CONTROL OF FIXED WING UNMANNED AERIAL VEHICLES IN AUTONOMOUS VARIABLE SHAPE FORMATION FLIGHT

THIAGO FELIPPE KURUDEZ CORDEIRO

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CONTROLE DE VEÍCULOS AÉREOS NÃO TRIPULADOS DE ASA FIXA EM VOO AUTÔNOMO EM FORMAÇÃO COM FORMATO VARIÁVEL

CONTROL OF FIXED WING UNMANNED AERIAL VEHICLES IN AUTONOMOUS VARIABLE SHAPE FORMATION FLIGHT

THIAGO FELIPPE KURUDEZ CORDEIRO

ORIENTADOR: HENRIQUE CEZAR FERREIRA
COORIENTADOR: JOÃO YOSHIYUKI ISHIHARA

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PGEAENE.TD - 136/2018

BRASÍLIA/DF: NOVEMBRO – 2018

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CONTROL OF FIXED WING UNMANNED AERIAL VEHICLES IN AUTONOMOUS VARIABLE SHAPE FORMATION FLIGHT

THIAGO FELIPPE KURUDEZ CORDEIRO

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.

APROVADA POR:

HENRIQUE CEZAR FERREIRA, Dr., ENE/UNB
(ORIENTADOR)

RENATO ALVES BORGES, Dr., ENE/UNB
(EXAMINADOR INTERNO)

Marco Henrique Terra

MARCO HENRIQUE TERRA, Dr., USP/EESC
(EXAMINADOR EXTERNO)

RLopes

RENATO VILELA LOPES, Dr., FGA/UNB
(EXAMINADOR INTERNO)

Brasília, 22 de novembro de 2018.

FICHA CATALOGRÁFICA

CORDEIRO, THIAGO FELIPPE KURUDEZ

Control of Fixed Wing Unmanned Aerial Vehicles in Autonomous Variable Shape Formation Flight [Distrito Federal] 2018.

xv, 103p., 210 x 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2018).

Departamento de Engenharia Elétrica

1. Unmanned Aerial Vehicles
2. Autonomous Formation Flight
3. Nonlinear Dynamic Inversion
4. Sliding Mode Control

I. ENE/FT/UnB
II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

CORDEIRO, T. F. K. (2018). Control of Fixed Wing Unmanned Aerial Vehicles in Autonomous Variable Shape Formation Flight. Tese de Doutorado em Engenharia Elétrica, Publicação PGEAENE.TD-136/2018, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 103p.

CESSÃO DE DIREITOS

AUTOR: Thiago Felippe Kurudez Cordeiro.

TÍTULO: Control of Fixed Wing Unmanned Aerial Vehicles in Autonomous Variable Shape Formation Flight.

GRAU: Doutor ANO: 2018

Thiago F. K. Cordeiro

Thiago Felippe Kurudez Cordeiro
Departamento de Eng. Elétrica (ENE) – FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 – Brasília – DF - Brasil

To my wife and children.

THIAGO FELIPPE KURUDEZ CORDEIRO

Acknowledgments

I would like to acknowledge my wife Manuela, my children Beatriz, Lucas and Juliana, my parents Romério and Mércia and my brothers Vinícius and Bruna, for being supportive to me during the PhD. Also, I would like to acknowledge the family of my wife, including her mother Nancy and her father João, whose also provided support when needed.

I would also like to acknowledge my advisor Prof. Henrique Ferreira for his dedication in advising, teaching and by his ability to be able to always find highly relevant literature related to the studied problem; Prof. João Ishihara, for his advising and by the prolific discussions we had; and to Prof. Geovany Borges, who advised me in the first two years of my PhD.

Lastly, I would like to acknowledge the Aerospace department from the Gama Campus of the University of Brasília, my workplace, which was supportive to me during the PhD course.

THIAGO FELIPPE KURUDEZ CORDEIRO

ABSTRACT

CONTROL OF FIXED WING UNMANNED AERIAL VEHICLES IN AUTONOMOUS VARIABLE SHAPE FORMATION FLIGHT

Author: Thiago Felippe Kurudez Cordeiro

Supervisor: Prof. Dr. Henrique Cezar Ferreira, ENE/UnB

Graduate Program in Engineering of Electronic and Automation Systems - PGEA

Brasília, November 2018

Keywords: Unmanned Aerial Vehicles, Autonomous Formation Flight, Nonlinear Dynamic Inversion, Sliding Mode Control

In this thesis fixed wing unmanned aerial vehicle (UAV) formation flight controllers are proposed. First, it is shown that independently of the used coordinate frame to model the formation, nonlinear dynamic inversion (NLDI) based controllers simplify the formation dynamics to a double integrator. To the leader's frame, a NLDI-based controller is proposed to time-varying formation shape and maneuvering leader, an improvement to NLDI-based controllers from literature. A leader's reference frame based path planner and collision avoidance algorithm is also proposed. The performance of the NLDI-based relies in the precision of the model used in the project. The sliding mode control (SMC) technique can provide robustness to the control system. However, it can result in control laws with excessive chattering, which can compromise the durability of actuators. In this thesis, two SMC-based formation flight control architectures with low pass filters to attenuate the unwanted effect of the chattering are proposed. In these architectures, the control law in each UAV uses its own data and data from UAVs from its neighborhood. The stability of the proposed control architectures is proved by Lyapunov functions. The effectiveness of the proposed controllers is shown via simulations.

RESUMO

CONTROLE DE VEÍCULOS AÉREOS NÃO TRIPULADOS DE ASA FIXA EM VOO AUTÔNOMO EM FORMAÇÃO COM FORMATO VARIÁVEL

Autor: Thiago Felippe Kurudez Cordeiro

Orientador: Prof. Dr. Henrique Cezar Ferreira, ENE/UnB

Programa de Pós-Graduação em Engenharia de Sistemas Eletrônicos e de Automação - PGEA

Brasília, novembro de 2018

Palavras-chave: Veículos Aéreos Não Tripulados, Voo em Formação Autônomo, Inversão Não Linear, Controle por Modos Deslizantes.

Nesta tese são propostos controladores para voo em formação de veículos aéreos não tripulados (VANTs) de asa fixa. Primeiro, mostra-se que independente do sistema de referência adotado para modelar a formação, controladores baseados na técnica de inversão não linear simplificam a dinâmica da formação para de duplo integrador. Para o sistema de referência do líder, é proposto um controlador baseado em inversão não linear para formações com formato variante no tempo e com líder manobrante, o que representa um avanço em relação aos controladores baseados em inversão linear para voo em formação existentes na literatura. Também é proposto um algoritmo de planejamento de trajetória e de prevenção de colisão baseado no sistema de referência do líder. O desempenho de controladores baseados em inversão não linear depende da precisão dos modelos utilizados para projeto. A técnica de controle por modos deslizantes pode conferir robustez ao sistema de controle. No entanto, pode resultar leis de controle que chaveiam em excesso, o que pode comprometer a durabilidade de atuadores. Nesta tese são então propostas duas arquiteturas de controle de voo em formação baseadas na técnica de controle por modos deslizantes e com filtros passa baixa para atenuar o efeitos indesejado do chaveamento. Nessas arquiteturas a ação de controle em cada VANT utiliza seus próprios dados e dados de VANTs de sua vizinhança. A estabilidade das arquiteturas de controle propostas é demonstrada utilizando-se funções de Lyapunov. A eficácia dos controladores propostos é mostrada por meio de simulações.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND NOTATIONS

Coordinate frames and attitude

S_x	Coordinate system, or frame, x
S_{NED}	North-East-Down (NED) frame
S_{bi}	body frame b of i-th aircraft
S_{wi}	Wind frame of i-th aircraft
S_{swi}	Wind frame of i-th aircraft
$S_{\chi i}$	χ frame of i-th aircraft
$\boldsymbol{D}_y^x(t)$	Direction Cosine Matrix, or rotation matrix, between frames S_x and S_y (e.g. NED to b)
$\boldsymbol{D}_i(\alpha)$	Rotation matrix, rotates an angle α around the i-th axis
$\boldsymbol{v}_x$	A vector v in a frame S_x
$\phi_i(t), \theta_i(t), \psi_i(t)$	Roll, pitch and yaw attitude angles, relating S_{bi} to S_{NED}
$\mu_i(t), \gamma_i(t), \chi_i(t)$	Bank, flight path and course attitude angles, relating S_{wi} to S_{NED}
$\alpha_i(t)$, $\beta_i(t)$	Angle-of-attack and sideslip angle, relating S_{bi} to S_{wi}.
$\boldsymbol{\omega}_z^{xy}$	Angular velocity between frames x and z, represented in frame z
x_b, y_b, z_b	Orthonormal body frame axes
x_w, y_w, z_w	Orthonormal wind frame axes

Aircraft model

m_i	Aircraft mass
$\bar{c}$	Mean aerodynamic chord
b	Wing span
S	Wing area
ρ	Air density
$\bar{q}_i(t)$	Dynamic pressure
$\boldsymbol{\xi}_i(t)$	State vector
$\boldsymbol{u}_i(t)$	Input vector
$T_i(t)$	Thrust
$D_i(t)$, $Y_i(t)$, $L_i(t)$	Drag, side and lift aerodynamic forces, in S_{wi} frame
$n_i(t)$	Load factor
J_x, J_y, J_z,	Aircraft moments of inertia
J_{xy}, J_{xz}, J_{yz}	Aircraft products of inertia
$\boldsymbol{J}$, $\boldsymbol{M}_0$, $\boldsymbol{M}_1$, $\boldsymbol{M}_2$	Inertia-related matrices
$\bar{L}_i(t)$, $M_i(t)$, $N_i(t)$	Aerodynamic moments, in the S_{bi} frame
$P_i(t)$, $Q_i(t)$, $R_i(t)$	Angular velocity in the S_{bi} frame
$\delta_{e,i}(t)$, $\delta_{a,i}(t)$, $\delta_{r,i}(t)$	Elevator, aileron and rudder angular position

$\delta_{T,i}(t)$	Engine throttle
$\delta_{d,e,i}(t)$, $\delta_{d,a,i}(t)$, $\delta_{d,r,i}(t)$	Commanded (desired) elevator, aileron and rudder angular position
$\delta_{d,T,i}(t)$	Commanded (desired) engine throttle
$\delta_{\text{surface}}(t)$	Control surface angular position
$\delta_{\text{commanded}}(t)$	Commanded (desired) control surface angular position
$t_{d,a}$, $t_{d,T}$	Control surface actuators time delay, and engine time delay
τ_a, τ_T	Control surface actuators and engine time constant
K_T, T_b	Engine model coefficients
$C_{D,i}(t)$, $C_{L,i}(t)$, $C_{Y,i}(t)$	Drag, lift and side aerodynamic forces coefficients
$C_{l,i}(t)$, $C_{m,i}(t)$, $C_{n,i}(t)$	Rolling, pitching and yawing aerodynamic moments coefficients.
$C_{xy}(t)$	Aerodynamic derivative. $x = D$, L, Y, l, m, n, and $y = 0$, α, β, p, q, r, δ_e, δ_a, δ_r
g	Magnitude of gravity acceleration
$\boldsymbol{g}_{\text{NED}}$	Gravity acceleration vector, represented in S_{NED}
$\bar{\boldsymbol{a}}$	Inertial acceleration
$\boldsymbol{a}$	Accelerometer-measured acceleration, or specific force. Does not include the gravity field generated acceleration.
$x_i(t)$, $y_i(t)$, $z_i(t)$	Aircraft position in S_{NED}
$V_{gi}(t)$	Ground speed
$V_i(t)$	Airspeed
$V_{N,i}(t)$, $V_{E,i}(t)$, $V_{D,i}(t)$	Components of the inertial velocity described in S_{NED} frame
$\bar{a}_{N,i}(t)$, $\bar{a}_{E,i}(t)$, $\bar{a}_{D,i}(t)$	Components of the inertial acceleration described in S_{NED} frame
$a_{N,i}(t)$, $a_{E,i}(t)$, $a_{D,i}(t)$	Components of the accelerometer-measured acceleration, described in S_{NED} frame
$L_{\text{xy},i}(t)$, $L_{z,i}(t)$	Horizontal and vertical components of the lift force
$\boldsymbol{p}_{\text{NED},i}(t)$	Aircraft position vector, described in S_{NED} frame
$\boldsymbol{V}_{\text{NED},i}(t)$	Aircraft velocity vector, described in S_{NED} frame
$\boldsymbol{V}_{bi}$	Inertial velocity in S_{bi}
$\omega_{1,i}(t)$, $\omega_{2,i}(t)$	Used in a linear equation to convert thrust to acceleration
$P(t)$, $P_e(t)$	Power and engine power
$F(t)$	Force
η	Engine efficiency
$\boldsymbol{A}_{\text{lat}}$, $\boldsymbol{B}_{\text{lat}}$	Linearized lateral model of the aircraft
$\boldsymbol{A}_{\text{lon}}$, $\boldsymbol{B}_{\text{lon}}$	Linearized longitudinal model of the aircraft

Inner loop controllers

$K_{P,h}$, $K_{I,h}$	Proportional and integral gains of the altitude controller
$K_{P,\theta}$, $K_{I,\theta}$, $K_{D,\theta}$	Proportional, integral and derivative gains of the pitch angle controller
K_{ff}	Roll feedforward gain of the pitch angle controller
$K_{P,U}$, $K_{I,U}$, $K_{D,U}$	Proportional, integral and derivative gains of the forward airspeed controller
$K_{P,\phi}$, $K_{I,\phi}$, $K_{D,\phi}$	Proportional, integral and derivative gains of the roll controller
K_{P,a_y}, K_{I,a_y}	Proportional and integral gains of the sideslip regulator
K_P, K_I, K_D	Proportional, integral and derivative gains
t_r, t_s, M_p	Rise time, settling time and overshoot

Point-mass model

$u_{t,i}(t)$, $u_{y,i}(t)$, $u_{p,i}(t)$	Inputs that affect, respectively, $\dot{V}_i(t)$, $\dot{\chi}_i(t)$, and $\dot{\gamma}_i(t)$
$\dot{V}_{\max}$, $\dot{\chi}_{\max}$, $\dot{\gamma}_{\max}$	Maximum allowed values of $\dot{V}_i(t)$, $\dot{chi}_i(t)$, and $\dot{\gamma}_i(t)$
r_h, r_v	Minimum horizontal and vertical turn radius
$\boldsymbol{a}_{swi}(t)$	Inertial acceleration described in S_{swi} frame
$a_{swxi}(t)$, $a_{swyi}(t)$ $a_{swzi}(t)$	Inertial acceleration components described in S_{swi} frame
$\boldsymbol{\Gamma}_i(t)$	Acceleration generated by the aircraft actuators in S_{swi} frame
$a_{ti}(t)$, $a_{yi}(t)$, $a_{pi}(t)$	Acceleration components generated by the aircraft actuators in S_{swi} frame
$\boldsymbol{b}_i(t)$	Acceleration generated by disturbances, in S_{swi} frame
$b_{ti}(t)$, $b_{yi}(t)$, $b_{pi}(t)$	Acceleration components generated by disturbances, in S_{swi} frame

Dynamic systems and NLDI

$f(\cdot)$, $\boldsymbol{f}(\cdot)$	Nonlinear state function / state function vector
$b(\cdot)$, $\boldsymbol{B}(\cdot)$	Nonlinear input function / input function vector
$h(\cdot)$, $\boldsymbol{h}(\cdot)$	Nonlinear output function / output function vector
$y(t), \boldsymbol{y}(t)$	Output vector
n,m	Number of states and outputs
r_i, r	Relative degree and total relative degree
$k_0 \dots k_{n-1}$	Coefficients of a linear system
$\boldsymbol{\xi}_i^d(t)$	Desired state vector
$y^d(t), \boldsymbol{y}^d(t)$	Output vector
$v(t), \boldsymbol{v}(t)$	Virtual control input

S_{NED}-NLDI formation controller

$\boldsymbol{\tau}_i(t)$	Virtual control input
$\tau_{xi}(t)$, $\tau_{yi}(t)$, $\tau_{zi}(t)$	Virtual control input components
$\boldsymbol{d}_i(t)$	Disturbances in S_{NED}

$d_{xi}(t)$, $d_{yi}(t)$, $d_{zi}(t)$	Disturbance components in S_{NED}
$\boldsymbol{e}_i(t)$	Output (position) error, in S_{NED}
$e_{xi}(t)$, $e_{yi}(t)$, $e_{zi}(t)$	Components of $\boldsymbol{e}_i(t)$
$\boldsymbol{k}_{di}(t)$, $\boldsymbol{k}_{pi}(t)$	Control gain diagonal matrices
$k_{dxi}(t)$, $k_{dyi}(t)$, $k_{dzi}(t)$	Components of $\boldsymbol{k}_{di}(t)$
$k_{pxi}(t)$, $k_{pyi}(t)$, $k_{pzi}(t)$	Components of $\boldsymbol{k}_{pi}(t)$
$\boldsymbol{p}^d_{\text{NED},i}(t)$	Desired position, in S_{NED}
$p^d_{xi}(t), p^d_{yi}(t), p^d_{zi}(t)$	Components of $\boldsymbol{p}^d_{\text{NED},i}(t)$
$\tilde{\boldsymbol{p}}_{\text{NED},i}(t)$	Desired relative position between the leader and the i-th follower, in S_{NED}
$\tilde{x}_i(t)$, $\tilde{y}_i(t)$, $\tilde{z}_i(t)$	Components of $\tilde{\boldsymbol{p}}_{\text{NED},i}(t)$
$\tilde{\boldsymbol{p}}_{Li}(t)$	Desired relative position between the leader and the i-th follower, described using some leader's frame.
$\boldsymbol{D}^L_{\text{NED}}(t)$	Direction Cosine Matrix between some leader's frame and S_{NED}
$\boldsymbol{\omega}_L^{L,\text{NED}}$	Angular velocity between some leader's frames and a frame z, represented in frame z

$S_{\chi 0}$-NLDI formation controller

$\Delta\boldsymbol{p}_{\text{NED},i}(t)$	Distance vector between leader and the i-th follower in S_{NED} frame
$\Delta\boldsymbol{p}_{\chi 0,i}(t)$	Distance vector between leader and the i-th follower in leader's $S_{\chi 0}$ frame
$f_i(t), l_i(t), v_i(t)$	Forward, lateral and vertical distance between leader and i-th follower, from $\Delta\boldsymbol{p}_{\chi 0,i}(t)$
$\Delta\boldsymbol{p}^d_{\chi 0,i}(t)$	Desired distance vector between leader and follower
$f_{d,i}(t)$, $l_{d,i}(t)$, $v_{d,i}(t)$	Desired forward, lateral and vertical distance between leader and follower, components of $\Delta\boldsymbol{p}^d_{\chi 0,i}(t)$
$\boldsymbol{e}_{\chi 0,i}$	Difference between $\Delta\boldsymbol{p}_{\chi 0,i}(t)$ and $\Delta\boldsymbol{p}^d_{\chi 0,i}(t)$
$f_{e,i}(t)$, $l_{e,i}(t)$, $v_{e,i}(t)$	Difference between the obtained and desired values of forward, lateral and vertical distances, components of $\boldsymbol{e}_{\chi 0,i}$
$\Delta\chi_i(t)$	$\chi_0(t)$ - $\chi_i(t)$
$V_{xy,i}(t)$	Horizontal speed
$\dot{\chi}_{d,i}(t)$	Desired rate-of-turn
$T_{d,i}(t)$	Desired thrust
$\dot{f}'_i(t)$, $\dot{l}'_i(t)$, $\dot{f}'_{e,i}(t)$, $\dot{l}'_{e,i}(t)$, $\dot{\chi}'_{d,i}$, $T'_{d,i}$	Variables calculated as in literature, and not as in the proposed controller
$v_{f,i}$, $v_{l,i}$	Virtual control inputs in forward and lateral directions
K_{pfi}, K_{dfi}	Forward distance control gains
K_{pli}, K_{dli}	Lateral distance control gains

Path planner

$a_{f,i,\min}$, $a_{f,i,\max}$, $a_{l,i,\min}$, $a_{l,i,\max}$	Acceleration limits on forward and lateral directions, used in path planning
$v_{f,i,\min}$, $v_{f,i,\max}$, $v_{l,i,\min}$, $v_{l,i,\max}$	Speed limits on forward and lateral speed, used in path planning
$\Delta f_{i,j}(t)$, $\Delta l_{i,j}(t)$	Forward and lateral distances between UAVs i and j
$\Delta p_{i,j}(t)$	Magnitude of distance between UAVs
$\Delta p_{\min}$	Minimum allowed distance
$t_{j,i}$	Time instant j, i-th UAV

Sliding mode control

$s(t)$	Scalar function that defines the sliding surface
ϵ	Defines t_{reach} from initial error
t_{reach}	Finite time to reach the sliding surface
Φ	Scale factor of the sat function
$\hat{f}$	Estimated nonlinear state function
$\hat{b}$, $\hat{\boldsymbol{B}}$	Estimated input matrix
F	Upper bound on state function uncertainty
β	Upper bound in amplification generated by $\boldsymbol{B}$ matrix
Δ	Upper bound of the disturbance
η	Gain applied in the discontinuous function sgn
Δ_{xi}, Δ_{yi}, Δ_{zi}	Upper bound of the disturbance, described in S_{NED}
Δ_{dxi}, Δ_{dyi}, Δ_{dzi}	Upper bound of the derivative of disturbance, described in S_{NED}
δ_{ti}, δ_{pi}, δ_{yi}	Upper bound of the disturbance, described in S_{swi}
δ_{dti}, δ_{dpi}, δ_{dyi}	Upper bound of the derivative of disturbance, described in S_{swi}
$\Delta\boldsymbol{e}_{ij}$	$\boldsymbol{e}_i$ - $\boldsymbol{e}_j$, errors of i-th and j-th UAVs
$\boldsymbol{e}_i^c$	Composed formation error
$\boldsymbol{E}_x(t)$	Concatenation of $e_{xi}(t)$, $\forall i = \{1 \dots N\}$
$\boldsymbol{E}_x^c(t)$	Concatenation of $e_{xi}^c(t)$, $\forall i = \{1 \dots N\}$
$\mathcal{N}_i$	Neighborhood of the i-th aircraft
a_{ij}	Gain applied in error $\boldsymbol{e}_j(t)$ in the i-th aircraft controller
λ_i	Gain applied in own error
$\boldsymbol{\Lambda}$	Diagonal matrix containing $\lambda_1, \dots \lambda_N$
$\boldsymbol{L}$	Laplacian communication matrix
$\boldsymbol{H}$	$\boldsymbol{\Lambda}$ + $\boldsymbol{L}$. Is positive definite
$u_{xi}(t)$, $u_{yi}(t)$, $u_{zi}(t)$	Control signal, applied in the low pass filter, axes x, y, z
$\boldsymbol{u}_i(t)$	Concatenation of $u_{xi}(t)$, $u_{yi}(t)$, $u_{zi}(t)$
$\boldsymbol{U}_x(t)$	Concatenation of $u_{xi}(t)$, $\forall i = \{1 \dots N\}$
$\tau_{xi}(t)$, $\tau_{yi}(t)$, $\tau_{zi}(t)$	Control signal, applied in the UAV, axes x, y, z
$\boldsymbol{\tau}_i(t)$	Concatenation of $\tau_{xi}(t)$, $\tau_{yi}(t)$, $\tau_{zi}(t)$
$\boldsymbol{\tau}_x(t)$	Concatenation of $\tau_{xi}(t)$, $\forall i = \{1 \dots N\}$

