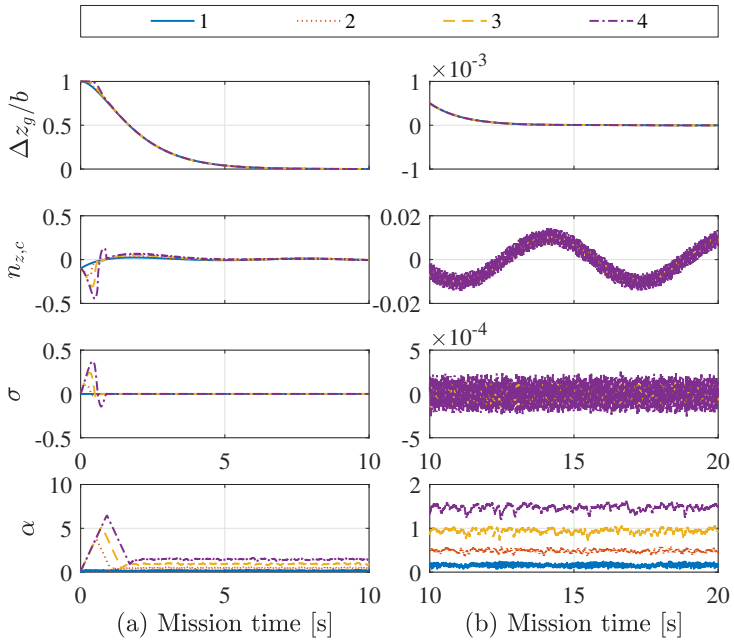

Figure B.1: Chain of 5 vehicles, identical sliding surfaces, $T_s = 10^{-4}s$

with $x = (\Delta p_i \ \Delta v_i)^T$, $x_r = (\Delta p_{i,r} \ \Delta v_{i,r})^T$ and $\mathbf{x}_r(t = 0) = \mathbf{x}(t = 0)$ and where the reference closed loop dynamics matrix $\mathbf{A}_r$ results from applying the LQ state feedback to the nominal disturbance free system.

$$\mathbf{A}_r = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{K}_x \right) \tag{B.0.8}$$

The projection matrix $\mathbf{G}$ fulfills the usual conditions to ensure asymptotic convergence to the reference state in sliding mode. Applying identical adaptive super twisting controllers [112] to each subsystem and simulating a chain of 5 vehicles, where the first vehicle is excited by a periodic disturbance, we observe two phenomena. Firstly, during the state convergence phase, control inputs grow with the vehicle index. Secondly, while the first subsystem maintains at least first order sliding mode (we check only $\sigma = 0$ here) from $t = 0$ on, the following vehicles cannot maintain sliding mode and enter a reaching phase, exhibiting control inputs growing with vehicle index as well.

Bibliography

[1] R. E. Gill, T. L. Tibbitts, D. C. Douglas, C. M. Handel, D. M. Mulcahy, J. C. Gottschalck, N. Warnock, B. J. McCaffery, P. F. Battley, and T. Piersma, "Extreme endurance flights by landbirds crossing the pacific ocean: ecological corridor rather than barrier?," Proceedings of the Royal Society of London B: Biological Sciences, vol. 276, no. 1656, pp. 447–457, 2009.

[2] K. Swider-Lyons, M. Schuette, R. Stroman, J. Rodgers, G. Page, and J. Mackrell, "Liquid hydrogen fuel system for small unmanned air vehicles," in 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, p. 467, 2013.

[3] P. Oettershagen, "High-Fidelity Solar Power Income Modeling for Solar-Electric UAVs: Development and Flight Test Based Verification," arXiv preprint arXiv:1703.07385, 2017.

[4] M. Stolle, J. Bolting, C. Döll, and Y. Watanabe, "A vision-based flight guidance and navigation system for autonomous cross-country soaring UAVs," in Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, pp. 109–117, IEEE, 2015.

[5] H. Weimerskirch, J. Martin, Y. Clerquin, P. Alexandre, and S. Jiraskova, "Energy saving in flight formation," Nature, vol. 413, pp. 697–698, oct. 2001.

[6] M. Beukenberg and D. Hummel, "Aerodynamics, performance and control of airplanes in formation flight," in ICAS, Congress, 17 th, Stockholm, Sweden, Proceedings., vol. 2, pp. 1777–1794, 1990.