$u_{xai}(t)$, $u_{yai}(t)$, $u_{zai}(t)$	Smooth component of $u_{xi}(t)$, $u_{yi}(t)$, $u_{zi}(t)$
$\boldsymbol{u}_{ai}(t)$	Concatenation of $u_{xai}(t)$, $u_{yai}(t)$, $u_{zai}(t)$
$\boldsymbol{U}_{xa}(t)$	Concatenation of $u_{xai}(t)$, $\forall i = \{1 \dots N\}$
$\tau_{xai}(t)$, $\tau_{yai}(t)$, $\tau_{zai}(t)$	Smooth component of $\tau_{xi}(t)$, $\tau_{yi}(t)$, $\tau_{zi}(t)$
$\boldsymbol{\tau}_{ai}(t)$	Concatenation of $\tau_{xai}(t)$, $\tau_{yai}(t)$, $\tau_{zai}(t)$
$\boldsymbol{\tau}_{xa}(t)$	Concatenation of $\tau_{xai}(t)$, $\forall i = \{1 \dots N\}$
$u_{xbi}(t)$, $u_{ybi}(t)$, $u_{zbi}(t)$	Chattering component of $u_{xi}(t)$, $u_{yi}(t)$, $u_{zi}(t)$
$\boldsymbol{u}_{bi}(t)$	Concatenation of $u_{xbi}(t)$, $u_{ybi}(t)$, $u_{zbi}(t)$
$\boldsymbol{U}_{xb}(t)$	Concatenation of $u_{xbi}(t)$, $\forall i = \{1 \dots N\}$
$\tau_{xbi}(t)$, $\tau_{ybi}(t)$, $\tau_{zbi}(t)$	Filtered chattering component of $\tau_{xi}(t)$, $\tau_{yi}(t)$, $\tau_{zi}(t)$
$\boldsymbol{\tau}_{bi}(t)$	Concatenation of $\tau_{xbi}(t)$, $\tau_{ybi}(t)$, $\tau_{zbi}(t)$
$\boldsymbol{\tau}_{xb}(t)$	Concatenation of $\tau_{xbi}(t)$, $\forall i = \{1 \dots N\}$
$\xi_{xi}(t)$, $\xi_{yi}(t)$, $\xi_{zi}(t)$	Cutoff frequency of the low pass filter in axes x, y, z
$\boldsymbol{\Xi}_x(t)$	Diagonal matrix containing ξ_{xi}, $\forall i = \{1 \dots N\}$
$\boldsymbol{P}_x(t)$	Concatenation of $x_i(t)$, $\forall i = \{1 \dots N\}$
$\boldsymbol{D}_x(t)$	Concatenation of $d_{xi}(t)$, $\forall i = \{1 \dots N\}$
$s_{xi}(t)$	Scalar function that defines the sliding surface in x axis for the i-th UAV
$\boldsymbol{S}_x(t)$	Concatenation of $s_{xi}(t)$, $\forall i = \{1 \dots N\}$
$\bar{s}_{xi}(t)$	Sliding surface from literature, in x axis, for the i-th UAV
$\bar{\boldsymbol{S}}_x(t)$	Concatenation of $\bar{s}_{xi}(t)$, $\forall i = \{1 \dots N\}$
$\boldsymbol{E}_x^{cc}(t)$	Coupling of coupled error
$\mathcal{V}_x(t)$	Lyapunov candidate function

Subscripts

i	i-th aircraft, where 0 is the leader and 1 to N are followers
d	Desired value
e	Error, or difference, between the desired and true (or measured) value
NED, bi, wi, swi, χi	Frame in which the vector ir represented

Upperscripts

$\dot{}$	Time derivative of a signal
$\tilde{}$	Measured value
$\hat{}$	Estimated value
(n)	n-th derivative
c	Composed

Acronyms and Abbreviations

6-DoF	6 Degrees of Freedom nonlinear aircraft model
AFF	Autonomous Formation Flight
ANAC	Agência Nacional de Aviação Civil (National Civil Aviation Agency)
CG	Center of Gravity
DCM	Direction Cosine Matrix, or attitude matrix
GPS	Global Positioning System
GNC	Guidance, Navigation, and Control
IMU	Inertial Measurement Unit
LPF	Low Pass Filter
LQR	Linear Quadratic Regulator
MIMO	Multiple Input, Multiple Output
NED	North-East-Down
NLDI	NonLinear Dynamical Inversion
NLFC	NonLinear Formation Controller, a controller present in literature
E-NLFC	Extended NLFC, a proposed formation controller
PD	Proportional-Derivative controller
PI	Proportional-Integral controller
PID	Proportional-Integral-Derivative controller
SISO	Single Input, Single Output
SMC	Sliding Mode Control
S_{NED}-NLDI	NLDI controller applied in the S_{NED}-modeled formation
$S_{\chi 0}$-NLDI	NLDI controller applied in the $S_{\chi 0}$-modeled formation
UAS	Unmanned Aerial Systems
UAV	Unmanned Aerial Vehicle
YF-22	An model aircraft

Operators

$\frac{d}{dt}\boldsymbol{v}\big\|_{S_x}$	Derivative of vector $\boldsymbol{v}$, evaluated at frame S_x
$\boldsymbol{a} \times \boldsymbol{b}$	Cross-product between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
$\\|\boldsymbol{v}\\|$	2-norm of a vector
$\\|\boldsymbol{v}\\|_1$	1-norm of a vector
$\mathrm{sat}(\cdot)$	Saturation function
$\mathrm{sgn}(\cdot)$	Sign function

1 INTRODUCTION

Unmanned aerial vehicles (UAVs), also known as unmanned aerial systems (UAS), are vehicles with no pilot/crew onboard, and that sustains itself in the airspace via aerodynamic reactions [1], which includes fixed-wing (airplane) and rotary wing (helicopter, quadcopter) vehicles. By this definition, both remotely-piloted and fully autonomous aircraft are classified as UAVs.

The civil, military and academic interest in UAVs is high, being a current and relevant topic. As an example of civil interest, the Brazilian National Civil Aviation Agency (ANAC) presented recently, in 2017, the RBAC-E n^o 94 [1], a regulation about civilian non-recreational use of UAVs. The regulation defines itself as restrictive, but temporary, in which the agency aims to gain experience in the topic and to promote a safe and sustainable development of the technology. For example, currently it is still forbidden the use of fully autonomous aircraft, or the use of UAV to transport passengers or animals. The regulation list some applications, such as in agriculture, photography, cargo transportation, disease vectors control, public security, and civil defense. As military use, the UAVs are interesting because, for example, there is not an human pilot limiting the maximum allowed intensity of maneuver or the duration of operation and, if the aircraft is shot down, there is no life loss. There are several countries that uses UAVs as intelligence and/or combat equipment. Further discussion in the military use of UAVs can be found in, e.g, [2]. The academic interest can be seen from most of the bibliography presented in this thesis. Also, there is a conference specific of this topic, the International Conference on Unmanned Aircraft Systems, see for instance [3]. Some journals explicitly specifies UAVs as one of the areas of interest, such as the Journal of Intelligent & Robotic Systems.

An interesting and active topic in unmanned aerial vehicles is the multi-agent systems. The interaction between two UAVs can be classified as follows [4]

- **Cooperative**: both UAVs actively maneuvers to achieve some objective, such as maintain the formation shape or to succeed in an autonomous aerial refueling maneuver.
- **Evasive**: one UAV maneuver to avoid the interaction with the other. For example, an UAV can actively try to evade an missile.
- **Uncooperative**: when one of the UAVs does not actively maneuver to cooperates with or to evade the other.

Examples of cooperative behavior in literature are: a fleet of UAVs that must realize a set of tasks (e.g., to visit a region in space or attack a target) in the least amount of time or minimizing fuel comsumption, respecting the UAV movement constraints [5, 6], a fleet of UAVs that seeks the source of a radiant emission, such as thermal, acoustic, luminous, or radioactive [7], the guaranteed search of a smart and fast evader by a fleet of UAVs[8], and the cooperative tracking of a moving ground target in an urban environment in which occlusions can occur [9].

Evasive behavior usually occur in hostile environment. A counter example can be found in [10]. In this paper, an aerial refueling maneuver is being made. The drogue, which is a cone-shaped component of the aerial refueling system located at the tip of the fuel hose, is carried by the wind. Because of the intensity of the wind gusts, the random movement of the drogue makes the aerial refueling maneuver harder to the aircraft that is being refueled. Even being not purposely evasive, the movement of the drogue evades the maneuver.

The uncooperative behavior is common in leader-follower formation flight. In this case, a follower actively follows the leader, but the leader does not change in any way its trajectory to achieve the formation shape faster. Examples includes [11, 12, 13, 14, 15].

It is worth to note that the cooperation classification is specific to maneuvers. In most of the cited uncooperative formation flight, the leader broadcasts information about itself, which means that it cooperates in the sense of providing data.

1.1 AUTONOMOUS FORMATION FLIGHT

Autonomous formation flight (AFF) is a specific case of multi-agent systems. It consists of UAVs flying together, either in a predetermined shape, or simply near each other. A simple benefit can be seen as to control a fleet as a single entity. Formation control can be used also to reduce the fuel consumption of a fleet: by correctly positioning itself behind other aircraft, the aircraft can achieve better lift-to-drag ratio, which reduces the force that must be generated by the engine to maintain the aircraft's speed [13, 16]. Inter-aircraft physical interactions, such as autonomous aerial refueling [17, 4, 10] or the rendezvous between small UAV and a mothership [18] are also applications of autonomous formation flight. Finally, another interesting use of formation flight is to generate a flying sensor array. Sensor arrays are sets of sensors in which the spatial distribution is important. For example the Very Large Telescope in Atacama Desert, in Chile, consists of four 8.2 m diameter lens telescopes that can function as a virtual single 16.4 m diameter lens telescope. This idea is applied to small satellites, in which they use an adaptive formation control to act as a rigid structure to be used as an optical interferometer [19]. In aircraft formation flight, an example of sensor array is the radiant source seeker [7], which uses a sensor in each aircraft to estimate the spatial gradient of the measured variable.

There are several approaches to produce a formation. An usual classification of multi agent autonomous formation, which includes autonomous formation flight, is ilustred in Fig. 1.1 and is described below [20, 21, 14, 15, 22]

- **Leader-follower** [11, 12, 23]: an vehicle is defined as leader, and the others are defined as followers. The leader moves following some trajectory that can or can not be affected by the followers position or movement, i.e, the leader can be cooperative or uncooperative. The followers must track a trajectory defined relatively to the leader, which means that the followers' movement are affected by the leader's one. From a communication view, the leader follower can be seen as one agent broadcasting its data, or commands, to all other agents. As advantages, the leader-follower is one of the simplest approaches to implement. It also provides a way to describe precise formation shapes. However, if the leader presents some fault as, e.g, being shot down, the formation loses its reference.

- **Cascaded leader-follower** [13]: it is similar to the leader-follower approach, but it provides a tree hierarchy. A root agent is the leader, and a subset of the follower agents tracks this agent. These followers are the individual leaders of other subsets of agents. This is repeated until achieving the leaves of the tree, in which the agents follows other, but are not followed. This approach is useful, for example, if there is communication restrictions, and is more scalable than the single leader approach.

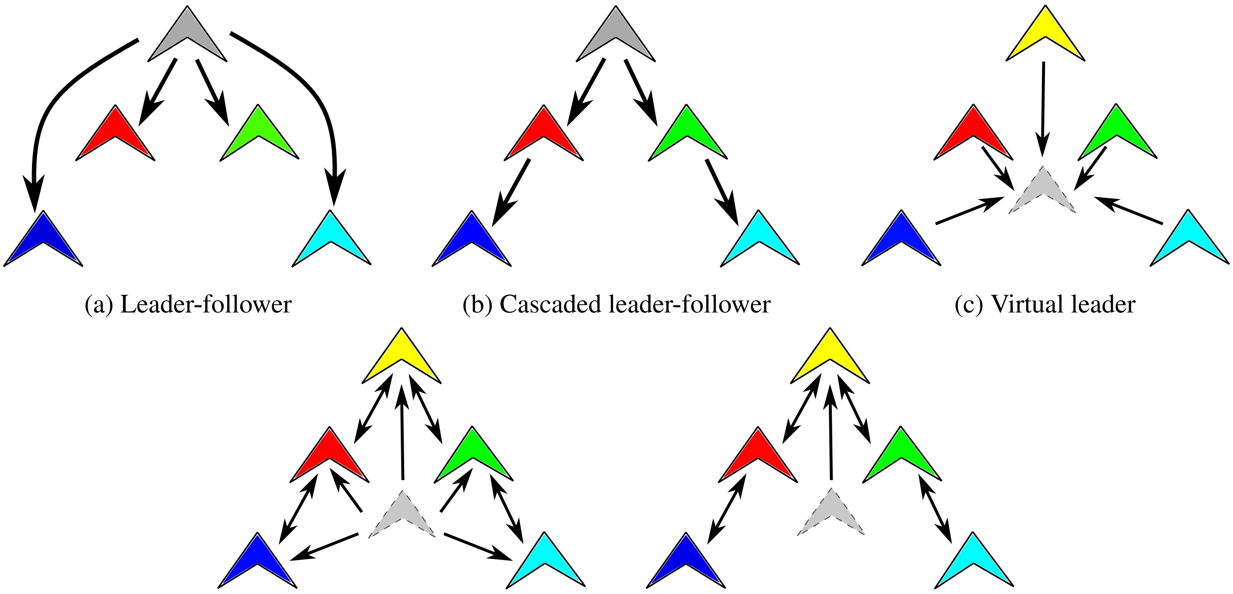
- **Virtual structure or virtual leader** [20]: a virtual entity is defined as leader. This entity position and/or orientation is known by all vehicles by some mechanism, such as consensus, known mathematical model, or because it is predefined. The control is virtually the same of the leader-follower, but it is not affected by communication restrictions. Also, compared to the leader-follower, there is no single point-of-failure: if a vehicle fails, the others still can obtain the virtual leader position. A feedback to the virtual leader can be implemented, which results in a cooperative virtual leader. If an agent is far from its desired position, for example because of a faulty actuator, the virtual leader, together with the other agents, can approach this agent. This is useful if maintaining formation shape is more important than each agent position itself in its absolute desired position. However, in this case the virtual leader's trajectory must be computed by all agents in real time, and requiring an agreement between all agents over the computed trajectory.

- **Synchronous** [24, 15, 14]: it is generalizations of the leader-follower and virtual leader approaches. Each agent not only aims to reduce its position error relatively to the (virtual) leader, but also aims to reduce the error difference between itself and its set of neighbors. The synchronous approach assumes that all agents have access to the leader's data, and each agent has access to its neighborhood data. The tracking errors synchronization allows the controller designer to weight between two control objectives: 1) each agent individually achieves its own desired position in the formation, without cooperation and 2) the agents move collectively to achieve the desired relative position between agents, i.e., the desired formation shape, even if this means that some agents will temporarily distance itself from their own desired individual position. In this way, the synchronous control achieves a similar effect of the virtual leader with feedback (cooperative virtual leader), but in a more distributed way, i.e., by only using the neighbor data and the pre-programmed virtual leader's trajectory.

- **Distributed** [25, 26, 22]: it is assumed that some of the agents does not have access to the leader's data. In this case, reducing the relative error between neighbors achieves a reduction in the the relative error between the agents and the (virtual) leader, even to the agents that does not directly communicates with the leader.

- **Behavioral, flocking** [26, 27] in this approach, a set of behaviors can be defined to each vehicle: formation keeping, collision ans obstacle avoidance, goal achieving, etc. The control action is obtained from a weighted mean of these behaviors. As explained in [20], this approach makes easier to derive a control strategies to competing objectives, but it is difficult to model the group behavior and to prove the group stability. Also, the obtained formation usually is amorphous [21], i.e., there is no explicit formation shape definition.

A cathegorization on how the formation is described and which sensors wold be necessary in au-



(a) Leader-follower

(b) Cascaded leader-follower

(c) Virtual leader

(d) Synchronized approach with virtual leader

(e) Distributed approach with virtual leader

Figure 1.1: Some formation flight styles classification.

tonomous formation control is presented by [21]

- **Position-based control**: each vehicle knows its own position in a global inertial reference frame, such as the North-East-Down frame. The formation is defined in this control scheme as a global (absolute) position to each vehicle, and each vehicle tracks its own desired position. Relative measurements can be included to increase performance or improve shape tracking.

- **Displacement-based control**: there is a global inertial reference system. The vehicles, however, must know only the orientation of the global frame, not its origin. The vehicles measure the displacement to the neighbors, where the displacement is defined as the the relative position, described in this global reference system. As example, the north and vertical direction can be known by all vehicles (by, e.g., using magnetometer and accelerometer), but not their relative position to the ground or to a non-neighbor leader aircraft. This control scheme controls the displacement to a desired value.

- **Distance-based control**: each vehicle known only its own body frame. The measured relative position to the neighbors can be described in the own vehicle body frame, but not in the global one. In this case, the control scheme tracks a desired distance, which is the magnitude of the relative position vector.

This classification is sensor based. The position based control needs a sensor that provides an absolute vector position of each agent, such as a global positioning system (GPS). The displacement-based control only needs a relative vector position between two agents, measured for example by a visual system (camera). The distance-based control depends only on the scalar distance, which can be obtained by, for example, by ultrasound or by evaluating the received power of a signal transmitted by the agent.

The leader-follower, virtual leader, synchronous and distributed can be implemented as a position-based control. As example, [14] presents a position-based synchronous autonomous formation flight controller, in which each UAV tracks an absolute position, but includes relative positions in the control law.

However, one usual approach in autonomous flight control is to use an approach similar, but distinct, to the displacement-based control. There is a global frame in which the formation is described but, instead of being inertial, it is based in a leader's frame, such as its wind frame. Being leader dependent, the frame orientation must be broadcast by the leader, or calculated from the virtual leader's pre-defined trajectory. Examples include [11, 4, 28, 29, 30].

The distance-based control can also be used in formation flight control. It is usually used in uncooperative or evasive scenario, such as missile tracking a target, or when a fleet does not communicate with each other to avoided being detected by enemy radio. In most cases, the measured and controlled variable is the line-of-sight between the controlled vehicle and its reference, and the control algorithm is the proportional navigation one or some of its variants [31, 32, 33].

The formations can also be classified by its shape [21]

- **Fixed shape**: the relative position between the agents remain fixed as, for example, a fleet that remains in a fixed V-shape during its flight.
- **Time-varying shape**: it is more general, allowing movement between the agents. For example, autonomous aerial refueling and rendezvous with a mothership can be seen as formations in which its shape varies in time.

1.2 AIRCRAFT CONTROL ARCHITECTURE

A control system is needed so that the UAV automatically make maneuvers, follow waypoints, and/or track desired paths. In formation flight, the control system achieves and maintains the formation.

A common way to implement a control system in a UAV is to break the problem in two or three nested loops [11, 4, 28, 14], in which the innermost loop deals directly with the aircraft model and its actuators, and each loop presents a more abstract model to the outer one. A three-loop configuration, containing the inner loop, the outer loop and the path planner, is shown in Fig. 1.2. It is shown also the sensors and estimator block, which provides feedback to the controllers.

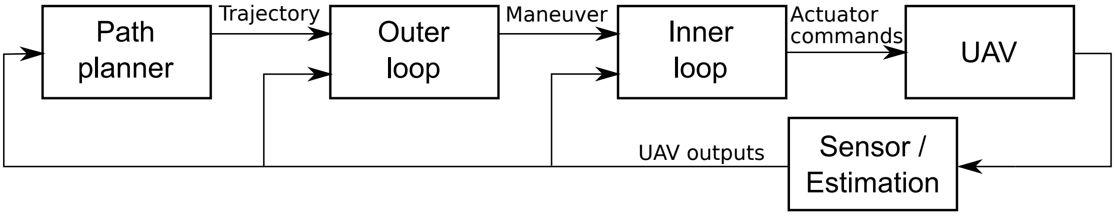


Figure 1.2: Three nested loops controlling a UAV.

The inner loop and outer loop are also known, respectively, as the control and guidance loops, whereas the estimation block is known as the navigation system. It is common in literature to work with these blocks together as the guidance, navigation and control (GNC) system [34]. The path planner is an optional block, specific to each problem, and usually outside of the control scope. Since estimation is outside the scope of this thesis, the estimation block is not discussed here.

1.2.1 Inner loop controller

The first loop is called the inner, stabilization, or control loop. There are some variants, but in all of these, the controller command the actuators of the UAV: the engine and three control surfaces: aileron, rudder, and elevator. In the most common variant, the controller aims to track a prescribed desired forward speed, desired altitude, and desired rate of turn [11, 4, 34] and, usually, to regulate the sideslip angle to zero [34, 28].

A common approach is to use linear PID (proportional-integral-derivative) controllers. This controller can be used even if the parameters of the UAV are not well known [34]. If the aircraft nonlinear model is known, a linearization of the model can be found, and the controller gains can be adjusted by using usual linear control techniques, such as root locus [11]. The linear controller is adequate to mildly maneuvers, but not to acrobatic ones [34].

The linear control techniques work as expected only if the system is near the operational point in which the system was linearized. Extra precision and performance can be achieved by using other control techniques, such sliding mode control, as in [35].

In this work, the linear PID inner loop controller presented by [34] is discussed and implemented in Chapter 2. The function of the inner loop, here, is only to provide a stabilized aircraft to the proposed outer loop controllers proposed by this thesis. In this way, the scope of this thesis resides mostly in the outer loop controller.

1.2.2 Outer loop controller

The guidance or outer loop receives as input the desired trajectory, which can be described in many ways, such as waypoints, a target position, or a 3D curve in space. Evaluating the state of the aircraft and the error between the desired trajectory and the true/estimated one, the outer loop generates reference inputs to the inner loop. A common simplifying assumption is that the trajectory control calculated by the outer loop is much slower than the inner loop control. In this way, from the time frame of the outer loop, the inner loop instantly corrects its controlled variables, uncoupling the inner loop dynamics from the outer loop one. Usually, single and multi-aircraft scenarios have distinct guidance algorithms.

Here are listed some examples of guidance algorithms. In the non-formation scenario, a common approach is to use pursuit-based algorithms, in which the line-of-sight between the aircraft and the waypoint is used to control the aircraft. In these cases, the control algorithm changes the course of the aircraft in a way that the flight direction is equal (or approaches) the line-of-sight. Other approach is to control directly the error between the aircraft actual position and the desired one. In multi-aircraft systems, usually is used a relative position (displacement-based control) or the absolute position (position-based control).

The proportional navigation and variants [31] are pursuit-based guidance algorithms. They maximize the chance of a vehicle achieve some moving point, called the target. Usually the vehicle is a missile aiming at an enemy vehicle, but can be, e.g., an aircraft going to encounter a flying mothership, an aerial refueling system, or fixed waypoint.

In [12], a pursuit-based guidance algorithm is used to follow waypoints. The algorithm assumes a straight line connecting the waypoints, and the aircraft makes an exponential approach to these lines, i.e, the distance between the aircraft and the trajectory reduces exponentially over time, without overshoot.

In [36, 34], the L_2^+ guidance algorithm is presented. This algorithm can approach any trajectory, but usually it assumes straight lines connecting the waypoints. It uses the GPS measured ground speed instead of Pitot-measured airspeed as input, to remove some of the wind perturbation effect. Also, it connect each straight line linking two waypoints with a turn.

In relative position methods, such as in [13, 37, 29, 11], it is assumed that relative positions between the aircraft and the target are measured or estimated, and the controller generates actuator commands to adjust these relative position. Similarly, in the absolute position methods [15, 14, 38, 26], it is assumed that the absolute positions are measured.

1.2.2.1 Autonomous formation flight controllers

Outer loop controllers can be used to implement autonomous formation flight.

When developing a formation flight controller, one approach is to use linear controllers, such as PID, to directly control the nonlinear system [12, 39]. However, linear controllers are tuned to an operational point. When a system moves away from this operational point, the performance can be degraded or even become unstable.

Other approach is to use a nonlinear dynamic inversion (NLDI) control strategy first. This technique applies a transformation in a nonlinear model to make it linear. If the transformation is applied in full state space, the transformation is complete, and the linear model correctly represents the system. It is worth to note that the NLDI is distinct to obtain a linearized approximation via Jacobian calculations.

After obtaining the linear model via NLDI, any linear control technique can be used, and the performance and stability are not degraded in the operational envelope in which the inversion is valid. Some examples include: a linear controller, with dynamics adjusted by pole placement, used to maintain a fixed formation shape [11], a linear distributed control [26] used to maintain a fleet of UAVs flying together in a loose shape formation, but contained inside a pre-determined 3D shape in space, a linear distributed differential game controller [38] used to achieve a control law that reduces a cost function that weights between reducing the formation tracking quadratic error and control signal energy consumption, and a robust synchronous linear $\mathcal{H}_\infty$ controller that guarantees the stability of the AFF control even in the presence of wind, measurements noise, and communication delays.

The NLDI, however, does not provide a perfectly linear model if there are model uncertainties. Because of this, the linear controllers performance can be degraded based on the amount of the modeling error. The sliding mode controller (SMC), an extension of the NLDI, is a robust nonlinear controller that provide the designed performance even in an inexact linearization, and it is robust to model uncertainties and bounded disturbances. As example of fixed wing AFF via SMC (or variants) are [13, 14]. An example of a rotary-wing formation control is [25] and a generic (nonspecied vehicle) multi agent autonomous formation control is [22]. Finally, examples of single agent use of SMC include [40, 41].

The SMC assumes that the disturbance is bounded, but is unknown and can vary from one extreme to another instantaneously. Because of this, the control output must overcome this disturbance, which is achieved by using a discontinuous, high bandwidth, high-magnitude chattering control signal, which mathematically is presented in the control law as a discontinuous signum function [13, 22].

Several techniques have been developed to circumvent this chattering effect, each one presenting a distinct trade-off. For example, the signum function can be changed to a similar but continuous one, such as hyperbolic tangent [27] or saturation [42, 25, 40]. This, however, inserts a trade-off between the chattering intensity and precision.

Another approach is the 2nd order SMC, in which an integrator is placed between the chattering signal and the plant. This approach significantly reduces chattering, maintaining the precision of the controller. As a drawback, it must assume that the derivative of the disturbance is bounded. A generalization of the 2nd order SMC is the low pass filter (LPF) SMC [43, 24, 41, 44, 14, 45], which provides a way to adjust between chattering smoothing and control bandwidth [41].

Autonomous formation flight controllers is the main topic of this thesis. Chapter 3 discusses nonlinear dynamic inversion based autonomous formation flight controllers, including a proposed controller as contribution. Chapter 4 discusses sliding mode based autonomous formation flight controllers, including two proposed controllers as contribution.

1.2.3 Path planner

The path planner is the outermost loop. As in guidance, there are single and multi-aircraft variants. It generates a path or a set of waypoints which the outer loop can follow. It usually uses some simplified model of the UAV as, e.g, that it moves as a Dubins vehicle [46], or that it makes clothoid arc turns [47]. The path planning is usually an optimization problem, as, e.g., to obtain the shortest path that brings the UAV from its initial position to the desired one while avoiding collision with static and moving obstacles and respecting UAV actuator limits [48] or to obtain paths to a set of cooperative UAVs in which they or make the mission with the least amount of time or with the least overall fuel consumption, where the mission is to attack a set of targets [6].

The path planning is implemented only when the trajectory must be defined in real time during the mission. Usually, the path or waypoits are preloaded in the aircraft, or are transmitted by a human pilot from a ground station.

Here, only a proof-of-concept path planning algorithm is implemented, in multi-aircraft scenario, to achieve a non-colliding rendezvous path between a set of cooperative UAV. It is explained in Chapter 3. Its main objective is to enrich the discussion about a proposed formation flight controller. The development and comparison between path planner algorithms is outside the scope of this thesis.

1.3 PROBLEM DEFINITION

The evaluated problem can be described as follows. There are at least two UAVs flying together, and they must achieve some prescribed formation, which can be of any shape, and this shape can vary with time. During the formation, the fleet can maneuver, for example, making a turn or increasing speed. A single UAV is defined as a leader, or alternatively, a virtual entity known as virtual leader can be computed in each UAV. All non-leader UAVs are defined as followers. The leader or virtual leader flies unaffected by the followers, i.e., is uncooperative. The formation is defined based on the leader position and/or orientation. In this way, the desired position of each follower is affected by the (virtual) leader's movement. It is assumed that all followers has access to the leader's data: either it broadcasts the data or it is a virtual leader implemented in the software of each follower. The followers can communicate bidirectionally only with a subset of the fleet. This limitation can be caused, for example, by a limited-band communication channel, or because some followers are too distant to others. The bidirectional communication graph between the followers is assumed known and fixed in each flight. The obtained controller must be robust to a bounded unknown disturbance and bounded unknown model imprecision. Only a theoretical development and computational simulations are made, but as a future work, it is expected that the obtained controller will be implemented and evaluated in the off-the-shelf model airplanes shown in Fig. 1.3 that are present in the Aerial Robotics Laboratory from the University of Brasília (UnB).

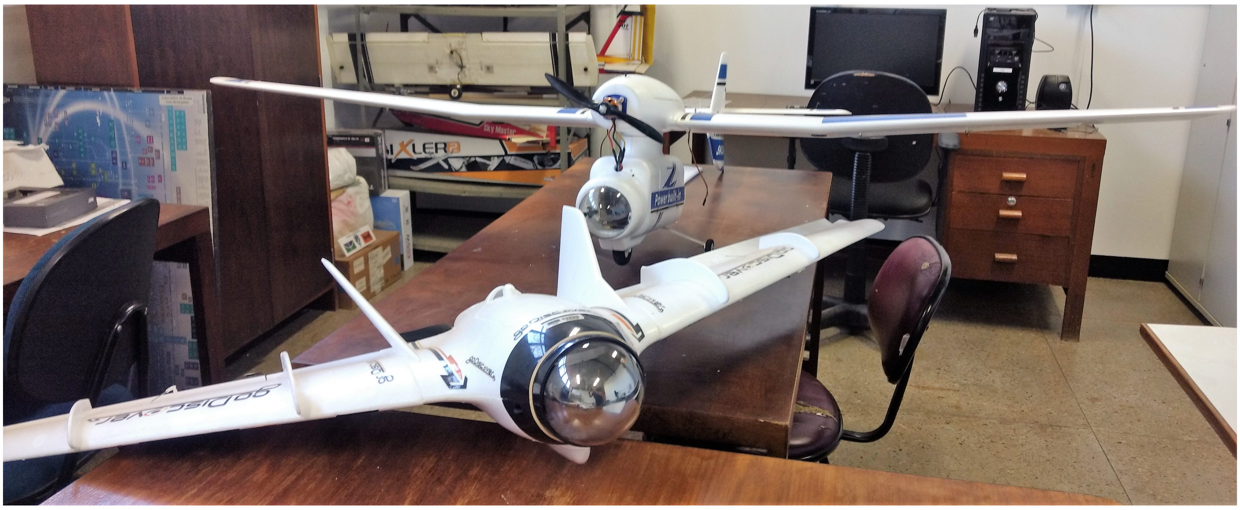


Figure 1.3: Model airplanes from the Aerial Robotics Laboratory - UnB.

1.4 CONTRIBUTIONS

The main contributions of this thesis are listed below

- **Development of a realistic fixed wing unmanned aerial vehicle simulator, including an inner loop controller**. It is used to evaluate the outer loop autonomous formation flight controllers.
- **A NLDI based autonomous formation flight controller is proposed**. It expands a controller from

literature [11] to be used in more scenarios, such as time-varying formation shape and aggressive maneuvers. The obtained controller is published in a conference [3].

- **A proof-of-concept path planner is presented**. It provides extra insight when evaluating the proposed NLDI based autonomous formation flight controller.

- **A recent published SMC based autonomous formation flight controller from literature [14] is evaluated, and it is demonstrated to be conceptually wrong**. The controller assumes a communication topology in which each UAV can only communicate to neighbors, however, the presented control law requires information from the entire fleet.

- **A new SMC based autonomous formation flight controller is proposed**. Differently from the literature controller, the proposed controller uses only the available data. It is also mathematically simpler. It is robust to bounded uncertainties and disturbances.

- **Another new SMC based autonomous formation flight controller is proposed**. It presents improvements compared to the previous proposition, being mathematically even simpler, and with improved transient response.

1.5 OUTLINE OF THIS WORK

In Chapter 2, the nonlinear aircraft model is reviewed and a realistic aircraft simulator is developed. Then, a simple PID inner loop is used, and its gains are tuned by trial and error via simulation. In Chapter 3, two distinct groups of formation flight dynamics modeling styles are reviewed, and a NLDI based autonomous formation flight is proposed. It is also shown that a time-varying formation shape, when described in a leader's reference frame, can be used as a way to easily prescribe a close range rendezvous between two or more aircraft and a proof-of-concept path planner and collision avoidance algorithm are developed. Simulations validate the proposed improved controller and path planner. In Chapter 4, the sliding mode controller is shown to be an extension of the nonlinear dynamic inversion that provides robustness. It is also explained that robustness can also be increased by adding communication between the followers aircraft, which generates the so called synchronous controller. A synchronous autonomous formation flight controller based in the low pass filter variant of the sliding mode controller shown in literature is reviewed. After, two controllers are proposed. Lyapunov candidate functions demonstrate the stability of the proposed controllers. Simulations shows that the proposed controller acts as expected.In Chapter 5, the conclusions of the work are presented. Finally, a list containing all publications made by the author that are related to the proposed thesis or that are product of disciplines taken during the PhD are presented.

2 AIRCRAFT MODEL

2.1 INTRODUCTION

In this Chapter, basic concepts are reviewed. The main objective is to describe each of the i-th aircraft from a fleet of N. In Section 2.2, the North-East-Down, the body and the wind coordinate frames are reviewed. It is reviewed also how to change the vector representation from one frame to another, and how to compute the derivative of vectors in distinct frames. In Section 2.3, the aircraft and actuator models are reviewed. In Section 2.4, the usual UAV nested controller loops and the point-mass model that represents the inner loop controlled aircraft are explained. After being tuned by trial and error, the inner loop is evaluated by simulation in Section 2.5. Finally, in Section 2.6, the conclusion of this Chapter is presented.

2.2 COORDINATE FRAMES

This Section discusses introductory topics in flight mechanics, and are included here for the sake of completeness. Detailed explanation can be found in textbooks such as [49, 50, 51].

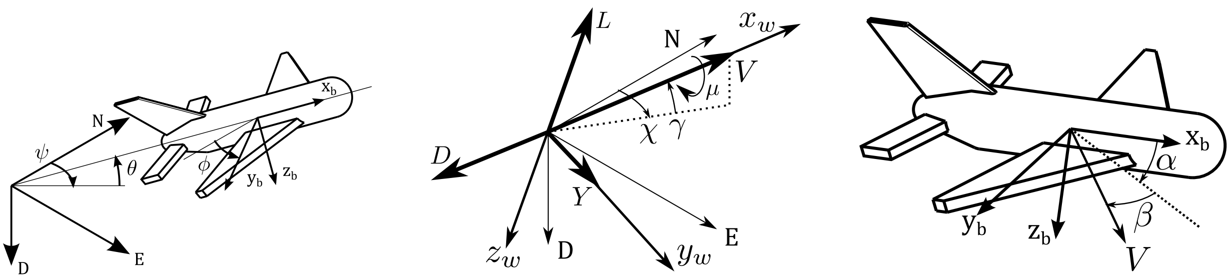
The North-East-Down frame S_{NED} is described by three orthonormal basis vectors that points, respectively, to the local north, local east and downward. The origin of this frame is usually positioned at a fixed point at the surface of the Earth, such as in a ground station or where the aircraft took-off.

The S_{NED} is well suited to develop and describe control strategies. Also, to describe small movements near the origin of the S_{NED}. Note that, since the Earth is round, the north, east and down vectors varies their direction based in the S_{NED} origin position.

The body frame S_{bi} has its origin positioned at the center of gravity (CG) of the i-th vehicle. The orthonormal vectors x_b, y_b and z_b points, respectively, to the nose of the aircraft, to its right, and to its bottom. The frames rotates with the aircraft. The S_{bi} frame can be used to describe the position of aircraft elements, such as sensors, antennas and actuators.

Assuming that the aircraft sensors are installed fixed to the aircraft body, they rotate with it. Because of this, sensors such as magnetometer, gyro, and accelerometer, that measures respectively magnetic field, angular velocity and acceleration, provides vector measurements that are best described in the S_{bi} frame. One exception is the GPS receiver that, because of a distinct measurement principle, measures position and velocity vectors described in an inertial frame such as S_{NED}.

The orientation of an aircraft related to some reference frame is called attitude [50]. By defining the reference frame as S_{NED} and by knowing that S_{bi} is related to the i-th aircraft, the attitude is defined as the relationship between both frames. The Euler angles $\psi_i(t)$, $\theta_i(t)$ and $\phi_i(t)$ relates S_{NED} to S_{bi} by applying respectively the yaw, pitch and roll rotations. Figure 2.1a shows both frames and its relationship. Note that the sequence of rotations 3-2-1, i.e., around respectively the z, y and finally the x axis must be respected.



(a) S_{NED}, $S_{b,i}$ and the sequence of Euler angle rotations $\psi_i(t)$, $\theta_i(t)$, $\phi_i(t)$ relating both frames.

(b) S_{NED}, and the sequence of Euler angle rotations $\chi_i(t)$, $\gamma_i(t)$ to achieve the $\boldsymbol{V}_i(t)$ direction, and $\mu_i(t)$, that rotates the lift vector around the velocity vector.

(c) S_{bi} and the sequence of Euler angle rotations $\alpha_i(t)$, $\beta_i(t)$ to achieve the $\boldsymbol{V}_i(t)$ direction.

Figure 2.1: Reference frames.

The wind frame S_{wi} is described in a similar fashion of S_{bi} and is shown in Fig. 2.1b. x_w is similar to x_b, but it aligns itself with the velocity vector of the aircraft related to the air mass whereas x_b aligns with the fuselage. The three Euler angles rotating from S_{NED} to S_w are $\chi_i(t)$, $\gamma_i(t)$ and $\mu_i(t)$, respectively the heading, flight path, and bank angles. The S_{wi} is mainly used to describe aerodynamic forces. Figure 2.1b shows the lift, drag and side force vectors, respectively $\boldsymbol{L}_i(t)$, $\boldsymbol{D}_i(t)$, and $\boldsymbol{Y}_i(t)$. Since lift is usually upwards and drag is always backwards, it is defined in literature (e.g. [50]) that positive values of the scalars L and D indicates vectors that are, respectively, in the negative direction of the x_w and z_w axes. The aerodynamic forces are further explained in Subsection 2.3.3. The wind frame can also be used to describe trajectories. In this case, the ground velocity vector, i.e., the velocity related to a fixed point in the ground is used instead of the air velocity vector. In a no wind scenario, the air velocity vector is equal to the ground velocity vector. Unfortunately, the two similar but distinct wind axes have the same name in literature.

There are two simplified versions of the wind frame. In both cases, they use the ground velocity vector (velocity related to the ground) instead of the air velocity vector. The first variant, which is defined here as S_{swi}, assumes that $\mu_i(t) = 0$ when rotating from S_{NED} to S_{swi}. This frame can be obtained by using the velocity vector only (instead of needing also the lift vector) and, by using the ground velocity, it is appropriate to describe trajectories. The other simplified wind frame, defined here as S_{χ_i}, assumes that both $\mu_i(t)$ and $\gamma_i(t)$ are null, is also defined related to the ground velocity vector, and describes the horizontal projection of the trajectory, i.e, the trajectory as seen from above. In the simplified versions of the wind frame, in which the ground velocity vector is used instead of the air velocity vector, $\chi_i(t)$ and $\gamma_i(t)$ can be obtained by

$$\tan\chi_i(t) = \frac{\dot{y}_i(t)}{\dot{x}_i(t)}, \qquad \sin\gamma_i(t) = -\frac{\dot{z}_i(t)}{V_i(t)}, \qquad V_i^2(t) = \dot{x}_i^2(t) + \dot{y}_i^2(t) + \dot{z}_i^2(t), \tag{2.1}$$

where $\boldsymbol{p}_{\text{NED},i} = [x_i(t)\ y_i(t)\ z_i(t)]^T$ is the position of the i-th aircraft in the S_{NED} frame.

The angles $\alpha_i(t)$ and $\beta_i(t)$, respectively angle of attack and sideslip angle, rotates from S_{bi} to S_{wi} when S_{wi} is defined related to the air velocity. These angles are shown in Fig. 2.1c. The angles $\alpha_i(t)$ and

$\beta_i(t)$ shows the airflow direction related to the fuselage. In a no wind scenario, the airflow direction is the same of the travel direction or ground velocity vector.

2.2.1 Vectors and coordinate frames

The coordinate frames are described by right-handed 3D orthonormal basis vector. A vector $\boldsymbol{v}(t)$, such as aircraft position or velocity, can be represented in any frame. Here a subscript shows in which frame the vector is represented. For example, $\boldsymbol{v}_{bi}(t)$ means that the vector is being represented in S_{bi} frame and $\boldsymbol{v}_{\mathrm{NED}}(t)$ means that the same vector is being represented in S_{NED}. For example, the aircraft velocity can, at same time, point forward in aircraft perspective, i.e., in S_b and point to East in S_{NED}.

The vector $\boldsymbol{v}_{\mathrm{NED}}(t)$ can be converted to $\boldsymbol{v}_{bi}(t)$ by using the direction cosine matrix (DCM) $\boldsymbol{D}_{bi}^{\mathrm{NED}}(t)$

$$\boldsymbol{v}_{bi}(t) = \boldsymbol{D}_{bi}^{\mathrm{NED}}(t)\boldsymbol{v}_{\mathrm{NED}}(t), \tag{2.2}$$

where $\boldsymbol{D}_{bi}^{\mathrm{NED}}(t)$ is obtained by applying the rotations generated by each Euler angle, represented by the matrices $\boldsymbol{D}_3(\psi_i(t))$, $\boldsymbol{D}_2(\theta_i(t))$, and $\boldsymbol{D}_1(\phi_i(t))$, respectively the yaw, pitch and roll rotations which are, respectively, rotation around the 3rd, 2nd and 1st axis.

$$\begin{aligned}
\boldsymbol{D}_{bi}^{\mathrm{NED}}(t) &= \boldsymbol{D}_1(\phi_i(t))\boldsymbol{D}_2(\theta_i(t))\boldsymbol{D}_3(\psi_i(t)) = \\
&\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_i(t) & \sin\phi_i(t) \\ 0 & -\sin\phi_i(t) & \cos\phi_i(t) \end{bmatrix}
\begin{bmatrix} \cos\theta_i(t) & 0 & -\sin\theta_i(t) \\ 0 & 1 & 0 \\ \sin\theta_i(t) & 0 & \cos\theta_i(t) \end{bmatrix}
\begin{bmatrix} \cos\psi_i(t) & \sin\psi_i(t) & 0 \\ -\sin\psi_i(t) & \cos\psi_i(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\
&\begin{bmatrix} \cos\psi_i(t)\cos\theta_i(t) & \sin\psi_i(t)\cos\theta_i(t) & -\sin\theta_i(t) \\
\cos\psi_i(t)\sin\theta_i(t)\sin\phi_i(t) - \sin\psi_i(t)\cos\phi_i(t) & \sin\psi_i(t)\sin\theta_i(t)\sin\phi_i(t) + \cos\psi_i(t)\cos\phi_i(t) & \cos\theta_i(t)\sin\phi_i(t) \\
\cos\psi_i(t)\sin\theta_i(t)\cos\phi_i(t) + \sin\psi_i(t)\sin\phi_i(t) & \sin\psi_i(t)\sin\theta_i(t)\cos\phi_i(t) - \cos\psi_i(t)\sin\phi_i(t) & \cos\theta_i(t)\cos\phi_i(t) \end{bmatrix}.
\end{aligned} \tag{2.3}$$

The superscript/subscript notation is chosen to resemble a fraction, in which the superscript acts as a numerator and the subscript acts as denominator. In this way, the superscript from the DCM cancels the subscript from the vector, and the resulting vector contains the non-canceled subscript from the DCM.

The inverse operation, i.e., obtaining $\boldsymbol{v}_{\mathrm{NED}}(t)$ from to $\boldsymbol{v}_b(t)$ is obtained similarly by

$$\boldsymbol{v}_{\mathrm{NED}}(t) = \boldsymbol{D}_{\mathrm{NED}}^{bi}(t)\boldsymbol{v}_b(t), \tag{2.4}$$

where

$$\boldsymbol{D}_{\mathrm{NED}}^{bi}(t) = (\boldsymbol{D}_{bi}^{\mathrm{NED}}(t))^{-1} = (\boldsymbol{D}_{bi}^{\mathrm{NED}}(t))^T, \tag{2.5}$$

Similarly, the DCM $\boldsymbol{D}_{wi}^{\mathrm{NED}}(t)$ between S_{NED} and S_{wi} is given by

$$\boldsymbol{D}_{wi}^{\mathrm{NED}}(t) = \boldsymbol{D}_1(\mu_i(t))\boldsymbol{D}_2(\gamma_i(t))\boldsymbol{D}_3(\chi_i(t)), \tag{2.6}$$

the DCM $\boldsymbol{D}_{swi}^{\mathrm{NED}}(t)$ between S_{NED} and S_{swi} is given by

$$\boldsymbol{D}_{swi}^{\mathrm{NED}}(t) = \boldsymbol{D}_2(\gamma_i(t))\boldsymbol{D}_3(\chi_i(t)) = \begin{bmatrix} \cos\gamma_i(t)\cos\chi_i(t) & \cos\gamma_i(t)\sin\chi_i(t) & -\sin\gamma_i(t) \\ -\sin\chi_i(t) & \cos\chi_i(t) & 0 \\ \sin\gamma_i(t)\cos\chi_i(t) & \sin\gamma_i(t)\sin\chi_i(t) & \cos\gamma_i(t) \end{bmatrix}, \tag{2.7}$$

and the DCM $\boldsymbol{D}_{\chi i}^{\mathrm{NED}}(t)$ between S_{NED} and $S_{\chi i}$ is given by

$$\boldsymbol{D}_{\chi i}^{\mathrm{NED}}(t) = \boldsymbol{D}_3(\chi_i(t)) = \begin{bmatrix} \cos\chi_i(t) & \sin\chi_i(t) & 0 \\ -\sin\chi_i(t) & \cos\chi_i(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.8}$$

Similar operations can be used to convert any vector, from any to any coordinate system. Since all coordinate systems are orthonormal, the inverse of the DCM is equal to the transpose of the DCM. A brief discussion about using non-orthonormal coordinate system can be seen in [52].

The Euler angles varies in time. This means that a frame can rotate related to other. The angular velocity of a S_x frame related to a S_y frame, described in the S_z frame axis is given by $\boldsymbol{\omega}_z^{xy}$. For example

$$\boldsymbol{\omega}_{bi}^{bi,\mathrm{NED}}(t) = \begin{bmatrix} \dot{\phi}_i(t) - \dot{\psi}_i(t)\sin\theta_i(t) \\ \dot{\theta}_i(t)\cos\phi_i(t) + \dot{\psi}_i(t)\cos\theta_i(t)\sin\phi_i(t) \\ \dot{\psi}_i(t)\cos\theta_i(t)\cos\phi_i(t) - \dot{\theta}_i(t)\sin\phi_i(t) \end{bmatrix}, \tag{2.9}$$

$$\boldsymbol{\omega}_{swi}^{swi,\mathrm{NED}}(t) = \begin{bmatrix} -\dot{\chi}_i(t)\sin\gamma_i(t) & \dot{\gamma}_i(t) & \dot{\chi}_i(t)\cos\gamma_i(t) \end{bmatrix}^T, \tag{2.10}$$

$$\boldsymbol{\omega}_{\chi i}^{\chi i,\mathrm{NED}}(t) = \begin{bmatrix} 0 & 0 & \dot{\chi}_i(t) \end{bmatrix}^T, \tag{2.11}$$

where $\boldsymbol{\omega}_{bi}^{bi,\mathrm{NED}}(t)$ is the angular velocity of S_{bi} related to S_{NED} and represented in the S_{bi} frame, $\boldsymbol{\omega}_{swi}^{swi,\mathrm{NED}}(t)$ is the angular velocity of S_{swi} related to S_{NED} and represented in the S_{swi} frame, and $\boldsymbol{\omega}_{\chi i}^{\chi i,\mathrm{NED}}(t)$ is the angular velocity of $S_{\chi i}$ related to S_{NED} and represented in the $S_{\chi i}$ frame. Figure 2.2 shows $\boldsymbol{\omega}_{bi}^{bi,\mathrm{NED}}(t) = [P_i(t)\ Q_i(t)\ R_i(t)]^T$. It is worth to note that the derivative of the attitude angles and the angular velocity $\boldsymbol{\omega}_{bi}^{b,\mathrm{NED}}(t)$ are distinct concepts. This can be seen by comparing Fig. 2.2, which shows the angular velocity definition, with Fig. 2.1a, with shows the attitude angles definition.