[7] V. Jake, R. Ray, K. Ennix, and K. Walsh, "F/A-18 Performance Benefits Measured During the Autonomous Formation Flight Project," tech. rep., NASA/TM-2002-210732, Sept, 2003.

[8] J. Pahle, D. Berger, M. W. Venti, J. J. Faber, C. Duggan, and K. Cardinal, "A preliminary flight investigation of formation flight for drag reduction on the C-17 aircraft," NASA technical report, 2012.

[9] D. Hummel, "Aerodynamic aspects of formation flight in birds," Journal of theoretical Biology, 1982.

[10] S. Park, Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles. PhD thesis, MIT, 2004.

[11] Y. Gu, B. Seanor, G. Campa, M. R. Napolitano, L. Rowe, S. Gururajan, and S. Wan, "Design and flight testing evaluation of formation control laws," Control Systems Technology, IEEE Transactions on, vol. 14, no. 6, pp. 1105–1112, 2006.

[12] C. Rice, Y. Gu, H. Chao, T. Larrabee, S. Gururajan, M. Napolitano, T. Mandal, and M. Rhudy, "Control performance analysis for autonomous close formation flight experiments," in 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 2014.

[13] S. Park, J. Deyst, and J. How, "A new nonlinear guidance logic for trajectory tracking," in AIAA guidance, navigation, and control conference and exhibit, p. 4900, 2004.

[14] D. B. Wilson, A. H. Göktogan, and S. Sukkarieh, "Vision-aided guidance and navigation for close formation flight," Journal of Field Robotics, vol. 33, no. 5, pp. 661–686, 2016.

[15] C. Rice, Y. Gu, H. Chao, T. Larrabee, S. Gururajan, M. Napolitano, T. Mandal, and M. Rhudy, "Autonomous close formation flight control with fixed wing and quadrotor test beds," International Journal of Aerospace Engineering, vol. 2016, 2016.

[16] D. Galzi and Y. Shtessel, "Formation flight strategy and control using higher order sliding mode," in AIAA Guidance Navigation and Control Conference, vol. 4, pp. 4100–4125, 2007.

[17] T. Yamasaki, S. Balakrishnan, and H. Takano, "Sliding mode based integrated guidance and autopilot for chasing UAV with the concept of time-scaled dynamic inversion," in Proceedings of the 2011 American Control Conference, pp. 1598–1603, IEEE, 2011.

[18] D. Galzi and Y. Shtessel, "Unmanned rotorcraft tight formation flight control using sliding mode control driven by sliding mode disturbance observers," in AIAA guidance, navigation and control conference and exhibit, Honolulu, Hawaii, pp. 18–21, 2008.

[19] T. Yamasaki, S. Balakrishnan, and H. Takano, "Separate-channel integrated guidance and autopilot for a path-following UAV via high-order sliding modes," in Proc of the AIAA Guidance, Navigation, and Control Conference, 2012.

[20] D. Galzi and Y. Shtessel, "Closed-coupled formation flight control using quasi-continuous high-order sliding-mode," in 2007 American Control Conference, pp. 1799–1804, IEEE, 2007.

[21] S. Rao and D. Ghose, "Sliding mode control-based autopilots for leaderless consensus of unmanned aerial vehicles," IEEE transactions on control systems technology, vol. 22, no. 5, pp. 1964–1972, 2014.

[22] J. N. Gross, Y. Gu, and M. B. Rhudy, "Robust UAV relative navigation with DGPS, INS, and peer-to-peer radio ranging," IEEE Transactions on Automation Science and Engineering, vol. 12, no. 3, pp. 935–944, 2015.

[23] W. R. Williamson, M. F. Abdel-Hafez, I. Rhee, E.-J. Song, J. D. Wolfe, D. F. Chichka, and J. L. Speyer, "An instrumentation system applied to formation flight," IEEE transactions on control systems technology, vol. 15, no. 1, pp. 75–85, 2007.

[24] J. Bolting, J.-M. Biannic, and F. Defay, "Structured control law design and robustness assessment for the automatic launch of small UAVs," in Proceedings of the CEAS EuroGNC conference, 2015.

[25] J. Bolting, S. Fergani, J.-M. Biannic, F. Defay, and M. Stolle, "Discrete sliding mode control of small UAS in tight formation flight under information constraints," in Proceedings of IFAC Symposium on Automatic Control in Aerospace (ACA) 2016, 2016.