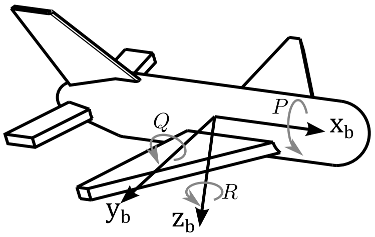


Figure 2.2: Angular velocity of the aircraft $\boldsymbol{\omega}_{bi}^{bi,\mathrm{NED}}(t) = [P_i(t)\ Q_i(t)\ R_i(t)]^T$, described in S_{bi}

The time-derivative of vectors are affected by the frame used to represent the vector. For example, a sensor fixed in a rigid aircraft has a constant position if this position is described in S_{bi} but, since the aircraft can be moving and/or rotating, it has a variable position if this position is defined in S_{NED}. In this way, the sensor velocity is null or not null depending on the used frame. The resulting derivative is a vector and can be represented in a distinct frame from which the derivative was made. For example, the aircraft velocity $\boldsymbol{V}_{bi}(t)$ is the variation in position that occurs in the S_{NED} inertial frame, represented in S_{bi}. In other words, $\boldsymbol{V}_{bi}(t)$ is the inertial velocity as seen by the aircraft.

The Theorem of Coriolis [50] can be used to convert the derivative made in some frame S_x to the derivative made in another frame S_y

$$\left.\frac{d}{dt}\boldsymbol{v}_z(t)\right|_{S_y} = \left.\frac{d}{dt}\boldsymbol{v}_z(t)\right|_{S_x} + \boldsymbol{\omega}_z^{xy}(t) \times \boldsymbol{v}_z(t), \tag{2.12}$$

where the subscript z in all vectors indicates that the vectors are represented in S_z frame, and the subscripts S_x and S_y in the derivatives shows in which frame the derivative is made. Usually S_z is chosen to be S_x, but it can be S_y or any unrelated frame.

2.3 AIRCRAFT MODEL

For simulation and algorithm evaluation purposes, it is used both the equations and the parameter values of the nonlinear 6 degress-of-freedom (6-DoF) model from the YF-22 model aircraft from West Virginia University, described in [11, 53]. The focus of this thesis is to develop formation flight controllers, and the use of the presented model is mainly to validate the controller in a realistic simulation. The implementation of the developed controllers in a real model aircraft is not at the scope of this thesis, but is a desired future work.

Table 2.1 contains the main characteristics of the YF-22 model aircraft. The thrust is generated by a miniature turbine engine.

Table 2.1: YF-22 aircraft model specification

Parameter	Symbol	Value
Mass	m	20.64 Kg (with 60% fuel capacity)
Mean aerodynamic chord	$\bar{c}$	0.76 m
Wing span	b	1.96 m
Wing area	S	1.37 m
Thrust	T	up to 125 N
Cruise speed	-	42 m/s

Here it is presented the nonlinear state model of the aircraft. This description is well known in the literature, and is presented here for the sake of completeness. They are general equations, i.e., they are not specific to the YF-22 model aircraft, and can be used to simulate any model aircraft, assuming that the necessary coeficients are identified by some technique. Detailed deductions of these equations can be found, for example, in [50]. A detailed step-by-step in the identification of the aircraft aerodynamics and inertial coefficients can be found in, e.g., [11, 54]

Usually, the aircraft model in separated in several blocks, listed as follows and shown in Figure 2.3.

- **Actuators**: Includes the propulsion and control surfaces.
  - **Propulsion**: Describes the forces, moments and/or energy provided by the propulsion system, and its dynamics.
  - **Control surfaces**: Describes the dynamics of the control surfaces: aileron, rudder and elevator.
- **Environment**: Models elements not related to the aircraft, but that affects its flight, such as air density, presence of wind and wind gusts, and gravitational acceleration.
- **Aerodynamics**: Provides the aerodynamics forces and moments based in the control surface positions, environment and the state of the 6-DoF point mass.
- **6 Degrees of Freedom (6-DoF) point mass**: It is a generic point-mass model representing the CG of the aircraft. Receives as input a force and a torque (momentum) vector. The aircraft mass and matrix inertia converts respectively the force to an acceleration vector and the torque to an angular acceleration. By integrating (angular) accelerations, it is obtained (angular) velocities and position (attitude).

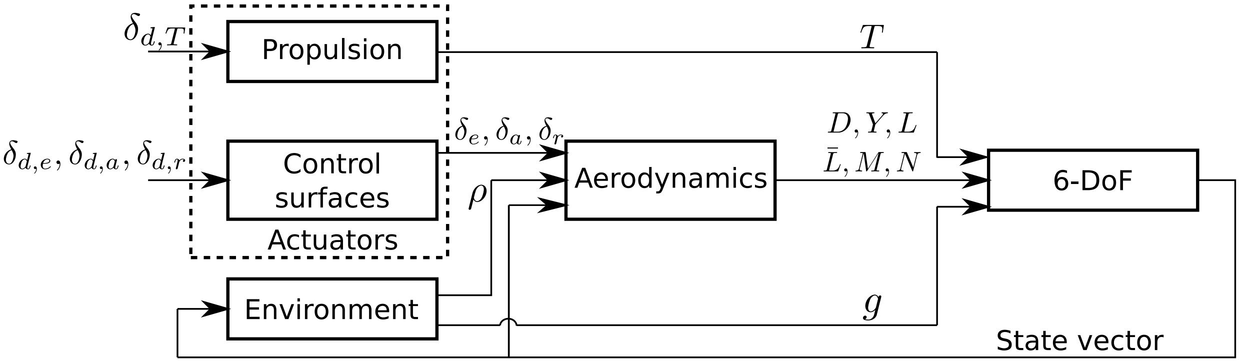


Figure 2.3: The nonlinear aircraft model.

The propulsion and the three control surfaces act as actuators, receiving as control inputs $\boldsymbol{u}_i(t)$ = $[\delta_{d,T,i}(t)\ \delta_{d,a,i}(t)\ \delta_{d,e,i}(t)\ \delta_{d,r,i}(t)]^T$, where $\delta_{d,T,i}(t)$ is the desired engine throttle, and $\delta_{d,a,i}(t)$, $\delta_{d,e,i}(t)$, and $\delta_{d,r,i}(t)$ are respectively the desired aileron, elevator, and rudder angular position for the i-th aircraft. The actuators are shown in Fig. 2.4. An aircraft can have more control surfaces, such as flaps and/or spoilers. However, usually they are used to adjust the aircraft to specific situations, such as take-off and landing. When used as control inputs, their behavior can usually be described similarly as the aileron, rudder and/or elevator surfaces.

The 6-DoF model provides a 12-state vector $\boldsymbol{\xi_i}(t)$ = $[V_i(t)\ \alpha_i(t)\ \beta_i(t)\ P_i(t)\ Q_i(t)\ R_i(t)\ \phi_i(t)\ \theta_i(t)\ \psi_i(t)\ x_i(t)\ y_i(t)\ z_i(t)]^T$, in which $V_i(t)$ is the airspeed magnitude, $\alpha_i(t)$ and $\beta_i(t)$ defines the velocity direction related to the fuselage (Fig. 2.1c), $\boldsymbol{\omega}_{b,i}^{bi,\mathrm{NED}}(t)$ = $[P_i(t)\ Q_i(t)\ R_i(t)]^T$ is the angular velocity described in S_{bi} (Fig. 2.2), $\phi_i(t)$, $\theta_i(t)$ and $\psi_i(t)$ are the attitude Euler angles (Fig. 2.1a), and $\boldsymbol{p}_{\mathrm{NED},i}$ = $[x_i(t)\ y_i(t)\ z_i(t)]^T$ are positions measured in S_{NED} relative to some origin as, e.g., the take-off point.

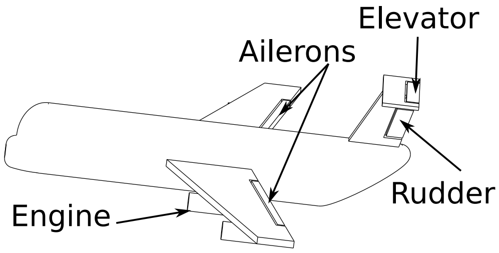


Figure 2.4: Actuators

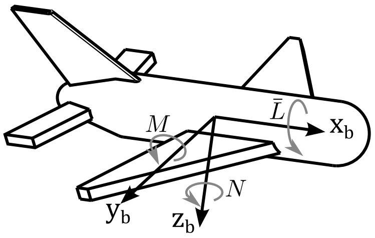


Figure 2.5: Moments

2.3.1 Actuators

The propulsive system main objective is to provide energy to the aircraft movement, which constantly loses energy mainly to the resistive aerodynamic drag force. The exceeding energy become kinetic and/or potential energy [49, 4], i.e., is used to gain speed and/or altitude. If the drag energy loss is greater than the propulsive added one, the vehicle loses energy, which means that it must lose speed and/or altitude. Alternatively, the propulsive system can be seen as a provider of the thrust force, which accelerates the aircraft forward if the force is greater than forces in opposite direction, mainly the component of the drag force. The extra velocity implies in extra generated lift in the wings, resulting in an altitude gain.

The throttle $\delta_{T,d,i}(t)$ is how is called the commanded input of a propulsive system, and is usually a value between 0 and 1. The propulsive system can be a jet engine or a propeller, which can be attached to an internal combustion engine or electric motor. The throttle of a jet engine propulsion system commands a thrust force output. This means that the jet engine provides a constant force if the input is a constant throttle [49]. The throttle in a propelled propulsive system commands a power output. This means that, for a constant throttle input, the output is a constant power [49]. Since the relationship between power P and force is given by

$$P(t) = F(t)V(t), \tag{2.13}$$

this means that a propeller provides less force if the aircraft speed is higher. Also, if the power of the internal combustion engine or electric motor $P_e(t)$ that propels the propeller is known, and the efficiency η of the engine plus propeller is also known, the provided thrust is

$$T(t) = P_e(t)\eta/V(t). \tag{2.14}$$

The spinning propeller or an unbalanced thrust output in a twin jet engine generates torque. A propeller in front of the wing creates extra airflow over the wing, increasing lift. Here, by simplicity, it is assumed that these effects are negligible. It is also assumed that the force generated by the propulsive system is perfectly aligned with the fuselage, i.e., with the x_b axis.

Finally, the propulsion system does not respond instantaneously. There is a time delay $t_{d,T}$ that accounts to command transmission and, in jet engine, fluid transport delay. The dynamics can be approximated by a first order transfer function with time constant τ_T. The jet engine dynamic model from [11] is given by

$$\frac{T(s) - T_b(s)}{\delta_T(s)} = \frac{K_T}{1 + \tau_T s} e^{-t_{d,T} s}, \tag{2.15}$$

where τ_T = 0.25 s and $t_{d,T}$ = 0.26 s. Note that this model includes also a conversion from a provided throttle δ_T to the obtained thrust T, which uses a gain K_T = 0.624 and an offset $T_b = -25.86$ N.

The steady state relationship between the throttle and thrust of this particular jet engine is given by

$$T(t) = K_T \delta_T(t) + T_b. \tag{2.16}$$

The control surfaces changes the format of their airfoil (e.g. wing), generating lift-like forces, which generates moments [50].

The aileron pair moves in opposite direction. This increases the lift in one of the wings, and reduces in another. The net force is null, but it creates a roll moment $\bar{L}(t)$. In this way, ailerons are used to roll the aircraft. It is also used to make turns via the coordinated turn, as explained in subsection 2.4.2. When using the aileron, the wing with higher lift has higher drag, whereas the opposite occurs in the other. This generates a small undesired yaw moment $N(t)$.

The rudder generates a horizontal force in the back of the aircraft. The high lever arm between the CG and the force generates a yaw moment N. As side effect, there is a small lever arm upward, that generates a small undesired rolling moment $\bar{L}$. It also increases slightly the drag. The rudder is used mainly to reduce sideslip. The rudder can also be used to make turns, but is used mainly when rolling the aircraft is undesired, such as in take-off and landing. The coordinated turn via roll provide sharper turns with better aerodynamic efficiency.

The elevator generates a vertical force in the back of the aircraft. This creates a pitching moment M, and increases slightly the drag. The elevator changes between kinetic and potential energy, and is used to gain or lose altitude and, as side effect, lose or gain speed.

As the engine, the control surfaces has a dynamic model. The dynamics of servo motor tracking the received command to change desired control surface position δ_{surface} can be approximated by a first order transfer function with time constant τ_a. There is also a time delay $t_{d,a}$ to account to the autopilot processing time. The transfer function between δ_{surface} and $\delta_{d,\text{surface}}$ is given by.

$$\frac{\delta_{\text{surface}}(s)}{\delta_{\text{commanded}}(s)} = \frac{1}{1 + \tau_a s} e^{-t_{d,a} s}, \tag{2.17}$$

where [11] obtained τ_a = 0.04 s and delay $t_{d,a}$ = 0.02 s for all control surfaces of its aircraft.

Both the engine and control surfaces have operational limits, that are enforced in the simulation. Table 2.2 shows the used actuator limits.

Table 2.2: YF-22 actuator limits

Actuator	Lower limit	Upper limit
Engine	70 ($\approx$ 10 N)	255 ($\approx$ 133 N)
Control surfaces*	-8°	8°

* [11] does not include the control surface limits. These values are defined arbitrarily here.

It is worth to note that the dynamics of four actuators requires a total of four states, representing the four actuator positions. However, since the actuator dynamics are uncoupled to each other and to the point-mass dynamics, historically these states are not explicitly included in the aircraft nonlinear dynamic model.

2.3.2 Environment

The environment proprieties are the gravity vector, air density and stochastic properties of the wind gusts.

The gravity vector $\boldsymbol{g}$ magnitude is proportional to the inverse of the squared distance between the aircraft and the center of the Earth [50], and it points downward. However an usual commercial aircraft, as exemplified in [49], has a service ceiling altitude of about 15 km, which is a small amount compared to Earth's radius, which is about 6371 km. A small UAV can only fly at a much slower altitude. In this way, the magnitude of $\boldsymbol{g}$ is assumed constant, and $\boldsymbol{g}_{\text{NED}} = [0\ 0\ 9.81]\text{m/s}^2$ when represented in S_{NED}.

The air density ρ can be approximately obtained by inputting the altitude of the aircraft in the standard atmosphere model [49]. For simplicity, here it is assumed that the air density is constant and equal to $\rho = 1.225\text{kg/m}^3$, which is the value obtained at sea level.

For simplicity, it is assumed a scenario with no wind or wind gusts.

2.3.3 Aerodynamic model

The aerodynamic model calculates the aerodynamic force and moment vectors. The components of the aerodynamic force are drag $D_i(t)$, lift $L_i(t)$ and side force $Y_i(t)$, which are aligned with the axes from S_{wi} and are shown in Fig. 2.1b. The moment or torque components are $\bar{L}_i(t)$, $M_i(t)$ and $N_i(t)$, where the bar in $\bar{L}$ is used to differentiate from the lift force, as is usual in literature. The moments are described in S_{bi} and shown in Fig. 2.5. The result is calculated by a nonlinear function of the geometry of the aircraft, airflow direction, air density and the input vector $\boldsymbol{u}_i(t)$.

First, the dimensionless coefficients are calculated. $C_{D,i}(t)$, $C_{L,i}(t)$, $C_{Y,i}(t)$ are respectively the drag, lift and side aerodynamic forces coefficients, and $C_{l,i}(t)$, $C_{m,i}(t)$, and $C_{n,i}(t)$ are respectively the rolling, pitching and yawing aerodynamic moments coefficients. These coefficients provides the aerodynamic forces and moments applied to the aircraft and are nonlinear functions of the geometry of the aircraft and the input $\boldsymbol{u}_i(t)$. Assuming an operation point, the nonlinear functions can be approximated to affine functions

$$\begin{bmatrix} C_{D,i}(t) \\ C_{L,i}(t) \\ C_{m,i}(t) \end{bmatrix} = \begin{bmatrix} C_{D0} & C_{D\alpha} & C_{Dq} & C_{D\delta e} \\ C_{L0} & C_{L\alpha} & C_{Lq} & C_{L\delta e} \\ C_{m0} & C_{m\alpha} & C_{mq} & C_{m\delta e} \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_i(t) \\ \frac{\bar{c}}{2V_i(t)} Q_i(t) \\ \delta_{e,i}(t) \end{bmatrix}, \tag{2.18a}$$

$$\begin{bmatrix} C_{Y,i}(t) \\ C_{l,i}(t) \\ C_{n,i}(t) \end{bmatrix} = \begin{bmatrix} C_{Y0} & C_{Y\beta} & C_{Yp} & C_{Yr} & C_{Y\delta a} & C_{Y\delta r} \\ C_{l0} & C_{l\beta} & C_{lp} & C_{lr} & C_{l\delta a} & C_{l\delta r} \\ C_{n0} & C_{n\beta} & C_{np} & C_{nr} & C_{n\delta a} & C_{n\delta r} \end{bmatrix} \begin{bmatrix} 1 \\ \beta_i(t) \\ \frac{b}{2V_i(t)} P_i(t) \\ \frac{b}{2V_i(t)} R_i(t) \\ \delta_{a,i}(t) \\ \delta_{r,i}(t) \end{bmatrix}, \tag{2.18b}$$

where the mean aerodynamic chord $\bar{c}$ is given in Table 2.1, and the C_{ab} coefficients are called aerodynamics derivative, and models the linear contribution of the state or actuator position b in the aerodynamic coefficient C_a. Note, however, that nonlinear effects present in high-intensity maneuvers are not modeled. For example, the aircraft initially increases C_L when α increases, but there is a sudden decrease in C_L if α increases too much, resulting in loss of control of aircraft and an fast decrease in altitude. This effect, known as stall, is not included in (2.18). It is worth note that the simulations does not include acrobatic maneuver. Also, the simulation uses saturation functions to limit the commanded roll and pitch angles, limiting the maneuvering under a non-stall region.

The dimensionless coefficients are now converted to forces and momentum

$$D_i(t) = \bar{q}_i(t) S C_{D,i}(t), \quad L_i(t) = \bar{q}_i(t) S C_{L,i}(t), \quad Y_i(t) = \bar{q}_i(t) S C_{Y,i}(t), \tag{2.19a}$$

$$\bar{L}_i(t) = \bar{q}_i(t) S b C_{l,i}(t), \quad M_i(t) = \bar{q}_i(t) S \bar{c} C_{m,i}(t), \quad N_i(t) = \bar{q}_i(t) S b C_{n,i}(t), \tag{2.19b}$$

where the wing span b and area S are given in Table 2.1 and

$$\bar{q}_i(t) = 0.5 \rho V_i^2(t). \tag{2.20}$$

2.3.4 6-DoF model

The equations presented here are obtained from [11, 50, 53]. Detailed deductions of these equations can be found, for example, in [50].

$$\begin{aligned} \dot{V}_i(t) = {} & \frac{1}{m_i} \left(-D_i(t) \cos\beta_i(t) + Y_i(t) \sin\beta_i(t) + T_i(t) \cos\alpha_i(t) \cos\beta_i(t) \right) \\ & - g \left(\sin\theta_i(t) \cos\alpha_i(t) \cos\beta_i(t) - \cos\theta_i(t) \sin\phi_i(t) \sin\beta_i(t) \right. \\ & \left. - \cos\theta_i(t) \cos\phi_i(t) \sin\alpha_i(t) \cos\beta_i(t) \right), \end{aligned} \tag{2.21a}$$

$$\begin{aligned} \dot{\alpha}_i(t) = {} & \frac{1}{m_i V_i(t) \cos\beta_i(t)} \left[-L_i(t) - T_i(t) \sin\alpha_i(t) + m_i g \left(\cos\theta_i(t) \cos\phi_i(t) \cos\alpha_i(t) \right. \right. \\ & \left. \left. + \ \sin\theta_i(t) \sin\alpha_i(t) \right) \right] + Q_i(t) - \left(P_i(t) \cos\alpha_i(t) + R_i(t) \sin\alpha_i(t) \right) \tan\beta_i(t), \end{aligned} \tag{2.21b}$$

$$\begin{aligned}\dot{\beta}_i(t) = &\frac{1}{m_i V_i(t)}\left[D_i(t)\sin\beta_i(t) + Y_i(t)\cos\beta_i(t) - T_i(t)\cos\alpha_i(t)\sin\beta_i(t)\right.\\ &+ m_i g\left(\sin\theta_i(t)\cos\alpha_i(t)\sin\beta_i(t) + \cos\theta_i(t)\sin\phi_i(t)\cos\beta_i(t)\right.\\ &\left.\left. -\ \cos\theta_i(t)\cos\phi_i(t)\sin\alpha_i(t)\sin\beta_i(t)\right)\right] + P_i(t)\sin\alpha_i(t) - R_i(t)\cos\alpha_i(t),\end{aligned} \tag{2.21c}$$

$$\begin{bmatrix}\dot{P}_i(t)\\ \dot{Q}_i(t)\\ \dot{R}_i(t)\end{bmatrix} = \boldsymbol{M}_1\begin{bmatrix}P_i^2(t)\\ Q_i^2(t)\\ R_i^2(t)\end{bmatrix} + \boldsymbol{M}_2\begin{bmatrix}Q_i(t)R_i(t)\\ P_i(t)R_i(t)\\ P_i(t)Q_i(t)\end{bmatrix} + \boldsymbol{M}_0\begin{bmatrix}\bar{L}_i(t)\\ M_i(t)\\ N_i(t)\end{bmatrix}, \tag{2.21d}$$

$$\dot{\theta}_i(t) = Q_i(t)\cos\phi_i(t) - R_i(t)\sin\phi_i(t), \tag{2.21e}$$

$$\dot{\psi}_i(t) = \sec\theta_i(t)\left(Q_i(t)\sin\phi_i(t) + R_i(t)\cos\phi_i(t)\right), \tag{2.21f}$$

$$\dot{\phi}_i(t) = P_i(t) + \tan\theta_i(t)\left(Q_i(t)\sin\phi_i(t) + R_i(t)\cos\phi_i(t)\right), \tag{2.21g}$$

$$\begin{aligned}\dot{x}_i(t) = &V_i(t)\left[\cos\beta_i(t)\cos\alpha_i(t)\cos\theta_i(t)\cos\psi_i(t) + \sin\beta_i(t)\left(\sin\phi_i(t)\sin\theta_i(t)\cos\psi_i(t)\right.\right.\\ &\left. -\ \cos\phi_i(t)\sin\psi_i(t)\right) + \cos\beta_i(t)\sin\alpha_i(t)\left(\cos\phi_i(t)\sin\theta_i(t)\cos\psi_i(t)\right.\\ &\left.\left. +\ \sin\phi_i(t)\sin\psi_i(t)\right)\right],\end{aligned} \tag{2.21h}$$

$$\begin{aligned}\dot{y}_i(t) = &V_i(t)\left[\cos\beta_i(t)\cos\alpha_i(t)\cos\theta_i(t)\sin\psi_i(t) + \sin\beta_i(t)\left(\sin\phi_i(t)\sin\theta_i(t)\sin\psi_i(t)\right.\right.\\ &\left. + \cos\phi_i(t)\cos\psi_i(t)\right)\cos\beta_i(t)\sin\alpha_i(t)\left(\cos\phi_i(t)\sin\theta_i(t)\sin\psi_i(t)\right.\\ &\left.\left. -\ \sin\phi_i(t)\cos\psi_i(t)\right)\right],\end{aligned} \tag{2.21i}$$

$$\dot{z}_i(t) = -V_i(t)\left(\cos\beta_i(t)\cos\alpha_i(t)\sin\theta_i(t) - \sin\beta_i(t)\sin\phi_i(t)\cos\theta_i(t)\right. \tag{2.21j}$$

$$\left. -\ \cos\beta_i(t)\sin\alpha_i(t)\cos\phi_i(t)\cos\theta_i(t)\right). \tag{2.21k}$$

where $\boldsymbol{M}_0$, $\boldsymbol{M}_1$, and $\boldsymbol{M}_2$ are 3×3 matrices defined from the aircraft inertial matrix $\boldsymbol{J}$

$$\boldsymbol{M}_0 = \frac{1}{\det(\boldsymbol{J})}\begin{bmatrix}J_yJ_z - J_{yz}^2 & J_{xy}J_z + J_{yz}J_{xz} & J_{xy}J_{yz} + J_yJ_{xz}\\ J_{xy}J_z + J_{yz}J_{xz} & J_xJ_z - J_{xz}^2 & J_{yz}J_x + J_{xy}J_{xz}\\ J_{xy}J_{yz} + J_yJ_{xz} & J_{yz}J_x + J_{xy}J_{xz} & J_xJ_y - J_{xy}^2\end{bmatrix}, \tag{2.22a}$$

$$\boldsymbol{M}_1 = \boldsymbol{M}_0\begin{bmatrix}0 & J_{yz} & -J_{yz}\\ -J_{xz} & 0 & J_{xz}\\ J_{xy} & -J_{xy} & 0\end{bmatrix}, \tag{2.22b}$$

$$\boldsymbol{M}_2 = \boldsymbol{M}_0\begin{bmatrix}J_y - J_z & -J_{xy} & -J_{xz}\\ J_{xy} & J_z - J_x & -J_{yz}\\ -J_{xz} & J_{xz} & J_x - J_y\end{bmatrix}, \tag{2.22c}$$

$$\boldsymbol{J} = \begin{bmatrix}J_x & -J_{xy} & -J_{xz}\\ -J_{xy} & J_y & -J_{yz}\\ -J_{xz} & -J_{yz} & J_z\end{bmatrix}. \tag{2.22d}$$

It is worth to note that the complete nonlinear aircraft model must include all described blocks, and not only the 6-DoF model. For example, besides the control inputs $\boldsymbol{u}_i(t) = [\delta_{d,T,i}(t)\ \delta_{d,a,i}(t)\ \delta_{d,e,i}(t)\ \delta_{d,r,i}(t)]^T$ does not appearing explicit in (2.21), they affect the model by controlling the forces and moments values.

The geometric, inertial and aerodynamics derivatives from YF-22 obtained by [11] are presented in Tables 2.1 and 2.3.

Table 2.3: YF-22 aircraft model inertial parameters and aerodynamic derivatives

Inertial parameters (Kg $\cdot$ m^2)			
$J_x = 1.61$	$J_y = 7.51$	$J_z = 7.18$	$J_{xz} = -0.24$
$J_{xy} = 0$	$J_{yz} = 0$		
Longitudinal aerodynamic derivatives (adimentional)			
C_{D0} = 0.0085	$C_{D\alpha}$ = 0.5079	C_{Dq} = 0	$C_{D\delta E}$ = -0.0339
C_{L0} = -0.0492	$C_{L\alpha}$ = 3.2580	C_{Lq} = -0.0006	$C_{L\delta E}$ = 0.1898
C_{m0} = 0.0226	$C_{m\alpha}$ = -0.4739	C_{mq} = -3.4490	$C_{m\delta E}$ = -0.3644
Lateral-directional aerodynamic derivatives (adimentional)			
C_{Y0} = 0.0156	$C_{Y\beta}$ = 0.2725	C_{Yp} = 1.2151	C_{Yr} = -1.1618
$C_{Y\delta A}$ = 0.1836	$C_{Y\delta R}$ = -0.4592	$C_{l\delta A}$ = -0.0559	$C_{l\delta R}$ = 0.0141
C_{l0} = -0.0011	$C_{l\beta}$ = -0.0380	C_{lp} = -0.2134	C_{lr} = 0.1147
C_{n0} = -0.0006	$C_{n\beta}$ = 0.0361	C_{np} = -0.1513	C_{nr} = -0.1958
$C_{n\delta A}$ = -0.0358	$C_{n\delta R}$ = -0.0555		

2.4 AIRCRAFT INNER LOOP CONTROLLER

As discussed in Chapter 1 and shown in Fig. 1.2, the control system of an UAV is composed usually of three nested loops. Here it is discussed the inner loop controller.

In cruise flight, in which the aircraft is in straight level flight at constant altitude and speed, the 6-DoF nonlinear model can be linearized and after decoupled to two state space linear models [50]: the longitudinal and the lateral-directional one

$$\begin{bmatrix}\dot{V}_i(t)\\ \dot{\alpha}_i(t)\\ \dot{Q}_i(t)\\ \dot{\theta}_i(t)\end{bmatrix} = \boldsymbol{A}_{\text{lon}}\begin{bmatrix}V_i(t)\\ \alpha_i(t)\\ Q_i(t)\\ \theta_i(t)\end{bmatrix} + \boldsymbol{B}_{\text{lon}}\begin{bmatrix}\delta_{e,i}(t)\\ \delta_{T,i}(t)\end{bmatrix}, \qquad \begin{bmatrix}\dot{\beta}_i(t)\\ \dot{P}_i(t)\\ \dot{R}_i(t)\\ \dot{\phi}_i(t)\end{bmatrix} = \boldsymbol{A}_{\text{lat}}\begin{bmatrix}\beta_i(t)\\ P_i(t)\\ R_i(t)\\ \phi_i(t)\end{bmatrix} + \boldsymbol{B}_{\text{lat}}\begin{bmatrix}\delta_{a,i}(t)\\ \delta_{r,i}(t)\end{bmatrix}. \quad (2.23)$$

The linearized and decoupled model is aimed to evaluate the stability and performance of the aircraft, being useful, for example, in the design of linear controllers. It is worth to note that the model does not include the coordinates $x_i(t)$, $y_i(t)$, $z_i(t)$ and the yaw $\psi_i(t)$ direction of the aircraft, because they does not affect the dynamic modes of the aircraft. It is also worth of note that the linearized model and the linear controller developed using the linearized model are adequate to mildly maneuvers, but not to acrobatic ones [34].

The linearized model and the inner loop controller are not the focus of this thesis. A detailed description of the linearization process, the meaning of each element of the $\boldsymbol{A}_{\text{lon}}$, $\boldsymbol{B}_{\text{lon}}$, $\boldsymbol{A}_{\text{lat}}$, and $\boldsymbol{B}_{\text{lat}}$ matrices, the dynamic modes and how to design a controller based on the characteristics of the dynamic modes are given in [50]. Brief explanation about the linearization can be found in, e.g, [11], whereas a description of a practical implementation of the PID control laws described below can be found in [34]. The presented linearized model is used to give context to the presented inner loop controller, and the inner loop controller is developed to be an interface between the outer loop, which is the focus of this thesis, and the aircraft model.

A common inner loop design, depicted in Fig. 2.6, is to use single input single output (SISO) linear controllers. Here, the inner loops are based on [34]. The loops contains PID controllers. All controllers saturates if it achieves the maximum/minimum limit of the actuator output. When saturated, the integral channel input is opened, as an anti wind-up strategy. The PID controller starts with a trimmed command to straight level flight, obtained as explained in Appendix A.2. The saturation, trimming, and anti wind-up strategies are omitted from equations below for simplicity. When possible, the derivative of the signals are calculated analytically, as shown in [50, 34]. It is assumed that there are sensors and/estimators providing the information needed by the controller. This controller is further discussed in subsections 2.4.1 and 2.4.2

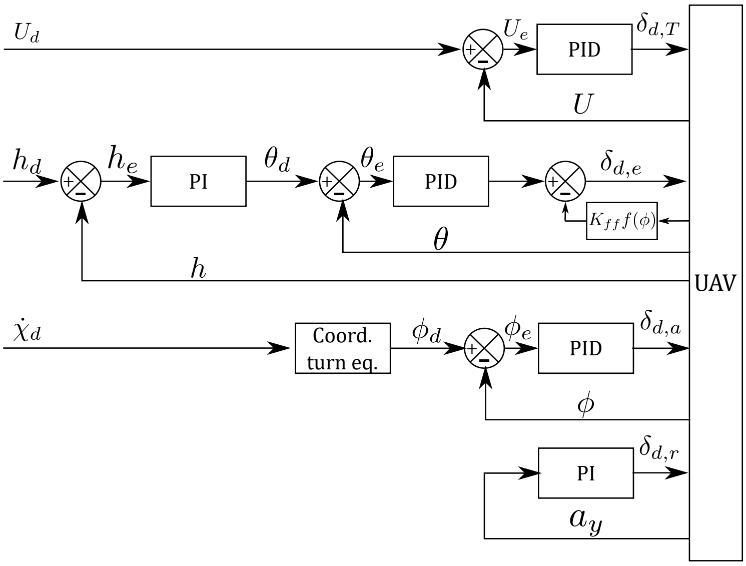


Figure 2.6: Inner loop control structure

Another inner loop variant receives an acceleration $\dot{V}_{d,i}(t)$ and the horizontal and vertical rate-of-turns $\dot{\chi}_{d,i}(t)$ and $\dot{\gamma}_{d,i}(t)$. This approach is interesting, because it can be easily adapted to control a tangent and two perpendicular accelerations relatively to the trajectory, which can be grouped as a single acceleration vector. This approach is presented in subsection 2.4.3

2.4.1 Longitudinal controllers

The longitudinal controllers controls the altitude and the airspeed of the aircraft. Both actuators, $\delta_{e,i}(t)$ and $\delta_{T,i}(t)$ affect both the altitude and airspeed. A simple approach is to use one actuator to control the altitude and other to control the speed. Here is chosen the elevator for altitude and the propulsion for speed. The effect of an actuator in the other state is treated as disturbance. The longitudinal and lateral controllers are explained assuming that the angular velocities are provided by a gyro sensor, the acceleration by an accelerometer and the attitude angles by an estimator. However, the simulation of sensors and estimation is outside the scope of this thesis. In the simulations presented in this work, the measurements are assumed ideal.

Subtracting the estimated altitude $h_i(t)$ from the desired one $h_{d,i}(t)$ provided by the outer loop, it is obtained the vertical error $v_{e,i}(t)$, i.e., $h_{e,i}(t) = h_{d,i}(t) - h_i(t)$. The vertical error $h_{e,i}(t)$ is the input of a PI controller, which output is a desired pitch angle $\theta_{d,i}(t)$, given by

$$\theta_{d,i}(t) = -K_{P,h} h_{e,i}(t) - K_{I,h} \int_0^t h_{e,i}(t) dt. \tag{2.24}$$

The difference between $\theta_{d,i}(t)$ and the estimated $\theta_i(t)$, $\theta_{e,i}(t)$, and the estimated derivative of the pitch, $\dot{\theta}(t)$ are then used as an input of a PID controller, which provides the commanded elevator command $\delta_e(t)$. A feedforward command is added to $\delta_{e,i}(t)$ to maintain the vertical component of the lift force constant during rolls.

$$\delta_{e,i}(t) = -K_{P,\theta}\theta_{e,i}(t) - K_{I,\theta} \int_0^t \theta_{e,i}(t) dt + K_{D,\theta}\dot{\theta}_i(t) - K_{\text{ff}} \left(\frac{1}{\cos(\phi_i(t))} - 1 \right). \tag{2.25}$$

To avoid stall, the magnitude of the calculated $\theta_{d,i}(t)$ is limited to ±20° in the simulation.

Using the measured angular velocities $Q_i(t)$ and $R_i(t)$ and the estimated roll angle $\phi_i(t)$, the derivative of $\theta_i(t)$ is calculated as

$$\dot{\theta}_i(t) = Q_i(t) \cos \phi_i(t) - R_i(t) \sin \phi_i(t). \tag{2.26}$$

The minus sign in (2.24) indicates that a negative change in $\theta_i(t)$ provides a positive change in $h_{e,i}(t)$ and the minus sign in (2.25) means that a negative change in angle in elevator is needed to provide a positive change in $\theta_i(t)$. It is worth note that a positive $h_{e,i}(t)$ means aircraft losing altitude relative to the desired altitude, since the z vector from S_{NED} points down.

The throttle controller uses $U_{e,i}(t)$, the difference between the desired airspeed $U_{d,i}(t)$ and the Pitot-tube measured one $U_i(t)$, as input of a PID controller to obtain the throttle command $\delta_{T,i}(t)$, given by

$$\delta_{T,i}(t) = -K_{P,U} U_{e,i}(t) - K_{I,U} \int_0^t U_{e,i}(t) dt + K_{D,U} \dot{U}_{e,i}(t). \tag{2.27}$$

The derivative of airspeed, $\dot{U}_{e,i}(t)$, is calculated numerically. Alternatively, it can be assumed that a Kalman filter estimator is providing the ground speed estimate and its derivative in a sufficiently high rate.

The control coefficients are empirically adjusted as shown in Section 2.5.

2.4.2 Lateral controller

The lateral controller controls the rate-of-turn $\dot{\chi}_i(t)$ and, in some cases, the sideslip angle $\beta_i(t)$. Again, both actuators affect both outputs. A common approach is to use both controllers to control only the rate-of-turn [11, 4]. The approach used here is the same from [34], which uses the aileron to control the rate-of-turn and the rudder to regulate the sideslip to zero.

The rate-of-turn control is projected under the assumption of a coordinated turn [34], which is shown in Fig. 2.7. In coordinated turn, the thrust and drag forces are perpendicular to the plane shown in Fig.

2.7, which is true for a symmetrical aircraft with null sideslip angle. Being $\beta_i(t)$ null, the velocity vector is also perpendicular to the plane in a no wind scenario. So, if there is a difference between drag and thrust, it will increase or decrease the speed of the aircraft, but not change its course.

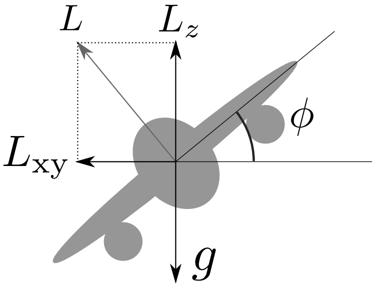


Figure 2.7: Coordinated turn. A level turn in which the vertical component of the lift perfectly counterbalances the the gravity force.

The only forces present in the represented plane are the lift $L_i(t)$ and the gravity g forces. Assuming a constant altitude flight, the vertical component $L_{z,i}(t)$ of the lift force must counterbalance the gravity force, so the resulting force is only the horizontal component $L_{\text{xy},i}(t)$ of the lift vector. Being perpendicular to the velocity vector, the force changes the velocity direction. In other words, the aircraft makes a turn, the $L_{\text{xy},i}(t)$ force creates a centripetal acceleration and the difference between drag and thrust generates a tangent acceleration. Calculating $L_{\text{xy},i}(t)$ from $L_{z,i}(t)$ and the roll angle $\phi_i(t)$, and using the known equations of the uniform circular motion, and assuming a no-wind scenario, it can be obtained:

$$\dot{\chi}_i(t) = \dot{\psi}_i(t) = \frac{g}{V_i(t)} \tan(\phi_i(t)). \tag{2.28}$$

Isolating $\phi_i(t)$ from (2.28), it can be calculated a reference value of roll $\phi_{d,i}(t)$ that achieves a desired rate-of-turn $\dot{\chi}_{d,i}(t)$ [11, 4, 34]

$$\phi_{d,i} = \arctan\left(\frac{\dot{\chi}_{d,i}(t)V_i(t)}{g}\right). \tag{2.29}$$

A more general turn equation which includes an arbitrary lift force and a non-null flight path angle is given by [55]

$$\dot{\chi}_i(t) = \frac{L_i(t)\sin\phi_i(t)}{m_i V_i(t)\cos\gamma_i(t)} \tag{2.30}$$

By defining the load factor $n_i(t)$ as [49]

$$n_i(t) \triangleq \frac{L_i(t)}{W_i(t)} = \frac{L_i(t)}{m_i g}, \tag{2.31}$$

Eq. (2.30) is rewritten as

$$\dot{\chi}_i(t) = \frac{g n_i(t)\sin\phi_i(t)}{V_i(t)\cos\gamma_i(t)}, \tag{2.32}$$

In this case, $\phi_{d,i}(t)$ is given by

$$\phi_{d,i}(t) = \arcsin\left(\frac{\dot{\chi}_{d,i}(t)V_i(t)\cos\gamma_i(t)}{gn_i(t)}\right). \tag{2.33}$$

The roll error $\phi_{e,i}(t)$, which is the estimated roll $\phi_i(t)$ subtracted by the desired roll $\phi_{d,i}(t)$, and the estimated derivative of the roll $\dot{\phi}_i(t)$ are used as a input of a PID controller, providing an aileron command $\delta_{a,i}(t)$

$$\delta_{a,i}(t) = -K_{P,\phi}\phi_{e,i}(t) - K_{I,\phi}\int_0^t \phi_{e,i}(t)dt + K_{D,\theta}\dot{\phi}_i(t), \tag{2.34}$$

where

$$\dot{\phi}_i = P(t) + [Q(t)\sin\phi_i(t) + R(t)\cos\phi_i(t)]\tan\theta_i(t). \tag{2.35}$$

The coordinated turn assumes that there is only the lift and gravity forces. Since the accelerometer does not measure the gravity acceleration, and the lift force, under the hypothesis of no-wind scenario and null $\beta_i(t)$, is aligned with the z_b axis of the S_{bi} from the aircraft, the accelerometer measured acceleration $a_{by,i}(t)$ in the y_b axis must be null. If it is not null, $\beta_i(t) \neq 0$. By applying a rudder command $\delta_{r,i}(t)$ to remove all accelerometer-measured acceleration $a_{by,i}(t)$ in the y_b axis, $\beta_i(t)$ can be regulated to zero. This is made by the use of the following PI controller

$$\delta_{r,i}(t) = K_{P,a_{by}}a_{by,i}(t) + K_{I,a_{by}}\int_0^t a_{by,i}(t)dt. \tag{2.36}$$

2.4.3 Alternative inner loop controller

In this inner loop variant, the controller receives as reference signal the values $\dot{V}_{d,i}(t)$, $\dot{\chi}_{d,i}(t)$, and $\dot{\gamma}_{d,i}(t)$, which can be easily changed to a three axial acceleration. Alternatively, the reference is $\dot{V}_{d,i}(t)$, $L_{d,i}(t)$ and $\phi_{d,i}(t)$. This inner loop usually also regulates $\beta_i(t)$ to zero.

The variant can be seen fully implemented in, e.g, [56, 35]. Here, a simplified version, adequate to horizontal formation flight, is presented.

Since the tracking of $\dot{\chi}_{d,i}(t)$ and the regulation of $\beta_i(t)$ is the same from lateral controller from previous inner loop controller, it uses the same equations (2.29), (2.34) and (2.36).

From the aircraft model (Section 2.3), the forward acceleration is given by [11, 50]

$$\dot{V}_i(t) = \omega_{1,i}(t)T_i(t) + \omega_{2,i}(t), \tag{2.37a}$$

$$\omega_{1,i}(t) \triangleq \frac{\cos\alpha_i(t)\cos\beta_i(t)}{m_i}, \tag{2.37b}$$

$$\omega_{2,i}(t) \triangleq -\frac{D_i(t)\cos\beta_i(t) - Y_i(t)\sin\beta_i(t)}{m_i} - g\sin\gamma_i(t), \tag{2.37c}$$

where $\omega_{1,i}$ and $\omega_{2,i}$ are defined to write (2.37) in a compact fashion. These equations assume a propulsive system that is aligned to the x_b axis of the aircraft.

In this way, to achieve a desired acceleration, the desired thrust is given by

$$T_{d,i}(t) = \frac{\dot{V}_{d,i}(t) - \omega_{2,i}(t)}{\omega_{1,i}(t)}. \tag{2.38}$$

It is assumed that the regulator maintains $\beta_i(t) = 0$, which can be used to simplify the obtained equation, as in [11]. By also assuming that the aircraft is in level cruise flight, and that the α angle is projected to be null in this scenario, is also reasonable to assume that $\alpha_i(t)$ is sufficiently near zero, applying extra simplifications, as in [55, 14, 15, 26]. In this case,

$$\dot{V}_i(t) = \frac{T_i(t) - D_i(t)}{m_i} - g\sin\gamma_i(t), \tag{2.39}$$

$$T_{d,i}(t) = m_i\left(\dot{V}_{d,i}(t) + g\sin\gamma_i(t)\right) + D_i(t). \tag{2.40}$$

The computation of $\delta_{T,i}(t)$ from $T_{d,i}(t)$ can be made, for example, by knowing the engine model. From (2.16), which is a model of a specific engine, $\delta_{T,i}(t)$ is computed as

$$\delta_{T,i}(t) = \frac{T_{d,i}(t) - T_b}{K_T} \tag{2.41}$$

The altitude is assumed to be maintained constant. In this case, $\gamma_{d,i}(t) = 0$. The altitude is maintained constant by using (2.25). If there is a control loop controlling the lift value, and the aircraft is making 3D trajectories, $\dot{\gamma}_i(t)$ is given by [14]

$$\dot{\gamma}_i(t) = \frac{L_i(t)\cos\phi_i(t)}{m_i V_i(t)} - \frac{g\cos\gamma_i(t)}{V_i(t)} = g\frac{n_i(t)\cos\phi_i(t) - \cos\gamma_i(t)}{V_i(t)} \tag{2.42}$$

2.4.4 The point-mass model

The alternative inner loop controller from subsection 2.4.3 can be used to encapsulate the aircraft model into a simpler model called point-mass model [55, 14, 15, 26] or the similar 3D Dubins vehicle [33, 6, 57, 58]. The 3D Dubins model is given as follows

$$\begin{bmatrix} \dot{\boldsymbol{p}}_{\mathrm{NED},i}(t) \\ \dot{V}_i(t) \\ \dot{\chi}_i(t) \\ \dot{\gamma}_i(t) \end{bmatrix} = \begin{bmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \\ \dot{z}_i(t) \\ \dot{V}_i(t) \\ \dot{\chi}_i(t) \\ \dot{\gamma}_i(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}^{sw}_{\mathrm{NED},i}(t) \begin{bmatrix} V_i(t) \\ 0 \\ 0 \end{bmatrix} \\ u_{t,i}(t)a_{\max} \\ u_{y,i}(t)\dot{\chi}_{\max} \\ u_{p,i}(t)\dot{\gamma}_{\max} \end{bmatrix} = \begin{bmatrix} V_i(t)\cos\gamma_i(t)\cos\chi_i(t) \\ V_i(t)\cos\gamma_i(t)\sin\chi_i(t) \\ -V_i(t)\sin\gamma_i(t) \\ u_{t,i}(t)a_{\max} \\ u_{y,i}(t)\dot{\chi}_{\max} \\ u_{p,i}(t)\dot{\gamma}_{\max} \end{bmatrix}, \tag{2.43}$$

where $u_{t,i}(t)$, $u_{y,i}(t)$ and $u_{p,i}(t)$ are control inputs, $|u_{t,i}(t)| \leq 1$, $|u_{y,i}(t)| \leq 1$ and $|u_{p,i}(t)| \leq 1$, and $a_{\max}$, $\dot{\chi}_{\max}$, and $\dot{\gamma}_{\max}$ are the maximum allowed forward acceleration, rate-of-turn and derivative in flight path angle. The subscripts t, p, and y means respectively thrust, pitch, and yaw. The model is obtained by assuming that the inner loop can achieve the desired $\dot{V}_d(t)$, $\dot{\chi}_d(t)$, and $\dot{\gamma}_d(t)$ fast enough.

The 3D Dubins model explicitly shows the aircraft has geometric limitations: by using the centripetal acceleration equations:

$$\dot{\chi}(t) = \frac{V_i(t)}{r_h}, \qquad \dot{\gamma}(t) = \frac{V_i(t)}{r_v}, \tag{2.44}$$

it can be seen that the maximum allowed rate-of-turn and maximum allowed derivative in flight path angle implies, to a given speed, that there is a minimum allowed horizontal and vertical turn radius r_h and r_v. It is worth to note that smaller radius means sharper (more intense) turns. The 3D Dubins model, in this way, is good to be used to define reasonable trajectories to aircrafts, and is used mainly in the path planner algorithm.

By using (2.12), the inertial acceleration described in the simplified wind frame is

$$\boldsymbol{a}_{swi}(t) = \begin{bmatrix} a_{swxi}(t) \\ a_{swyi}(t) \\ a_{swzi}(t) \end{bmatrix} = \dot{\boldsymbol{V}}_{swi}(t) + \omega_{swi}^{\mathrm{NED},swi}(t) \times \boldsymbol{V}_{swi}(t) = \begin{bmatrix} \dot{V}_i(t) \\ V_i(t)\dot{\chi}_i(t)\cos\gamma_i(t) \\ V_i(t)\dot{\gamma}_i(t) \end{bmatrix}. \tag{2.45}$$

In robust formation flight controllers, the literature [15, 14] separates the acceleration in three sources: $\boldsymbol{\Gamma}_i(t)$, generated by the UAV, $\boldsymbol{g}$, generated by the gravitational acceleration, and $\boldsymbol{b}_i(t)$, generated by disturbances and model imprecision

$$\boldsymbol{a}_{swi}(t) = \boldsymbol{\Gamma}_i(t) + \boldsymbol{D}_{swi}^{\mathrm{NED}}(t)\boldsymbol{g}_{\mathrm{NED}} + \boldsymbol{b}_i(t) = \begin{bmatrix} a_{ti}(t) - g\sin\gamma_i(t) + b_{ti}(t) \\ a_{yi}(t) + b_{yi}(t) \\ a_{pi}(t) - g\cos\gamma_i(t) + b_{pi}(t) \end{bmatrix}, \tag{2.46}$$

where

$$\boldsymbol{\Gamma}_i(t) = \begin{bmatrix} a_{ti}(t) \\ a_{yi}(t) \\ a_{pi}(t) \end{bmatrix} = \begin{bmatrix} \frac{T_i(t)-D_i(t)}{m_i} \\ gn_i(t)\sin\phi_i(t) \\ gn_i(t)\cos\phi_i(t) \end{bmatrix}, \tag{2.47}$$

is obtained from (2.39), (2.32), and (2.42) and the disturbance signal $\boldsymbol{b}_i(t) = \begin{bmatrix} b_{ti}(t) & b_{pi}(t) & b_{yi}(t) \end{bmatrix}$ encompass model approximations, parameter uncertainty, and disturbances in acceleration, generated by several sources, such as wind. It is worth to note that $a_{ti}(t)$ can be calculated with extra precision by using (2.37) instead of (2.39).