[26] J. Bolting, M. Stolle, S. Fergani, J.-M. Biannic, and F. Defay, "An open benchmark for distributed formation flight control of fixed-wing unmanned aircraft systems," in Proceedings of the 20th IFAC World Congress, 2017.

[27] M. Stolle, Y. Watanabe, C. Döll, and J. Bolting, "Vision-based lifespan and strength estimation of sub-cumulus thermal updrafts for autonomous soaring," in Unmanned Aircraft Systems (ICUAS), 2016 International Conference on, pp. 162–169, IEEE, 2016.

[28] J. Bolting, J.-M. Biannic, and F. Defay, "Set membership localization for safe tight formation flight," submitted to Robotics and Autonomous Systems.

[29] J. Bolting, "Formation Flying Benchmark." https://github.com/jtec/ffb, 2016.

[30] D. Galzi and Y. Shtessel, "UAV formations control using high order sliding modes," in American Control Conference, 2006, pp. 6–pp, IEEE, 2006.

[31] M. Stolle, Towards vision-based autonomous cross-country soaring. PhD thesis, ONERA, 2016.

[32] Y. Ameho, Du pilotage d'une famille de drones à celui d'un drone hybride via la commande adaptative. PhD thesis, Toulouse, ISAE, 2013.

[33] R. W. Beard and T. W. McLain, Small Unmanned Aircraft: Theory and Practice. Princeton, NJ, USA: Princeton University Press, 2012.

[34] D. Moorhouse and R. Woodcock, "US military specification MIL–F–8785C," US Department of Defense, 1980.

[35] D. Saban, J. F.Whidborne, and A. Cooke, "Simulation of Wake Vortex Effects for UAVs in Close Formation Flight," Aeronautical Journal, 2009.

[36] P. Tian, A. He, H. Chao, Z. C. Zheng, and Y. Gu, "Wake Encounter Simulation and Flight Validation with UAV Close Formation Flight," in AIAA Guidance, Navigation, and Control Conference, p. 1910, 2017.

[37] A. Dogan, S. Venkataramanan, and W. Blake, "Modeling of aerodynamic coupling between aircraft in close proximity," Journal of Aircraft, vol. 42, no. 4, pp. 941–955, 2005.

[38] K. Yang and S. Sukkarieh, "3D smooth path planning for a UAV in cluttered natural environments," in Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pp. 794–800, Sept 2008.

[39] G. Looye, "Helical flight path trajectories for autopilot evaluation," in Advances in Aerospace Guidance, Navigation and Control, pp. 79–90, Springer, 2011.

[40] H. Wang, J. Kearney, and K. Atkinson, "Arc-length parameterized spline curves for real-time simulation," in Proc. 5th International Conference on Curves and Surfaces, pp. 387–396, 2002.

[41] jmonkeyengine.org, "jmonkeyengine.org Java Game Engine." https://jmonkeyengine.org/, 2016.

[42] A. H. Bowers, J. W. Pahle, R. J. Wilson, B. C. Flick, and R. L. Rood, "An Overview of the NASA F-18 High Alpha Research Vehicle," NASA Technical Memorandum, 1996.

[43] I. K. Peddle, "Acceleration based manoeuvre flight control system for unmanned aerial vehicles," Stellenbosch University, Cape Town, PhD Thesis, 2008.

[44] C. J. Schumacher and R. Kumar, "Adaptive control of uavs in close-coupled formation flight," in American Control Conference, 2000. Proceedings of the 2000, vol. 2, pp. 849–853, IEEE, 2000.

[45] B. L. Stevens, F. L. Lewis, and E. N. Johnson, Aircraft control and simulation: dynamics, controls design, and autonomous systems. John Wiley & Sons, 2015.

[46] D. G. Klein, A preliminary investigation of the use of lift control in maneuvering flight. PhD thesis, Princeton University, 1966.

[47] W. E. McNeill, R. M. Gerdes, R. C. Innis, and J. D. Ratcliff, "A flight study of the use of direct-lift-control flaps to improve station keeping during in-flight refueling," NASA technical Memorandum, 1973.

[48] T. Lombaerts and G. Looye, "Design and flight testing of nonlinear autoflight control laws incorporating direct lift control," in Advances in Aerospace Guidance, Navigation and Control, pp. 549–568, Springer, 2013.