By reordering (2.45) and using (2.46), (2.43) is rewritten as the point mass model

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \\ \dot{z}_i(t) \\ \dot{V}_i(t) \\ \dot{\chi}_i(t) \\ \dot{\gamma}_i(t) \end{bmatrix} = \begin{bmatrix} V_i(t)\cos\gamma_i(t)\cos\chi_i(t) \\ V_i(t)\cos\gamma_i(t)\sin\chi_i(t) \\ -V_i(t)\sin\gamma_i(t) \\ a_{ti}(t) - g\sin\gamma_i(t) + b_{ti}(t) \\ \frac{a_{yi}(t)+b_{yi}(t)}{V_i(t)\cos\gamma_i(t)} \\ \frac{a_{pi}(t)-g\cos\gamma_i(t)+b_{pi}(t)}{V_i(t)} \end{bmatrix}. \tag{2.48}$$

In this model, the input can be seen as a 3D vector acceleration. This approach can be used to implement outer loop controllers such as formation flight controllers. Figure 2.8 depicts the point-mass model.

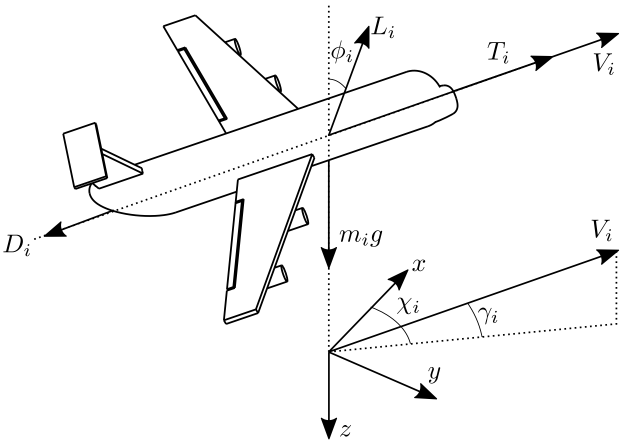


Figure 2.8: The i-th UAV, its force and velocity vectors, and attitude angles.

2.5 TUNNING THE INNER LOOP

Here the inner loop controllers are tuned by trial and error, by applying a step input in each controller. After, the nonlinear system is simulated three times, each time evaluating a different actuator: aileron, elevator, engine. When possible, high amplitude inputs are utilized, to include most of the flight envelope.

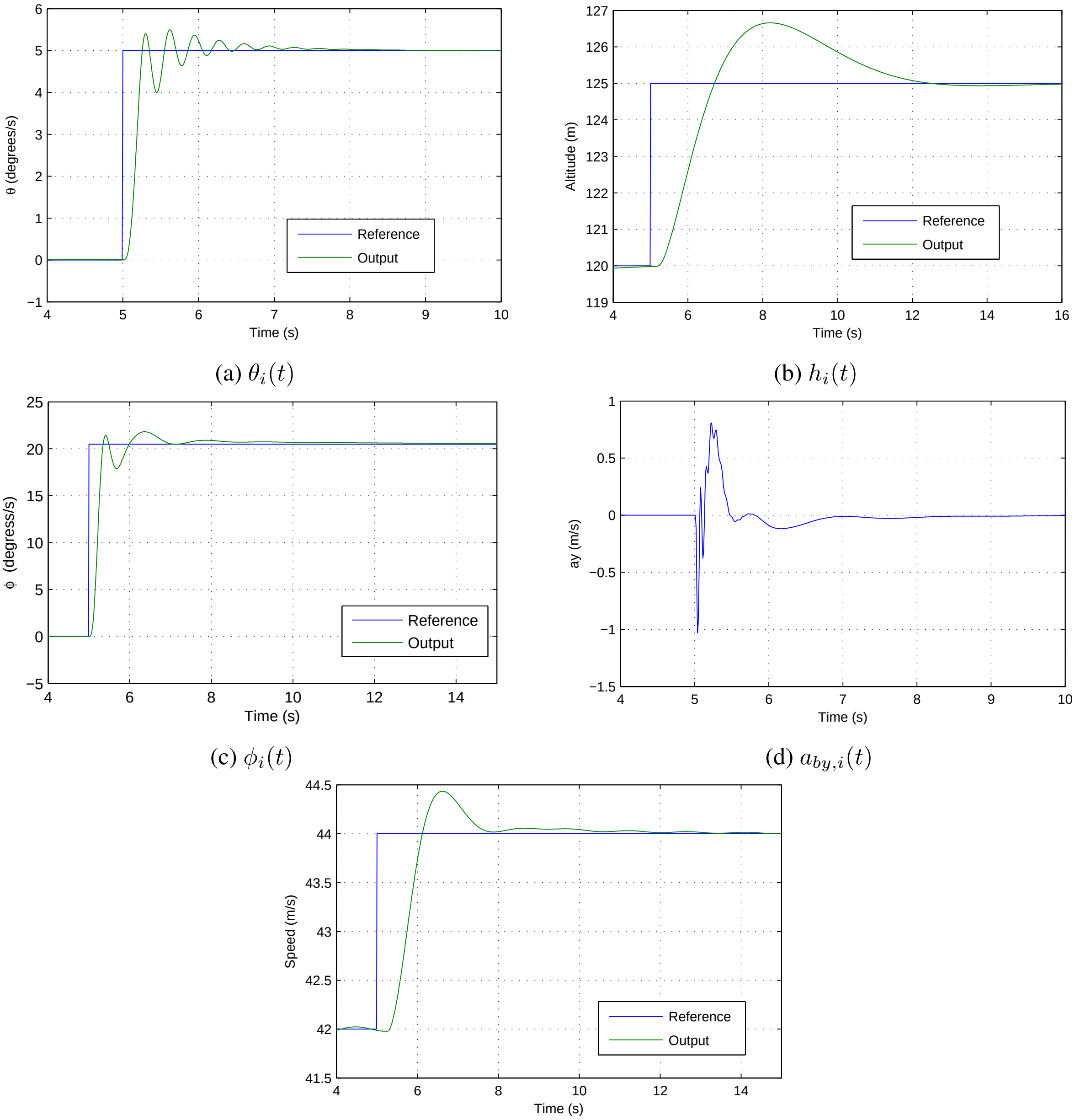
Since the rudder only regulates $a_{yb,i}$ to zero, it does not make sense to simulate a step input in it. Its regulation is evaluated during the aileron simulation.

The coefficients are defined as shown in Table 2.4. Figure 2.9 and Table 2.4 shows the results obtained by using the defined control tuning. The performance parameters are the rise time t_r, the settling time t_s and the overshoot M_p.

Table 2.4: Obtained results in nonlinear model

Transfer function	Actuator	K_P	K_I	K_D	K_{ff} *	t_r [s]	t_s [s]	M_p [%]
$\theta_i(t)/\theta_{d,i}(t)$	$\delta_e(t)$	4	2	0.24	0.0994	0.14	1.97	10.0
$h_i(t)/h_{d,i}(t)$	$\theta_{d,i}(t)$	0.02	0.01	-	-	1.08	6.89	33.2
$\phi_i(t)/\phi_{d,i}(t)$	$\delta_{a,i}(t)$	1.1464	0.22546	0.11391	-	0.20	2.97	6.5
$a_{yb,i}/a_{ybd,i}(t)$	$\delta_{r,i}(t)$	0.1	0.1	-	-	-	≈ 2	-
$V_i(t)/V_{d,i}(t)$	$\delta_{T,i}(t)$	70	20	10	-	0.59	5.03	21.7

* See (2.25).



(a) $\theta_i(t)$

(b) $h_i(t)$

(c) $\phi_i(t)$

(d) $a_{by,i}(t)$

(e) $V_i(t)$

Figure 2.9: Closed loop response. Except by the rudder regulating $a_{by,i}(t)$, plots shows the step response.

2.6 CONCLUSION

In this Chapter, a single aircraft simulator is developed, including the 6-DoF nonlinear model and a set of PID controllers that controls the UAV to achieve a desired altitude, desired rate-of-turn and a desired speed.

The PID inner loop controllers are one of the simplest solutions found in the literature. These loops can be changed to other linear or nonlinear approaches, but this is not evaluated here, since the scope of this thesis is the development of autonomous formation flight algorithms, which is made mainly by the outer loop, as discussed in Section 1.2.

The described model and the developed simulator are used in the design and simulation of formation flight controllers developed in Chapters 3 and 4.

3 AUTONOMOUS FORMATION FLIGHT NONLINEAR CONTROLLER

3.1 INTRODUCTION

The inner loop controller makes the UAV to behave as the point-mass model described in Subsection 2.4.4. The outer loop controllers, which includes most of the formation flight controllers, usually assumes the inner loop controller aircraft behaves exactly as described by this point-mass model, as discussed in Subsection 1.2.2.

As discussed in Chapter 1, this work choses the leader-follower and the very similar virtual-leader approaches. In these approaches, the formation is described in a single frame, which is known by all UAVs. The frame can be, for example, the inertial frame (S_{NED}) [38, 14, 15, 26], or a leader's defined frame, such as $S_{\chi 0}$ [39, 11, 12, 29] or S_{sw0} [13]. The dynamics of formation control, which are derived from the point-mass model, depends on the chosen shared frame but, in any case, is nonlinear.

The NLDI control is developed here. As discussed in the Chapter 1, the NLDI controller transforms the nonlinear model in a linear one. Choosing distinct frames to make the formation flight model results in distinct controllers.

First, the inertial frame S_{NED} is chosen. This results in a position-based formation control [21], which needs an absolute position sensor, such as a GPS receiver and less aircraft communication. It is the most usual approach, and is used in several recent articles, such as [15, 14]. Also, it is interesting because it can be used in more general situations, such as single aircraft following a pre-planned trajectory. Other benefit is that the obtained model can use generic multi-agent control techniques without significant modification. The NLDI controller obtained by choosing the S_{NED} frame is named here as S_{NED}-NLDI variant. It is a controller that already exists in the literature, and it is only reviewed here. Proposed modifications to this controller are presented in Chapter 4.

After, the leader's frame $S_{\chi 0}$ is chosen, and the obtained controller is named here as $S_{\chi 0}$-NLDI approach. It can be considered a variant of the displacement-based formation control [21] in which the leader broadcast its attitude to the followers in a way that all aircraft know the $S_{\chi 0}$ frame. Being a displacement-based formation, the followers must known only the relative position between itself and the leader, which allows to replace the GPS receiver to a sensor that measures relative positions, such as a camera plus computer vision algorithms. Other benefit is that, as shown in this chapter, the control gains can be used to tune separately the forward movement, which is engine generated and the lateral movement, which is aileron plus rudder generated. The $S_{\chi 0}$-NLDI approach is first reviewed here with an example from literature [11]. After, an improved version of this controller is proposed, providing a better performance under a broader set of scenarios, including variable shape formation flight and a maneuvering leader. Also as contribution, the stability of the controller is presented. It is worth to note that there are others leader's frames that can be chosen, such as S_{sw0} [13], which generates another NLDI variant, but they are not evaluated here.

The proposed $S_{\chi 0}$-NLDI approach simplifies the project of path planners. UAV path planners, as discussed in Subsection 1.2.3, usually models the aircraft as an unicycle, or Dubins, vehicle [46]. This means that the aircraft is constantly going forward, and can change the course by applying a limited rate-of-turn or, similarly, a limited centripetal acceleration. The aircraft can not move sideways, or backwards. It is shown that the linearized formation model behaves like a double-integrator. Relatively to the leader, the follower aircraft can fly forward, backwards, or sideways, moving like an omnidirectional robot. A proof-of-concept path planner is shown, in which a simple formation achieving maneuver is developed and collision avoidance is obtained by waiting the other cooperative aircraft moving first.

The main results presented here, which are the improved $S_{\chi 0}$-NLDI approach and its path planner, are published in [3]. A second $S_{\chi 0}$ based controller was also developed during this PhD research. It is a improvement in the pursuit-based controller presented in [12], which uses some of the equations present in the NLDI controler from [11]. This controller is published in [59]. The controller, however, performed worse than the NLDI one in any situation, and is not further discussed here. The S_{NED}-NLDI approach is only introduced in this chapter, whereas improvements and contributions are presented in Chapter 4.

In Section 3.2, the feedback linearization controller is briefly reviewed. In Section 3.3 a brief review in formation flight models are presented. In Section 3.4, the known S_{NED}-NLDI approach is presented. In Section 3.5, as a contribution of this work, a feedback linearization controller, its stability proof, and an path planner are presented. Section 3.7 presents a brief comparison between the S_{NED} and $S_{\chi 0}$ approaches. Simulations are presented in Section 3.8 and the conclusions are presented in Section 3.9.

3.2 FEEDBACK LINEARIZATION

The explanations, examples, and formulas from this Section are based in [42].

Feedback linearization can be defined as a technique in which the control input is chosen in a way that cancels the nonlinearities of a nonlinear system, achieving a linear closed loop description. A more formal explanation, from [42], is that this technique aims to achieve an algebraic transformation that, when applied in the nonlinear state space model, results into a full or partially linear state model, in which the usual linear control techniques can be applied.

The feedback linearization technique can be easily applied in nonlinear systems that can be represented in the companion form

$$\xi^{(n)}(t) = f(\boldsymbol{\xi}(t)) + b(\boldsymbol{\xi}(t))u(t), \tag{3.1}$$

where $\xi^{(n)}(t)$ is the n-th derivative of the state $\xi(t)$, $\boldsymbol{\xi}(t)$ = $[\xi(t)\ \dot{\xi}(t)\ ...\ \xi^{(n-1)}(t)]^T$ is the state vector and f and b are any nonlinear functions, $b(\boldsymbol{\xi}(t)) \neq 0$ in the region of interest in the state space.

If the previous conditions are achieved, $u(t)$ can be chosen to cancel b and f

$$u(t) = [v(t) - f(\boldsymbol{\xi}(t))]/b(\boldsymbol{\xi}(t)), \tag{3.2}$$

where $v(t)$ is a new input to be defined. Inserting (3.2) in (3.1)

$$\xi^{(n)}(t) = v(t). \tag{3.3}$$

The resulting equation is a n-th integrator. Choosing

$$v(t) = -k_{n-1}\xi^{(n-1)}(t) - ... - k_1\dot{\xi}(t) - k_0\xi(t), \tag{3.4}$$

Eq. (3.3) becomes

$$\xi^{(n)}(t) + k_{n-1}\xi^{(n-1)}(t) + ... + k_1\dot{\xi}(t) + k_0\xi(t) = 0, \tag{3.5}$$

which is a linear system, which is exponentially stable and converges to $\xi(t) = 0$ if all poles are strictly in the left half complex plane. It is worth to note that the nonlinear system, being fully linearized, is stable if (3.5) is stable. There is no need to prove the stability by, e.g, Lyapunov candidate functions.

Defining the input u to cancel the nonlinearities and obtain a linear state space is called input-state linearization.

If it is desired to track an output $x_d(t)$, $v(t)$ can be redefined as

$$v(t) = \xi_d^{(n)} - k_{n-1}e^{(n-1)}(t) - ... - k_1\dot{e}(t) - k_0e(t), \tag{3.6}$$

where $e(t) = \xi(t) - \xi_d(t)$. In this case the tracking error exponentially converges to zero, and $\xi^{(n)}(t) = \xi_d^{(n)}(t)$.

3.2.1 Input-output linearization

It is not always easy to find a relationship between a state-linearizing input $u(t)$ and some output $y(t)$, especially if the relationship between the states and the output is nonlinear. A solution is to find an input-output linearization. In this case, a relationship between the control input $u(t)$ and the output $y(t)$ is found.

The relationship is found as follows. It starts with $y(t) = h(\boldsymbol{x}(t))$. The output is derived r times, until the input appears in $y^{(r)}(t)$ equation:

$$y^{(r)}(t) = f(\boldsymbol{\xi}) + b(\boldsymbol{\xi})u(t), \tag{3.7}$$

where f and b are functions of the state vector $\boldsymbol{\xi}(t)$, and are distinct from the f and b functions presented in (3.1). The deductions now are the same from the input-state linearization. By choosing $u(t)$ as (3.2), it is found an r-th integrator:

$$y^{(r)}(t) = v(t). \tag{3.8}$$

To achieve tracking capability, v is defined as

$$v(t) = y_d^{(r)} - k_{r-1}e^{(r-1)}(t) - ... - k_1\dot{e}(t) - k_0e(t), \tag{3.9}$$

where $e(t) = y(t) - y_d(t)$.

The variable r is called the *relative degree* of the system and, in linear systems, is the difference between the number of poles and zeros. If the relative degree is equal to the number of states n, the output $y(t)$ and all $n-1$ derivatives can be used as states, achieving a fully linearized state vector. In this case, the input-output linearization is a specific case of the state-output linearization, and is stable if the linearized state is stable.

3.2.2 Multi-input multi-output systems

The presented SISO discussion can be easily extended to MIMO cases in the case of square systems, i.e., systems that has the same number of inputs and outputs.

Each $y_i(t)$ output is derived r_i times until at least one of the inputs appears in $y^{(r_i)}(t)$, where r_i is the relative degree of each input. As example, lets assume three inputs, three outputs, and that r_1, r_2, and r_3 are equal to two. In this case, the relationship between the inputs and outputs are

$$\begin{bmatrix} \ddot{y}_1(t) \\ \ddot{y}_2(t) \\ \ddot{y}_3(t) \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{\xi}(t)) \\ f_2(\boldsymbol{\xi}(t)) \\ f_3(\boldsymbol{\xi}(t)) \end{bmatrix} + \boldsymbol{B}(\boldsymbol{\xi}(t)) \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_2(t) \end{bmatrix}, \tag{3.10}$$

where $f_i(t)$ are scalar functions and $\boldsymbol{B}(\boldsymbol{\xi}(t))$ is a 3×3 matrix. Similarly to the SISO case, by defining

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} = \boldsymbol{B}(\boldsymbol{\xi}(t))^{-1} \begin{bmatrix} v_1(t) - f_1(\boldsymbol{\xi}(t)) \\ v_2(t) - f_2(\boldsymbol{\xi}(t)) \\ v_3(t) - f_3(\boldsymbol{\xi}(t)) \end{bmatrix}, \tag{3.11}$$

it is obtained the three uncoupled equations

$$\ddot{y}_1(t) = v_1(t), \qquad \ddot{y}_2(t) = v_2(t), \qquad \ddot{y}_3(t) = v_3(t), \tag{3.12}$$

where $v_i(t)$ can be defined, independently, to achieve an exponentially stable tracking capability.

$$v_i(t) = \ddot{y}_{d,i}(t) - k_1 \dot{e}_i(t) - k_0 e_i(t), \tag{3.13}$$

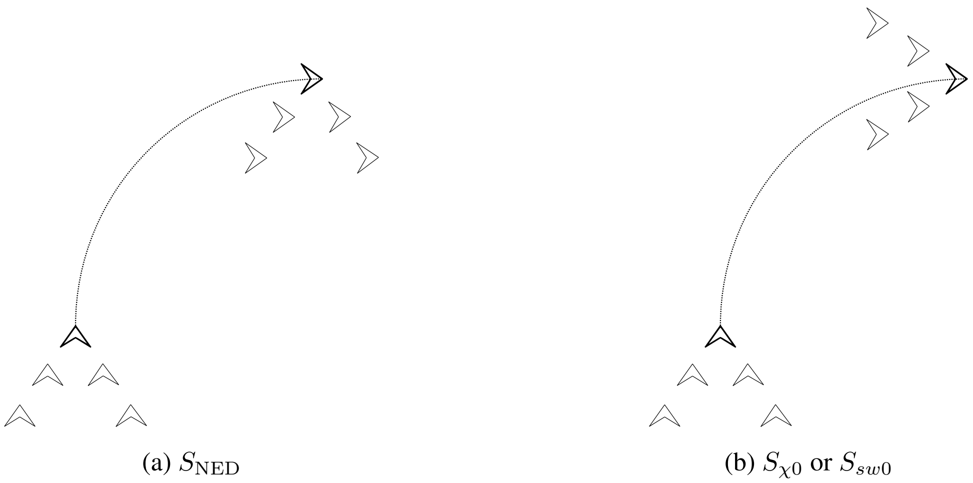
where $e_i(t) = y_i(t) - y_{d,i}(t)$.

The total relative degree r is the sum of r_i. In the example, $r = 2 + 2 + 2 = 6$. Assuming that the model is perfectly known and that $v_i(t)$ are correctly projected, if the total relative degree is equal to n, the nonlinear controller is asymptotically exponentially stable. As is shown in this chapter, the formation flight model is equivalent to this example: has six states, it is a square system with three outputs (3-axis position) commanded by three acceleration inputs. The relative degree between each of the positions and accelerations is two, providing a total relative degree of six. This makes the system fully linear after applying the NLDI.

3.3 FORMATION FLIGHT MODELS

In the proposed formation flight controllers in this thesis, it is assumed that there is a leader aircraft and a set of N followers, where the leader is defined as the 0-th aircraft and each of the followers are defined as the i-th aircraft, $i \in \{1, 2, \ldots, N\}$. The leader aircraft can be a real or a virtual aircraft and its movement is defined independently of the followers. The leader broadcast its information (position, orientation) to all followers and, in this chapter, it is assumed that each follower only receives data from the leader, and does not communicate to each other. Figures 1.1a and 1.1c can be seen as the communication graphs in the leader and virtual leader scenarios.

The formation shape is defined as a set of positions that each aircraft must achieve. For different situations, distinct descriptions of the formation in terms of reference systems are recommended. If the formation shape does not rotate related to the inertial frame as, for example, a V-shape fleet that always points north along the time regardless the flight direction of the fleet (Fig. 3.1a), its motion is well described in the inertial NED reference frame [38, 22]. This style of formation can be useful, for example, if the fleet is acting as a sensor array interacting with non-fleet elements, such as photographing the ground or measuring radio signals from the sky. If, in the example, the V-shape formation rotates horizontally, following the leader's course (Fig. 3.1b), the frame $S_{\chi 0}$ is appropriated [11, 14, 29, 12]. If the V-shape, pointing to the velocity vector of the leader, rotates in any direction, it is more interesting to consider a reference frame that accounts the leader's course and flight path directions [13, 15]. Leader dependent formation shape are useful for intra-fleet interactions, such as aerial refueling.



(a) S_{NED}

(b) $S_{\chi 0}$ or S_{sw0}

Figure 3.1: Above view of a fixed V-shape formation described in distinct frames.

In the $S_{\chi 0}$-NLDI variant proposed in Section 3.5 and in S_{sw0}-NLDI variant from [13], the formation shape can only be described in the shared global frame, i.e, respectively in $S_{\chi 0}$ and S_{sw0}. The S_{NED}-NLDI variant describes the formation shape more naturally in S_{NED} but, as shown in Section 3.4, it can be easily modified to describe the formation shape in any frame.

3.4 NED FRAME NLDI AUTONOMOUS FORMATION FLIGHT APPROACH

Here is presented the S_{NED} described NLDI trajectory tracker from [55], which can be used in different situations, including autonomous formation flight.

By rotating the inertial acceleration $\boldsymbol{a}_{swi}(t)$ from Eq. (2.46) to the inertial frame, it is obtained

$$\ddot{\boldsymbol{p}}_i(t) = \boldsymbol{D}_{\mathrm{NED}}^{swi}(t)\boldsymbol{a}_{swi}(t) = \boldsymbol{D}_{\mathrm{NED}}^{swi}(t)\left(\boldsymbol{\Gamma}_i(t) + \boldsymbol{b}_i(t)\right) + \boldsymbol{g}_{\mathrm{NED}}. \tag{3.14}$$

By defining

$$\boldsymbol{\tau}_i(t) = [\,\tau_{xi}(t) \quad \tau_{yi}(t) \quad \tau_{zi}(t)\,]^T \triangleq \boldsymbol{D}_{\text{NED}}^{swi}(t)\boldsymbol{\Gamma}_i(t) + \boldsymbol{g}, \tag{3.15}$$

$$\boldsymbol{d}_i(t) = [\,d_{xi}(t) \quad d_{yi}(t) \quad d_{zi}(t)\,]^T \triangleq \boldsymbol{D}_{\text{NED}}^{swi}(t)\boldsymbol{b}_i(t), \tag{3.16}$$

the dynamics are finally rewritten as

$$\ddot{\boldsymbol{p}}_i(t) = \boldsymbol{\tau}_i(t) + \boldsymbol{d}_i(t), \tag{3.17}$$

where $\boldsymbol{\tau}_i(t) \in \mathbb{R}^3$ is a virtual controller input and $\boldsymbol{d}_i(t) \in \mathbb{R}^3$ is the virtual disturbance described in the reference frame.

For the controller design in this section, the model given by Eq. (3.17) will be used, but assuming the disturbance null. Once $\boldsymbol{\tau}_i(t)$ is known, the virtual input $\boldsymbol{\Gamma}_i(t)$ can be obtained by using the NLDI

$$\boldsymbol{\Gamma}_i(t) = \boldsymbol{D}_{swi}^{\text{NED}}(t)\left(\boldsymbol{\tau}_i(t) - \boldsymbol{g}_{\text{NED}}\right), \tag{3.18}$$

and then $T_i(t)$, $n_i(t)$, $\phi_i(t)$ can be obtained from (2.47), which can finally be used as the input of the inner loop controller.

Equation (3.18) is the S_{NED} described NLDI of the point-mass model. It can be seen that is in the same format as (3.11), where $\boldsymbol{\Gamma}_i(t)$ is equivalent to $\boldsymbol{u}$, $\boldsymbol{D}_{swi}^{\text{NED}}(t) = \left(\boldsymbol{D}_{\text{NED}}^{swi}(t)\right)^{-1}$ is equivalent to $\boldsymbol{B}^{-1}$ and $\boldsymbol{g}_{\text{NED}}$ to $\boldsymbol{f}(\boldsymbol{x})$.

The relative degree is $2 + 2 + 2 = 6$, which is equal to the number of states of the point mass model, so the model is fully linearized by the NLDI. Also, $\left(\boldsymbol{D}_{\text{NED}}^{swi}(t)\right)^{-1} = \boldsymbol{D}_{swi}^{\text{NED}}(t)$ always can be found, since it is a rotational matrix. In this way, the NLDI is always valid, and the system is stable if the linearized system is stable.

By defining $\boldsymbol{p}_{\text{NED},i}^{d}(t)$ as the desired i-th aircraft position to achieve the formation and

$$\boldsymbol{e}_i(t) \triangleq \boldsymbol{p}_{\text{NED},i}(t) - \boldsymbol{p}_{\text{NED},i}^{d}(t), \tag{3.19}$$

as the error between the measured and desired positions, the control law

$$\boldsymbol{\tau}_i(t) = \ddot{\boldsymbol{p}}_{\text{NED},i}^{d}(t) - \boldsymbol{k}_{di}\dot{\boldsymbol{e}}_i - \boldsymbol{k}_{pi}\boldsymbol{e}_i, \tag{3.20}$$

forces the error dynamics to

$$\ddot{\boldsymbol{e}}_i(t) + \boldsymbol{k}_{di}\dot{\boldsymbol{e}}_i(t) + \boldsymbol{k}_{pi}\boldsymbol{e}_i(t) = 0, \tag{3.21}$$

where $\boldsymbol{k}_{di} = \text{diag}([k_{dxi}\ k_{dyi}\ k_{dzi}])$ and $\boldsymbol{k}_{pi} = \text{diag}([k_{pxi}\ k_{pyi}\ k_{pzi}])$ are control gains. The error converges asymptotically to zero if all elements of $\boldsymbol{k}_{di}$ and $\boldsymbol{k}_{pi}$ are greater than zero.

Remark 1 *The diagonal elements of $\boldsymbol{k}_{pi}$ and $\boldsymbol{k}_{di}$ define the natural frequency and damp factor of the 2nd order error dynamics in the three S_{NED} axis of the i-th UAV. As can be seen in [42], these gains also define a control bandwidth, which must be sufficiently small to account, for example, to actuator dynamics or, in this case, the inner loop dynamics.*

3.4.1 Describing the UAV formation

This subsection is based in [15, 26, 14], but includes also a contribution from this author.

The fleet formation is achieved by generating a distinct desired position $\boldsymbol{p}^d_{\mathrm{NED},i}(t) = [\,p^d_{xi}(t)\; p^d_{yi}(t)\; p^d_{zi}(t)\,]^T$ to each of the i-th UAV, which is described as

$$\boldsymbol{p}^d_{\mathrm{NED},i}(t) = \boldsymbol{p}_{\mathrm{NED},0}(t) + \tilde{\boldsymbol{p}}_{\mathrm{NED},i}(t), \tag{3.22}$$

where $\boldsymbol{p}_{\mathrm{NED},0}(t)$ is the leader or virtual leader position and $\tilde{\boldsymbol{p}}_{\mathrm{NED},i}(t) = [\,\tilde{x}_i(t) \quad \tilde{y}_i(t) \quad \tilde{z}_i(t)\,]^T$ is the desired (time varying) clearance, which is described in the reference frame. The virtual leader position is known by all UAVs, and describes a smooth trajectory as a function of time.

Being $\tilde{\boldsymbol{p}}_{\mathrm{NED},i}(t)$ defined in S_{NED}, it is adequate to describe formations that does not rotates along the leader's movement, as in Fig. 3.1a. However, by using [14]

$$\tilde{\boldsymbol{p}}_{\mathrm{NED},i}(t) = \boldsymbol{D}^L_{\mathrm{NED}}(t)\tilde{\boldsymbol{p}}_{Li}(t), \tag{3.23}$$

where $\tilde{\boldsymbol{p}}_{Li}(t)$ is the clearance vector described in a leader's frame, such as S_{b0}, S_{sw0} or $S_{\chi 0}$, and $\boldsymbol{D}^L_{\mathrm{NED}}(t)$ is a DCM that rotates from this leader's frame to the reference frame S_{NED}, it is achieved a formation that rotates along the leader's movement, as in Fig 3.1b.

Since the position error $\boldsymbol{e}_i(t)$ in (3.19) is function of $\boldsymbol{p}^d_{\mathrm{NED},i}(t)$ in (3.22), and the error dynamics in (3.21) includes up to the second derivative of the position error, the first and second derivatives of $\boldsymbol{p}^d_{\mathrm{NED},i}(t)$ must be computed. In [14], the author argues that, being the desired position a pre-planned path, it can be trivially be derived, since the path is projected to be made of smooth curves, but these derivatives are not provided. As a contribution of this thesis, the analytic derivatives of $\boldsymbol{p}^d_i(t)$, which accounts the leader's movement, are provided below. The development is made in a way that allows a non-virtual leader maneuvering freely, allowing also the use of paths that are defined in real time instead of being pre-planned. The derivatives are

$$\dot{\boldsymbol{p}}^d_{\mathrm{NED},i}(t) = \dot{\boldsymbol{p}}_{\mathrm{NED},0}(t) + \dot{\tilde{\boldsymbol{p}}}_{\mathrm{NED},i}(t), \tag{3.24}$$

$$\ddot{\boldsymbol{p}}^d_{\mathrm{NED},i}(t) = \ddot{\boldsymbol{p}}_{\mathrm{NED},0}(t) + \ddot{\tilde{\boldsymbol{p}}}_{\mathrm{NED},i}(t). \tag{3.25}$$

where, using the Theorem of Coriolis [50], the derivatives of $\tilde{\boldsymbol{p}}_i(t)$ in Eq. (3.23) are calculated as

$$\dot{\tilde{\boldsymbol{p}}}_{\mathrm{NED},i}(t) = \boldsymbol{D}^L_{\mathrm{NED}}(t)\left[\dot{\tilde{\boldsymbol{p}}}_{Li}(t) + \boldsymbol{\omega}^{L,\mathrm{NED}}_L(t) \times \tilde{\boldsymbol{p}}_{Li}(t)\right], \tag{3.26}$$

$$\begin{aligned}\ddot{\tilde{\boldsymbol{p}}}_{\mathrm{NED},i}(t) = \boldsymbol{D}^L_{\mathrm{NED}}(t)\Big\{&\ddot{\tilde{\boldsymbol{p}}}_{Li}(t) + 2\boldsymbol{\omega}^{L,\mathrm{NED}}_L(t) \times \dot{\tilde{\boldsymbol{p}}}_{Li}(t) + \dot{\boldsymbol{\omega}}^{L,\mathrm{NED}}_L(t) \times \tilde{\boldsymbol{p}}_{Li}(t) \\ &+ \boldsymbol{\omega}^{L,\mathrm{NED}}_L(t) \times [\boldsymbol{\omega}^{L,\mathrm{NED}}_L(t) \times \tilde{\boldsymbol{p}}_{Li}(t)]\Big\},\end{aligned} \tag{3.27}$$

where $\boldsymbol{\omega}^{L,\mathrm{NED}}_L(t)$ is the angular velocity between the rotating leader's frame and the reference frame.

As example, by defining $\boldsymbol{D}^{L}_{\text{NED}}(t) = \boldsymbol{D}^{sw0}_{\text{NED}}(t)$ (the transpose of (2.7) with $i = 0$) and $\boldsymbol{\omega}_{L}^{L,\text{NED}}(t) = \boldsymbol{\omega}_{L}^{sw0,\text{NED}}(t)$ (Eq. (2.10)), it is achieved formation description aligned with the (virtual) leader's trajectory as in, e.g., [13, 15].

If, instead, $\boldsymbol{D}^{L}_{\text{NED}}(t) = \boldsymbol{D}^{\chi 0}_{\text{NED}}(t)$ (the transpose of (2.8)) and $\boldsymbol{\omega}_{L}^{L,\text{NED}}(t) = \boldsymbol{\omega}_{L}^{\chi 0,\text{NED}}(t)$ (Eq. (2.11)), it is achieved a formation description aligned with the horizontal projection of the (virtual) leader's trajectory, used in, e.g, [11, 3, 14].

Finally, another option is $\boldsymbol{D}^{L}_{\text{NED}}(t) = \boldsymbol{D}^{b0}_{\text{NED}}(t)$ (the transpose of (2.3)) and $\boldsymbol{\omega}_{L}^{L,\text{NED}}(t) = \boldsymbol{\omega}_{L}^{b0,\text{NED}}(t)$ (Eq. (2.9)). This can be useful, for example, for maneuvers involving close interaction between the leader and the followers, such as to a boom-receptacle automatic aerial refueling.

It is worth to note that when using the non-virtual leader's body frame, the angular velocity $\boldsymbol{\omega}_{L}^{b0,\text{NED}}(t)$ is the body angular velocity, which can be directly measured by a gyro sensor at the leader. By using a non-virtual leader and any of the wind frame variants, the ground velocity obtained from GPS sensor or from a navigation algorithm must be used. When using a virtual leader approach, its trajectory is smooth, pre-known, and artificially generated, in a way that $\boldsymbol{\omega}_{L}^{L,\text{NED}}(t)$ can be pre-calculated analytically or numerically with arbitrary precision depending on how the trajectory is created.

3.5 SIMPLIFIED WIND FRAME NLDI AUTONOMOUS FORMATION FLIGHT APPROACH

In this approach, the formation is modeled in $S_{\chi 0}$. Figure 3.2 depicts the formation, showing the leader and an i-th follower.

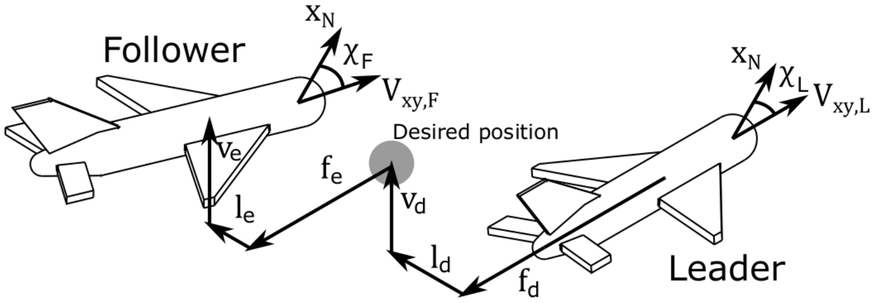


Figure 3.2: Flight formation definition.

The leader and follower positions are respectively $\boldsymbol{p}_{\text{NED,0}}(t)$ and $\boldsymbol{p}_{\text{NED,i}}(t)$, described on S_{NED}. The distance vector $\Delta\boldsymbol{p}_{\text{NED,i}}(t)$ between both aircraft is

$$\Delta\boldsymbol{p}_{\text{NED},i}(t) = \boldsymbol{p}_{\text{NED},0}(t) - \boldsymbol{p}_{\text{NED},i}(t). \tag{3.28}$$

This distance can be described in $S_{\chi 0}$ by

$$\Delta\boldsymbol{p}_{\chi 0,i}(t) = [f_i(t)\ l_i(t)\ v_i(t)]^T = \boldsymbol{D}_{\chi 0}^{\text{NED}}(t)\Delta\boldsymbol{p}_{\text{NED},i}(t), \tag{3.29}$$

where f, l and v means, respectively, forward, lateral and vertical. The distances $f_i(t)$, $l_i(t)$, and $v_i(t)$ are positive if, respectively, the leader aircraft is forward, to the right, and below the follower one.

The objective of the formation is to maintain a desired relative position $\Delta\boldsymbol{p}^d_{\chi 0,i} = [f_{d,i}(t)\ l_{d,i}(t)\ v_{d,i}(t)]^T$ between both aircraft. The difference between the desired and actual relative position is the formation error $\boldsymbol{e}_{\chi 0,i}(t) = [f_{e,i}(t)\ l_{e,i}(t)\ v_{e,i}(t)]^T$

$$\boldsymbol{e}_{\chi 0,i}(t) = \Delta\boldsymbol{p}_{\chi 0,i}(t) - \Delta\boldsymbol{p}^d_{\chi 0,i}(t) = \boldsymbol{D}^{\text{NED}}_{\chi 0}(t)\Delta\boldsymbol{p}_{\text{NED},i}(t) - \Delta\boldsymbol{p}^d_{\chi 0,i}(t). \tag{3.30}$$

Figure 3.3 shows the control structure used in the proposed formation flight controller. The structure is separated in the outer and the inner loops. The outer loop is further separated in the NLDI equations and in the linear state feedback controller, which is shown with thicker lines. The inner loop controller is explained in Chapter 2.

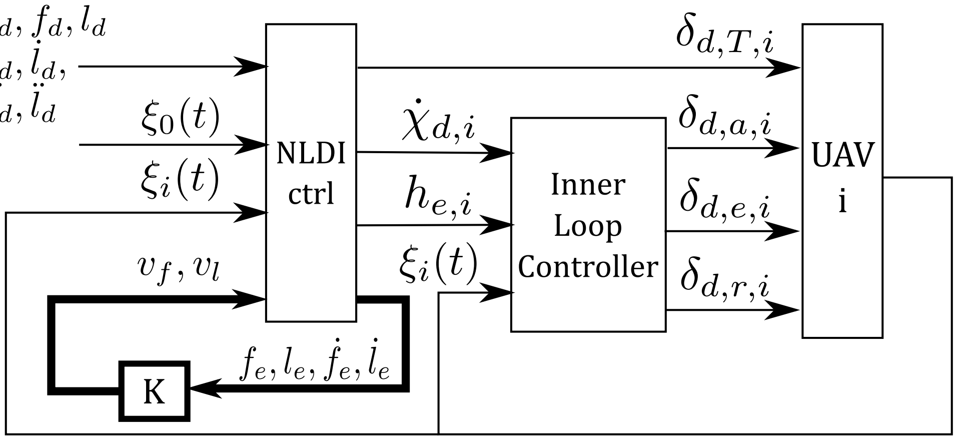


Figure 3.3: Outer and inner loop controllers.

The formation control is given in the outer loop. Receiving the formation error and its derivatives, and information about the leader and the follower state vectors, the controller provides the $\dot{\chi}_{d,i}(t)$ and $v_{e,i}(t)$ to the follower's inner loop, aiming to reduce this formation error. In the proposed controller, the outer loop directly sets the engine throttle command $\delta_{T,i}(t)$, without an intermediate inner loop step.

As a contribution, here it is proposed an outer loop NLDI autonomous formation flight controller, described in the leader's $S_{\chi 0}$ frame. The NLDI cancels the non-linearity relating the desired forward and lateral relative position to $\dot{\chi}_{d,i}(t)$ and $\delta_{T,i}(t)$. The formation error model becomes linear, and a simple state feedback regulator is used to remove the error. The vertical error is also calculated by the outer loop, but it is uncoupled to the lateral and forward error.

3.5.1 Proposed formation controller

The NLDI AFF controller proposed in [11] is called from now on as non linear formation controller (NLFC) and the controller proposed here is called as extended non linear formation controller (E-NLFC). The E-NLFC, also based on NLDI, uses a less restrictive set of assumptions than NLFC. The differences between the NLFC and the proposed E-NLFC are highlighted and discussed after obtaining all E-NLFC equations. It is worth note that, in [11], the formation errors $f_{e,i}(t)$ and $l_{e,i}(t)$, and its derivatives $\dot{f}_{e,i}(t)$, $\dot{l}_{e,i}(t)$, $\ddot{f}_{e,i}(t)$, and $\ddot{l}_{e,i}(t)$ are obtained directly, whereas here the full distances are obtained first, and only in the end of the deduction the formation error and its derivatives are calculated. It is also worth note that the notation here and in [11] are distinct.

From (3.29) and the definition of $\boldsymbol{D}_{\chi 0}^{\text{NED}}(t)$ in (2.8), it is clear that the vertical elements of the formation definition are uncoupled to the horizontal ones. The horizontal components $f_i(t)$ and $l_i(t)$ can be obtained from (3.29) as

$$\begin{bmatrix} f_i(t) \\ l_i(t) \end{bmatrix} = \begin{bmatrix} \cos\chi_0(t) & \sin\chi_0(t) \\ -\sin\chi_0(t) & \cos\chi_0(t) \end{bmatrix} \begin{bmatrix} x_0(t) - x_i(t) \\ y_0(t) - y_i(t) \end{bmatrix}. \tag{3.31}$$

The input-output linearization needs to derive the output until the input explicit appears.

The first derivative can be obtained from (3.29) by the use of the Theorem of Coriolis (2.12)

$$\Delta\dot{\boldsymbol{p}}_i(t) = (\dot{\boldsymbol{p}}_0(t) - \dot{\boldsymbol{p}}_i(t)) + (\boldsymbol{\omega}^{\text{NED},\chi_0}(t) \times \Delta\boldsymbol{p}_i(t)), \tag{3.32}$$

where $\dot{\boldsymbol{p}}_i$ is the inertial velocity of the aircraft, i.e., the derivative of $\boldsymbol{p}_i$ evaluated at S_{NED}. The first parenthesis is the relative velocity between the aircraft. The second parenthesis comes from the Theorem of Coriolis, an effect generated by the use of a rotating frame. $\boldsymbol{\omega}^{\text{NED},\chi_L}(t)$ is the relative angular velocity between S_{NED} and S_χ, and the distance $\Delta\boldsymbol{p}_i(t)$ between the leader and the i-th aircraft is the lever-arm between the origin of $S_{\chi 0}$, the point of rotation, and the CG of the follower aircraft.

From (2.11), $\boldsymbol{\omega}_{\text{NED}}^{\text{NED},\chi_L}(t) = [0\ 0\ -\dot{\chi}_0(t)]^T$. Defining $V_{\text{xy}}(t)$ as the horizontal speed, i.e., the magnitude of the horizontal projection of $\dot{\boldsymbol{p}}(t)$ and defining $\Delta\chi_i(t)$ as $\chi_0(t) - \chi_i(t)$, (3.32) can be rewritten in the $S_{\chi 0}$ frame as

$$\begin{bmatrix} \dot{f}_i(t) \\ \dot{l}_i(t) \end{bmatrix} = V_{\text{xy},0}(t) \begin{bmatrix} 1 \\ 0 \end{bmatrix} - V_{\text{xy},i}(t) \begin{bmatrix} \cos\Delta\chi_i(t) \\ -\sin\Delta\chi_i(t) \end{bmatrix} + \dot{\chi}_0(t) \begin{bmatrix} l_i(t) \\ -f_i(t) \end{bmatrix}. \tag{3.33}$$

The derivative of (3.33) can be obtained

$$\begin{bmatrix} \ddot{f}_i(t) \\ \ddot{l}_i(t) \end{bmatrix} = \begin{bmatrix} \dot{V}_{\text{xy},0}(t) \\ 0 \end{bmatrix} - \dot{V}_{\text{xy},i}(t) \begin{bmatrix} \cos\Delta\chi_i(t) \\ -\sin\Delta\chi_i(t) \end{bmatrix} + V_{\text{xy},i}(t)\Delta\dot{\chi}_i(t) \begin{bmatrix} \sin\Delta\chi_i(t) \\ \cos\Delta\chi_i(t) \end{bmatrix} \\ + \ddot{\chi}_0(t) \begin{bmatrix} l_i(t) \\ -f_i(t) \end{bmatrix} + \dot{\chi}_0(t) \begin{bmatrix} \dot{l}_i(t) \\ -\dot{f}_i(t) \end{bmatrix}. \tag{3.34}$$

Extra manipulation in (3.34) is needed to explicitly show the throttle $T_i(t)$. The horizontal speed $V_{\text{xy},i}(t)$ can related to the speed $V_i(t)$ and the flight path angle $\gamma_i(t)$, and its derivative can be obtained after

$$V_{\text{xy},i}(t) = V_i(t)\cos\gamma_i(t), \tag{3.35}$$

$$\dot{V}_{\text{xy},i}(t) = \dot{V}_i(t)\cos\gamma_i(t) - V_i(t)\dot{\gamma}_i(t)\sin\gamma_i(t) = \dot{V}_i(t)\frac{V_{\text{xy},i}(t)}{V_i(t)} + V_{D,i}(t)\dot{\gamma}_i(t). \tag{3.36}$$

By using the definition of $\dot{V}_i$ from (2.37) in (3.36) and inserting the obtained definition of $\dot{V}_{\text{xy},i}$ in (3.34)

$$\begin{bmatrix} \ddot{f}_i(t) \\ \ddot{l}_i(t) \end{bmatrix} = \begin{bmatrix} \dot{V}_{\text{xy},i}(t) \\ 0 \end{bmatrix} - \left(V_{\text{D},i}(t)\dot{\gamma}_i(t) + \frac{\omega_2}{V_i(t)} V_{\text{xy},i}(t) \right) \begin{bmatrix} \cos\Delta\chi_i(t) \\ -\sin\Delta\chi_i(t) \end{bmatrix} - V_{\text{xy},i}(t)\dot{\chi}_0(t) \begin{bmatrix} \sin\Delta\chi_i(t) \\ \cos\Delta\chi_i(t) \end{bmatrix} \\ + \begin{bmatrix} l_i(t) & \dot{l}_i(t) \\ -f_i(t) & -\dot{f}_i(t) \end{bmatrix} \begin{bmatrix} \ddot{\chi}_0(t) \\ \dot{\chi}_0(t) \end{bmatrix} - V_{\text{xy},i}(t) \begin{bmatrix} \sin\Delta\chi_i(t) & \frac{\omega_1}{V_i(t)}\cos\Delta\chi_i \\ \cos\Delta\chi_i(t) & -\frac{\omega_1}{V_i(t)}\sin\Delta\chi_i \end{bmatrix} \begin{bmatrix} \dot{\chi}_i(t) \\ T_i(t) \end{bmatrix}. \quad (3.37)$$

Equation (3.37) is written in the standard MIMO input-output linearization (3.10), where

$$\boldsymbol{B}_i(t) = -V_{\text{xy},i}(t) \begin{bmatrix} \sin\Delta\chi_i(t) & \frac{\omega_1}{V_i(t)}\cos\Delta\chi_i(t) \\ \cos\Delta\chi_i(t) & -\frac{\omega_1}{V_i(t)}\sin\Delta\chi_i(t) \end{bmatrix}, \qquad \boldsymbol{u}_i(t) = \begin{bmatrix} \dot{\chi}_i(t) \\ T_i(t) \end{bmatrix}, \quad (3.38)$$

and $\boldsymbol{f}$ are all the other elements in the right of the equal symbol in (3.37).