[49] R. Jaiswal, A. Shastry, S. Swarnkar, and M. Kothari, "Adaptive Longitudinal Control of UAVs with Direct Lift Control," IFAC-PapersOnLine, vol. 49, no. 1, pp. 296 – 301, 2016.

[50] R. Brockhaus, Flugregelung. Springer-Verlag, 2013.

[51] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, Sliding mode control and observation. Springer, 2014.

[52] A. Bartoszewicz, "Discrete-time quasi-sliding-mode control strategies," IEEE Transactions on Industrial Electronics, vol. 45, no. 4, pp. 633–637, 1998.

[53] C.-W. Chuang and C.-L. Hwang, "Applying discrete dynamic integral sliding surface control to hydraulic position control," in Industrial Technology, 2005. ICIT 2005. IEEE International Conference on, pp. 325–330, IEEE, 2005.

[54] M. Z. Shah, R. Samar, and A. I. Bhatti, "Guidance of air vehicles: A sliding mode approach," IEEE Transactions on Control Systems Technology, vol. 23, no. 1, pp. 231–244, 2015.

[55] Y.-W. Liao and J. K. Hedrick, "Robust model predictive control with discrete-time integral sliding surface," in 2015 American Control Conference (ACC), pp. 1641–1646, IEEE, 2015.

[56] M. Steinberger, I. Castillo, M. Horn, and L. Fridman, "Model predictive output integral sliding mode control," in Variable Structure Systems (VSS), 2016 14th International Workshop on, pp. 228–233, IEEE, 2016.

[57] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," Automatica, vol. 41, no. 2, pp. 219–224, 2005.

[58] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," Optimization and Engineering, vol. 13, no. 1, pp. 1–27, 2012.

[59] L. Van den Broeck, M. Diehl, and J. Swevers, "Model predictive control for time-optimal point-to-point motion control," IFAC Proceedings Volumes, vol. 44, no. 1, pp. 2458–2463, 2011.

[60] W.-C. Su, S. V. Drakunov, and U. Ozguner, "An $O(T^2)$ boundary layer in sliding mode for sampled-data systems," IEEE Transactions on Automatic Control, vol. 45, no. 3, pp. 482–485, 2000.

[61] B. M. Houda, D. Khadija, and N. A. Said, "New discrete predictive sliding mode control for non minimum phase systems," International Journal of Computer Applications, vol. 70, no. 11, 2013.

[62] M. Perez, E. Jimenez, and E. Camacho, "Design of an explicit constrained predictive sliding mode controller," IET Control Theory & Applications, vol. 4, no. 4, pp. 552–562, 2010.

[63] S. L. de Oliveira Kothare and M. Morari, "Contractive model predictive control for constrained nonlinear systems," IEEE Transactions on Automatic Control, vol. 45, no. 6, pp. 1053–1071, 2000.

[64] M. Alamir, "Contraction-based nonlinear model predictive control formulation without stability-related terminal constraints," Automatica, vol. 75, pp. 288–292, 2017.

[65] P. E. Orukpe, I. M. Jaimoukha, and H. M. El-Zobaidi, "Model predictive control based on mixed h2/h control approach," in American Control Conference, 2007. ACC'07, pp. 6147–6150, IEEE, 2007.

[66] L. Vandenberghe and V. Balakrishnan, "Algorithms and software tools for LMI problems in control: An overview," in Computer-Aided Control System Design, 1996., Proceedings of the 1996 IEEE International Symposium on, pp. 229–234, IEEE, 1996.

[67] ublox AG, Datasheet ublox LEA-6T, Mars 2012.

[68] P. Freda, A. Angrisano, S. Gaglione, and S. Troisi, "Time-differenced carrier phases technique for precise GNSS velocity estimation," GPS Solutions, vol. 19, no. 2, pp. 335–341, 2015.

[69] M. Smaoui, X. Brun, and D. Thomasset, "A robust differentiator-controller design for an electropneumatic system," in Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on, pp. 4385–4390, IEEE, 2005.

[70] P. Seiler, A. Pant, and J. Hedrick, "String instabilities in formation flight: Limitations due to integral constraints," Journal of dynamic systems, measurement, and control, vol. 126, no. 4, pp. 873–879, 2004.

[71] M. Brodecki and K. Subbarao, "Autonomous formation flight control system using in-flight sweet-spot estimation," Journal of Guidance, Control, and Dynamics, vol. 38, no. 6, pp. 1083–1096, 2014.

[72] emlid, Emlid Reach RTK, june 2017.

[73] G. MacGougan, K. O'Keefe, and D. Chiu, "Multiple UWB range assisted GPS RTK in hostile environments," in ION GNSS, pp. 3020–3035, 2008.

[74] P. Teunissen, P. De Jonge, and C. Tiberius, "The LAMBDA method for fast GPS surveying," in International Symposium "GPS Technology Applications" Bucharest, Romania, 1995.

[75] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp. 3131–3137, IEEE, 2015.

[76] B. Kempke, P. Pannuto, and P. Dutta, "Polypoint: Guiding indoor quadrotors with ultra-wideband localization," in Proceedings of the 2nd International Workshop on Hot Topics in Wireless, pp. 16–20, ACM, 2015.

[77] R. V. Dell'Aquila, G. Campa, M. R. Napolitano, and M. Mammarella, "Real-time machine-vision-based position sensing system for UAV aerial refueling," Journal of Real-Time Image Processing, vol. 1, no. 3, pp. 213–224, 2007.

[78] M. Mammarella, G. Campa, M. R. Napolitano, M. L. Fravolini, Y. Gu, and M. G. Perhinschi, "Machine vision/GPS integration using EKF for the UAV aerial refueling problem," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 6, pp. 791–801, 2008.

[79] R. A. Garcia, G. V. Raffo, M. G. Ortega, and F. R. Rubio, "Guaranteed Quadrotor Position Estimation Based on GPS Refreshing Measurements," IFAC-PapersOnLine, vol. 48, no. 9, pp. 67–72, 2015.

[80] F. Gu, Y. He, J. Han, and Y. Wang, "ESMF based multiple UAVs active cooperative observation method in relative velocity coordinates," in Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, pp. 3008–3013, IEEE, 2009.

[81] V. Drevelle and P. Bonnifait, "High integrity GNSS location zone characterization using interval analysis," in ION GNSS 2009, pp. 2178–2187, 2009.

[82] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo, "Localization of autonomous underwater vehicles by floating acoustic buoys: a set-membership approach," IEEE Journal of Oceanic Engineering, vol. 30, no. 1, pp. 140–152, 2005.

[83] P. Bonnifait, I. Fantoni, et al., "Cooperative Localization with Reliable Confidence Domains Between Vehicles Sharing GNSS Pseudoranges Errors with No Base Station," IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 1, pp. 22–34, 2017.

[84] E. Scholte and M. E. Campbell, "Online nonlinear guaranteed estimation with application to a high performance aircraft," in American Control Conference, 2002. Proceedings of the 2002, vol. 1, pp. 184–190, IEEE, 2002.

[85] E. Scholte and M. E. Campbell, "A nonlinear set-membership filter for on-line applications," International Journal of Robust and Nonlinear Control, vol. 13, no. 15, pp. 1337–1358, 2003.

[86] T. Alamo, J. M. Bravo, and E. F. Camacho, "Guaranteed state estimation by zonotopes," Automatica, vol. 41, no. 6, pp. 1035–1043, 2005.

[87] T. Alamo, J. M. Bravo, M. J. Redondo, and E. F. Camacho, "A set-membership state estimation algorithm based on DC programming," Automatica, vol. 44, no. 1, pp. 216–224, 2008.

[88] E. Hansen and G. W. Walster, Global optimization using interval analysis: revised and expanded, vol. 264. CRC Press, 2003.

[89] S. B. Pope, "Algorithms for ellipsoids," Cornell University Report No. FDA, pp. 08–01, 2008.

[90] E. Kaplan and C. Hegarty, Understanding GPS: Principles and Applications. Artech House, 2. ed., 2005.

[91] M. Kim and J. Kim, "A Long-term Analysis of the GPS Broadcast Orbit and Clock Error Variations," Procedia Engineering, vol. 99, pp. 654–658, 2015.

[92] D. L. Warren and J. F. Raquet, "Broadcast vs. precise GPS ephemerides: a historical perspective," GPS Solutions, vol. 7, no. 3, pp. 151–156, 2003.

[93] P. J. G. Teunissen, "A new method for fast carrier phase ambiguity estimation," in Position Location and Navigation Symposium, 1994., IEEE, pp. 562–573, 1994.

[94] J. Traugott, G. Dell'Omo, A. Vyssotski, D. Odijk, and G. Sachs, "A time-relative approach for precise positioning with a miniaturized L1 GPS logger," in Proceedings of ION GNSS 21th International Technical Meeting of the Satellite Division, pp. 1883–1894, ION GNSS, 2008.

[95] J. P. Traugott, Precise flight trajectory reconstruction based on time-differential GNSS carrier phase processing. Verlag Dr. Hut, 2011.

[96] J. Bolting, "A GPS Local - Area Augmentation System for small UAS using consumer - grade single frequency receivers," technical report, University of Braunschweig, Institute of Technology, 2013.

[97] C. Durieu, E. Walter, and B. Polyak, "Set-membership estimation with the trace criterion made simpler than with the determinant criterion," IFAC Proceedings Volumes, vol. 33, no. 15, pp. 1007–1012, 2000.

[98] D. Maksarov and J. Norton, "Computationally efficient algorithms for state estimation with ellipsoidal approximations," International Journal of Adaptive Control and Signal Processing, vol. 16, no. 6, pp. 411–434, 2002.

[99] G. Belforte and B. Bona, "An improved parameter identification algorithm for signals with unknown but bounded errors," in 7th IFAC Symposium on Identification and System Parameter Estimation, 1985.

[100] J. Bolting, F. Defay, and J.-M. Moschetta, "Differential GPS for small UAS using consumer-grade single-frequency receivers," in Proceedings of the International Conference on Micro Aerial Vehicles 2013, 2013.

[101] Z.-B. Xu, J.-S. Zhang, and Y.-W. Leung, "An approximate algorithm for computing multidimensional convex hulls," Applied mathematics and computation, vol. 94, no. 2-3, pp. 193–226, 1998.

[102] A. Bemporad and C. Rocchi, "Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles," in Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pp. 7488–7493, IEEE, 2011.

[103] J. Lagneaux, "Integration d'un capteur de distance dans un système de cinématique temps réel appliqué aux drones," technical report, ISAE, 2015.

[104] T. Ye, M. Walsh, P. Haigh, J. Barton, and B. O'Flynn, "Experimental impulse radio IEEE 802.15. 4a UWB based wireless sensor localization technology: Characterization, reliability and ranging," in ISSC 2011, 22nd IET Irish Signals and Systems Conference, Dublin, Ireland. 23-24 Jun 2011, Insitution of Engineering and Technology, 2011.

[105] A. R. J. Ruiz and F. S. Granja, "Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis," IEEE Transactions on Instrumentation and Measurement, 2017.

[106] R. Nelson, T. Corke, C. He, H. Othman, T. Matsuno, M. Patel, and T. Ng, "Modification of the flow structure over a UAV wing for roll control," in 45th AIAA aerospace sciences meeting and exhibit, p. 884, 2007.

[107] M. P. Patel, T. T. Ng, S. Vasudevan, T. C. Corke, and C. He, "Plasma actuators for hingeless aerodynamic control of an unmanned air vehicle," Journal of Aircraft, vol. 44, no. 4, pp. 1264–1274, 2007.

[108] S. Ponda, R. Kolacinski, and E. Frazzoli, "Trajectory optimization for target localization using small unmanned aerial vehicles," in AIAA guidance, navigation, and control conference, p. 6015, 2009.

[109] J. Grzymisch and W. Fichter, "Optimal rendezvous guidance with enhanced bearings-only observability," Journal of Guidance, Control, and Dynamics, vol. 38, no. 6, pp. 1131–1140, 2015.

[110] A. Pant, P. Seiler, and K. Hedrick, "Mesh stability of look-ahead interconnected systems," IEEE Transactions on Automatic Control, vol. 47, no. 2, pp. 403–407, 2002.

[111] S. Knorn, A. Donaire, J. C. Agüero, and R. H. Middleton, "Passivity-based control for multi-vehicle systems subject to string constraints," Automatica, vol. 50, no. 12, pp. 3224–3230, 2014.

[112] H. Castañeda, O. S. Salas-Peña, and J. de León Morales, "Adaptive super twisting flight control-observer for a fixed wing uav," in Unmanned Aircraft Systems (ICUAS), 2013 International Conference on, pp. 1004–1013, IEEE, 2013.