Using (3.11), it is obtained

$$\begin{bmatrix} \dot{\chi}_{d,i}(t) \\ T_{d,i}(t) \end{bmatrix} = \boldsymbol{B}^{-1}(t) \left(\begin{bmatrix} v_{f,i}(t) \\ v_{l,i}(t) \end{bmatrix} + \left(\dot{\gamma}_i(t) V_{D,i}(t) + \frac{\omega_2}{V_i(t)} V_{\text{xy},i}(t) \right) \begin{bmatrix} \cos\Delta\chi_i(t) \\ -\sin\Delta\chi_i(t) \end{bmatrix} \right. \\ \left. + V_{\text{xy},i}(t)\dot{\chi}_0(t) \begin{bmatrix} \sin\Delta\chi_i(t) \\ \cos\Delta\chi_i(t) \end{bmatrix} - \begin{bmatrix} \dot{V}_{\text{xy},0}(t) \\ 0 \end{bmatrix} - \begin{bmatrix} l_i(t) & \dot{l}_i(t) \\ -f_i(t) & -\dot{f}_i(t) \end{bmatrix} \begin{bmatrix} \ddot{\chi}_0(t) \\ \dot{\chi}_0(t) \end{bmatrix} \right), \quad (3.39)$$

which can be rewritten as

$$\begin{bmatrix} \dot{\chi}_{d,i}(t) \\ T_{d,i}(t) \end{bmatrix} = \begin{bmatrix} \dot{\chi}_0(t) \\ -\frac{\omega_2}{\omega_1} - \frac{\dot{\gamma}_i(t) V_{D,i}(t)}{\omega_1 \cos\gamma_i(t)} \end{bmatrix} + \boldsymbol{B}^{-1}(t) \left(\begin{bmatrix} v_{f,i}(t) \\ v_{l,i}(t) \end{bmatrix} - \begin{bmatrix} \dot{V}_{\text{xy},0}(t) \\ 0 \end{bmatrix} - \begin{bmatrix} l_i(t) & \dot{l}_i(t) \\ -f_i(t) & -\dot{f}_i(t) \end{bmatrix} \begin{bmatrix} \ddot{\chi}_0(t) \\ \dot{\chi}_0(t) \end{bmatrix} \right), \quad (3.40)$$

where

$$\boldsymbol{B}_i^{-1}(t) = -\frac{1}{V_{\text{xy},i}(t)} \begin{bmatrix} \sin\Delta\chi_i(t) & \cos\Delta\chi_i(t) \\ \frac{V_i(t)}{\omega_1}\cos\Delta\chi_i(t) & -\frac{V_i(t)}{\omega_1}\sin\Delta\chi_i(t) \end{bmatrix}, \quad (3.41)$$

and the included subscript d in $\dot{\chi}_{d,i}(t)$ and $T_{d,i}(t)$ means that they are the *desired* rate-of-turn and thrust, generated by the E-NLFC and used as input for the inner controllers. $v_{f,i}(t)$ and $v_{l,i}(t)$ are the equivalent input, which provides

$$\ddot{f}_i(t) = v_{f,i}(t), \qquad \ddot{l}_i(t) = v_{l,i}(t). \quad (3.42)$$

As explained in Section 3.2, being the relationship linear, any linear control technique can be used. A null tracking error can be achieved by the following solution

$$v_{f,i}(t) = \ddot{f}_{d,i}(t) - k_{dfi}\dot{f}_{e,i}(t) - k_{pfi}f_{e,i}(t), \qquad v_{l,i}(t) = \ddot{l}_{d,i}(t) - k_{dli}\dot{l}_{e,i}(t) - k_{pli}l_{e,i}(t). \quad (3.43)$$

where if the control coefficients $k_{dfi} > 0$, $k_{pfi} > 0$, $k_{dli} > 0$, and $k_{pli} > 0$, the error asymptotically converges to zero.

Equation (3.43) is shown by thicker lines in Fig. 3.3.

The calculated desired rate-of-turn $\chi_{d,i}(t)$ is sent to the inner loop equation (2.29). The computation of $\delta_T(t)$ from $T_d(t)$ is computed by using (2.41).

Equations (3.31), (3.33), (3.40), and (2.41) define the *NLFC ctrl* block from Fig. 3.3.

Remark 2 *The proposed controller is a generalization of the NLFC presented in [11], as is shown now. First, it is assumed in NLFC that the formation shape does not vary with time, i.e.,*

$$\dot{f}_{d,i}(t) = 0, \quad \dot{l}_{d,i}(t) = 0, \quad \ddot{f}_{d,i}(t) = 0, \quad \ddot{l}_{d,i}(t) = 0. \tag{3.44}$$

Restricting the leader's flight to a straight line or a low-intensity turn, and assuming that the desired distances $f_{d,i}(t)$ and $l_{d,i}(t)$ are small, it is obtained

$$\begin{aligned} \dot{\chi}_0(t) f_i(t) &= \dot{\chi}_0(t)(f_{e,i}(t) + f_{d,i}(t)) \approx \dot{\chi}_0(t) f_{e,i}(t), \\ \dot{\chi}_0(t) l_i(t) &= \dot{\chi}_0(t)(l_{e,i}(t) + l_{d,i}(t)) \approx \dot{\chi}_0(t) l_{e,i}(t). \end{aligned} \tag{3.45}$$

By using (3.44) *and* (3.45), (3.33) *is simplified to*

$$\begin{bmatrix} \dot{f}'_i(t) \\ \dot{l}'_i(t) \end{bmatrix} = \begin{bmatrix} \dot{f}'_{e,i}(t) \\ \dot{l}'_{e,i}(t) \end{bmatrix} = \begin{bmatrix} V_{\text{xy},0}(t) \\ 0 \end{bmatrix} - V_{\text{xy},i}(t) \begin{bmatrix} \cos \Delta\chi_i(t) \\ -\sin \Delta\chi_i(t) \end{bmatrix} + \dot{\chi}_0 \begin{bmatrix} l_{e,i}(t) \\ f_{e,i}(t) \end{bmatrix}, \tag{3.46}$$

where the upperscript $'$ indicates a value calculated in NLFC in a different way than the developed equations from E-NLFC.

It can be assumed that the leader's altitude change and horizontal velocity are constant, i.e.,

$$\dot{\gamma}_0(t) \approx 0, \qquad \dot{V}_{\text{xy},0}(t) \approx 0. \tag{3.47}$$

Using all these simplifications, the control law found in [11] is

$$\begin{bmatrix} \dot{\chi}'_{d,i}(t) \\ T'_{d,i}(t) \end{bmatrix} = \begin{bmatrix} \dot{\chi}_0(t) \\ -\frac{\omega_2}{\omega_1} \end{bmatrix} + \boldsymbol{B}^{-1}(t) \left(\begin{bmatrix} v'_{f,i}(t) \\ v'_{l,i}(t) \end{bmatrix} - \begin{bmatrix} l_{e,i}(t) & \dot{l}'_{e,i}(t) \\ -f_{e,i}(t) & -\dot{f}'_{e,i}(t) \end{bmatrix} \begin{bmatrix} \ddot{\chi}_0(t) \\ \dot{\chi}_0(t) \end{bmatrix} \right), \tag{3.48}$$

where

$$v'_{f,i} = -k_{dfi} \dot{f}'_{e,i}(t) - k_{pfi} f_{e,i}(t), \qquad v'_{l,i} = -k_{dli} \dot{l}'_{e,i}(t) - k_{pli} l_{e,i}(t). \tag{3.49}$$

Since all the discussed simplifications in remark are not used in the proposed controller, the obtained NLDI control laws for the proposed E-NLFC provides lower track error in a broader AFF scenario than NLFC, which includes a more maneuvering leader, a less restrictive and time-varying formation shape. To the best of the authors knowledge, the presented modifications were never suggested before in an input-output linearization formation flight controller. As an example, more recent works [53, 30] use the NLFC exactly how was implemented in [11] and the Eq. (3.46) is used in [12] also as implemented in [11]. In [29], the review section lists [11] as the feedback linearization approach existent in literature.

The vertical outer loop controller is implemented as in [11]. Using (3.30), the vertical error v_e can be obtained as

$$v_{e,i}(t) = z_0(t) - z_i(t) - v_{d,i}. \tag{3.50}$$

The altitude error $v_{e,i}(t)$ is used in an inner loop as the input of the controller shown in (2.24). The output is a command $\delta_{e,i}(t)$ to the elevator.

3.5.2 Controller stability

It must be evaluated first in which subset of the state space the control is valid. Singularities can occur if the $\boldsymbol{B}(t)$ matrix from (3.38) has no inverse, i.e., null determinant and/or if there is a division by zero.

The determinant of $\boldsymbol{B}(t)$ is

$$\det(\boldsymbol{B}(t)) = -V_{\mathrm{xy},i}(t)\frac{\omega_1}{V_i(t)} = -\cos\gamma_i(t)\frac{\cos\alpha_i(t)\cos\beta_i(t)}{m_i}. \tag{3.51}$$

The determinant is null only in extreme situations in which $\gamma_i(t)$ or $\alpha_i(t)$ or $\beta_i(t)$ is equal to $\pm 90°$, which is highly unlikely in usual flights.

There is also a term

$$-\frac{\omega_2}{\omega_1} - \frac{\dot{\gamma}_i(t)V_D}{\omega_1\cos\gamma_i(t)}, \tag{3.52}$$

in (3.40), in which the denominator is equal to zero in the same unlikely conditions as the determinant of $\boldsymbol{B}(t)$.

The stability of nonlinear controllers must be evaluated. The proposed controller is based on input-output feedback linearization. If the relative degree r is equal to the state dimension n, the linearization is complete, and no extra stability evaluation must be made. However, if there is a difference between both numbers, it means that there is $n - r$ unobserved states that must be evaluated.

Since there are two outputs that are derived 2 times, $r = 4$. To evaluate the state dimension, the state vector must be known. The formation is composed by at least two UAVs, each one modeled by 12 states. However, the leader flies independently, in a way that the formation does not affect its own states, so they can be considered stable assuming that the leader has its own controller.

The outer loop does not control directly the UAV model, but generates a trajectory that must be followed by the inner loop. Intuitively, assuming that the inner loop stabilizes the UAV, the outer loop will destabilize the vehicle only if it generates an unbounded or non-smooth trajectory. In this way, only the outer loop stability must be evaluated if the inner loop is already stable. This approach is used in [33], where a sliding mode controller is used to achieve a pursuit-based formation controller.

Using (2.43), the horizontal trajectory can be described by two positions, the velocity magnitude and the course angle, which results in four states, i.e,

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \\ \dot{V}_{\mathrm{xy},i}(t) \\ \dot{\chi}_i(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cos\chi_i(t) & 0 \\ 0 & 0 & \sin\chi_i(t) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i(t) \\ y_i(t) \\ V_{\mathrm{xy},i}(t) \\ \chi_i(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{V}_{\mathrm{xy},i,d}(t) \\ \dot{\chi}_{i,d}(t) \end{bmatrix}, \tag{3.53}$$

where $\dot{V}_{\mathrm{xy},F,d}(t)$ is the desired acceleration in horizontal component, which is obtained by changing the total acceleration $\dot{V}_{F,d}(t)$ via thrust $T_{F,d}(t)$. $\dot{V}_{\mathrm{xy},F,d}(t)$ is a control input generated by the inner loop.

There is also the vertical components of the trajectory, but the vertical control is made by another controller.

This indicates that the controller fully linearizes the outer loop. In this case, the obtained input-output linearization is stable if the linear system is stable, i.e., if (3.42) is stable.

3.6 PROPOSED PATH PLANNING ALGORITHM

In the proposed formation flight controller from Section 3.5, the formation shape is described in the leader's frame $S_{\chi 0}$, whereas in the controller from Section 3.4, the formation can be described in any frame, including $S_{\chi 0}$. The following proof-of-concept path planner shows that by describing the formation shape in $S_{\chi 0}$, the path planning of cooperative UAVs flying in formation in an ambient with no obstacles is simplified. This path planning can be used, for example, to an autonomous formation flight refueling or rendezvous. The path planner is explained by using the formation flight controller from Section 3.5 as convenience, but it can be readily applied in the controller from Section 3.4 by defining $\boldsymbol{D}^{L}_{\mathrm{NED}}(t) = \boldsymbol{D}^{\chi 0}_{\mathrm{NED}}(t)$ in (3.23).

From (3.42), it can be seen that the nonlinear nonholonomic-constrained movement is converted to a double integrator to each relative direction.

Since the obtained linear model has, as input, the accelerations $\ddot{f}_{d,i}(t)$ and $\ddot{l}_{d,i}(t)$, the trajectory can change abruptly its accelerations, generating continuous velocities and positions. It i assumed that there are maximum and minimum allowable accelerations and speed

$$\begin{aligned} a_{f,i,\min} \leq \ddot{f}_{d,i}(t) \leq a_{f,i,\max}, \qquad a_{l,i,\min} \leq \ddot{l}_{d,i}(t) \leq a_{l,i,\max}, \\ v_{f,i,\min} \leq \dot{f}_{d,i}(t) \leq v_{f,i,\max}, \qquad v_{l,i,\min} \leq \dot{l}_{d,i}(t) \leq v_{l,i,\max}, \end{aligned} \tag{3.54}$$

in which restrictions are dependent on the engine maximum thrust, maximum allowable turn-ratio and the in the leader's trajectory parameters, such as leader's actual velocity and turn-ratio. To achieve a final and fixed position, it is assumed that the minimum accelerations and velocities are negatives and the maximum accelerations and velocities are positives.

Maximum (minimum) acceleration is used to achieve maximum (minimum) velocity, then the vehicle stays in the maximum (minimum) velocity until the point in which using the minimum (maximum) acceleration will result in the vehicle in the correct desired position when the speed is null. For an i-th vehicle in an initial relative forward position $f_i(t_{0,i})$ and relative forward velocity $\dot{f}_i(t_{0,i})$, and desired final position $f_{\mathrm{d,i,final}}$ and desired final velocity $\dot{f}_{d,i,\mathrm{final}} = 0$, the planned path is described by

$$\dot{f}_{d,i}(t) = \int_{t_{0,i}}^{t} \ddot{f}_{d,i}(\tau) d\tau, \qquad \dot{f}_{d,i}(t_{0,i}) = \dot{f}_i(t_{0,i}), \tag{3.55}$$

$$f_{d,i}(t) = \int_{t_{0,i}}^{t} \dot{f}_{d,i}(\tau) d\tau, \qquad f_{d,i}(t_{0,i}) = f_i(t_{0,i}), \tag{3.56}$$

$$\ddot{f}_{d,i}(t) = \begin{cases} 0, & t < t_{0,i}, \quad t_{1,i} < t < t_{2,i}, \quad t > t_{3,i}, \\ a_{f,i,\min}, & t_{0,i} < t < t_{1,i}, \\ a_{f,i,\max}, & t_{2,i} < t < t_{3,i}. \end{cases} \tag{3.57}$$

It is worth noting that the $a_{f,i,\min}$ is applied first, in $t_{0,i}$, under the hypothesis of an approaching movement, because a negative relative acceleration will provide a negative relative velocity, which reduces $f_i(t)$ over time. In lateral movement, a positive acceleration will result in a right-to-left movement, and left-to-right is generated by a negative acceleration. The time instants $t_{1,i}$ to $t_{3,i}$ can be obtained from point-mass kinematic equations

$$t_{1,i} - t_{0,i} = (v_{f,i,\max} - \dot{f}_c(t_{0,i}))/a_{f,i,\min}, \tag{3.58}$$

$$t_{3,i} - t_{2,i} = (0 - v_{f,i,\max})/a_{f,i,\max}, \tag{3.59}$$

$$f_i(t_{3,i}) = f_{d,i,\text{final}} = f_i(0) + s_{1,i} + s_{2,i} + s_{3,i}, \tag{3.60}$$

where $s_{1,i}$ and $s_{3,i}$ are the variations in position generated during the constant acceleration movements and $s_{2,i}$ is the variation in position caused by the constant speed movement. Again, $a_{f,i,\min}$ and $a_{f,i,\max}$ must be swapped, and $v_{f,i,\max}$ changed to $v_{f,i,\min}$ in movements in opposite direction. Similar equations are used in the lateral path planner.

An example of planned path can be found in Fig. 3.6 in the simulations in Section 3.8. Note that the path planner is evaluated only once, at the beginning of the flight formation. The path planner provides the inputs $f_{d,i}(t)$, $l_{d,i}(t)$, and its derivatives to the NLDI equations, as shown in Fig. 3.3.

3.6.0.1 Collision avoidance

Here it is assumed that the UAVs are flying in a same horizontal plane, but the concepts can be easily adapted to include movements in the vertical axis. The collision avoidance algorithm is only a proof-of-concept to demonstrate why the proposed $S_{\chi 0}$-NLDI formation control is interesting.

By evaluating the maneuverability of each aircraft, and the precision and sample time of sensors that measure the distance between them, it can be defined a minimum distance $\Delta p_{\min}$ in which the UAV can securely maneuver and move away from each other without colliding. This defines a security circular area (or spheroidal volume) around each UAV. If the planned trajectories respect this minimum distance restriction, collision is avoided.

Since the proposed path planner describes independently the generated forward $f_{d,i}(t)$ and lateral $l_{d,i}(t)$ components of the trajectory of the i-th follower, the planned lateral and forward distances $\Delta f_{i,j}$ and $\Delta l_{i,j}$ between the i-th and j-th UAV are

$$\begin{aligned} \Delta f_{i,j}(t) &= |f_{d,i}(t) - f_{d,j}(t)|, \\ \Delta l_{i,j}(t) &= |l_{d,i}(t) - l_{d,j}(t)|. \end{aligned} \tag{3.61}$$

There is also the leader UAV, if non-virtual, which position is, by the formation definition, zero. The planned distance between the leader and a follower i is

$$\Delta f_{i,0}(t) = |f_{i,0}(t)|, \qquad \Delta l_{i,0}(t) = |l_{i,0}(t)|. \tag{3.62}$$

The distance can be calculated as

$$\Delta p_{i,j}(t) = \sqrt{\Delta f_{i,j}^2(t) + \Delta l_{i,j}^2(t)}. \tag{3.63}$$

If $\Delta p_{i,j}(t_c) < \Delta p_{\min}$, there is a collision in t_c between the i-th and j-th aircraft. Alternatively, the security circular area can be approximated to a square one, and the collision criteria is changed to: if $\Delta f_{i,j}(t_c) < d_{\min}$ and $\Delta l_{i,j}(t_c) < d_{\min}$, there is collision in t_c. This new criteria is easier to evaluate, but it is a little more restrictive than the first one.

To avoid a collision, the path planner follow these steps:

1. A path is planned to each follower;
2. The forward and lateral distance between each UAV, including the leader, if non-virtual, are evaluated;
3. If a collision is detected, the path of one of the UAV is re-planned to avoid the the detected collision. The algorithm returns to step 2 to evaluate the new path against the path of others UAVs;
4. When it is obtained collision-free paths, the algorithm stops.

A collision occurs if two or more aircraft are in the same place at the same time. The trajectory can be changed to avoid a intersection in the paths or, more easily, the time in which an aircraft is at some point of the trajectory can be delayed. This can be made by reducing the magnitude of the maximum acceleration an/or speed of one of the followers.

An interesting alternative approach, used here, is to delay the time when the trajectory of one follower starts. Even simpler, it is possible to delay only the forward or lateral component of the trajectory. This is only possible because of the nature of the formation description, in which an aircraft can remain still when observed from the leader's perspective. This means that one aircraft can, for example, "stop" while the other assumes its position.

3.7 COMPARISON BETWEEN THE NED AND WIND NLDI VARIANTS

In Sections 3.4 and 3.5 are presented two similar but distinct approaches to achieve a formation flight controller based on nonlinear dynamic inversion. Both being presented, here it is discussed the similarities and differences:

- The S_{NED}-NLDI variant is a general purpose NLDI outer loop controller. It can be used for formation flight or trajectory tracking controllers.
- The S_{NED}-NLDI variant is flexible, allowing to describe the formation shape in any frame, such as S_{NED}, S_{sw0}, or $S_{\chi 0}$.
- The $S_{\chi 0}$-NLDI variant is specifically to formation flight modeled in $S_{\chi 0}$ frame.
- They provide a desired acceleration vector $\boldsymbol{\Gamma_i(t)}$ (S_{NED}-NLDI) or a course derivative $\dot{\chi_{d,i}}(t)$ and forward thrust $T_{d,i}$ ($S_{\chi 0}$-NLDI). As seen in Subsection 2.4.3, both are interchangeable in the point-mass model so, in practice, they can interact with the same inner loop controller in a horizontal-only formation.

- After applying the NLDI, both controlled systems act as double integrators, and the acceleration input $\boldsymbol{\tau}_i(t)$ (Eq. (3.20)) or $\boldsymbol{v}_i(t)$ (Eq. (3.43)) acts as a pole placement (or PD controller) to allocate the poles in a desired stable position.

- The position, i.e., the result of doubling the acceleration input, has distinct meanings in each variant. In S_{NED}-NLDI variant, the position is really the aircraft inertial position, whereas in $S_{\chi 0}$-NLDI it is the distance to the leader described in leader's frame. This means that one is a position-based formation control and other is a displacement based formation control.

- In S_{NED}-NLDI variant, the obtained position must respect the Dubins model (2.43) and its restriction on the minimum allowed turn radius, lower and upper limits on the speed, the aircraft must not move backwards, and others. As example in which the designer is not aware of this limitations is in [26], in which the control objective is, absurdly, to an flying fixed wing aircraft to achieve a desired position and, after, remain still.

- In $S_{\chi 0}$-NLDI variant, assuming that the leader is not making overly aggressive maneuvers, the follower movement limitation is simply upper and lower bounds in speed and acceleration. The aircraft, under the leader's reference frame, can move backwards, sideways, or remain still. This makes much easier to develop path planners, as exemplified in Subsection 3.6. It is worth to note that S_{NED}-NLDI variant can indirectly describe the formation in the leader's frame which, in this case, provides the same simplification in path planner as the $S_{\chi 0}$-NLDI variant.

- The $\boldsymbol{k}_{di}$ and $\boldsymbol{k}_{pi}$ control gains from the S_{NED}-NLDI variant adjust the dynamics in the S_{NED} frame. The k_{dfi}, k_{pfi}, k_{dli} and k_{pli} gains from the $S_{\chi 0}$-NLDI variant adjusts the gains in the leader's axis, which is similar to the follower's axis. This mean that the gains from $S_{\chi 0}$-NLDI variant can better accommodate distinct dynamics from the engine controlling the forward movement and the aileron and rudder controlling the lateral movement, which means that the $S_{\chi 0}$-NLDI variant can be better tuned.

3.8 SIMULATION RESULTS

The aircraft model, the well known S_{NED}-NLDI variant AFF controller described in Section 3.4, the S_{sw0}-NLDI variant AFF controller (NLFC) from [11], the proposed S_{sw0}-NLDI variant AFF controller (E-NLFC) described in Section 3.5 and the proposed path planner algorithm described in Subsection 3.6 are implemented in MATLAB Simulink. In all simulations, sensors are assumed perfect and leader and followers can instantly communicate with each other, i.e., there is no noise, bias or time delay in any value. Also, the UAVs controllers start trimmed to fly at straight level flight. Leader and followers use the same controller tuning in all scenarios. The control gains are given in Table 2.4. Specifically in the S_{NED}-NLDI variant, it is used a virtual leader.

The NLFC and E-NLFC are compared and the proof-of-concept path planner and collision avoidance are shown in subsection 3.8.

The S_{NED}-NLDI variant is demonstrated in Subsection 3.8.2. It is worth to note that this variant is already known in literature, and it is only an initial step to the robust controllers presented and evaluated in Chapter 4.

3.8.1 Wind-NLDI formation control approach

The NLDI outer loop uses the tuning obtained in [11], which is shown in Table 3.1, for both NLFC and the proposed E-NLFC approach. In variable formation scenarios in which is made an approximation maneuver, the path planning algorithm is executed once, at t = 0.

Table 3.1: Forward and lateral feedback gains and obtained closed-loop poles

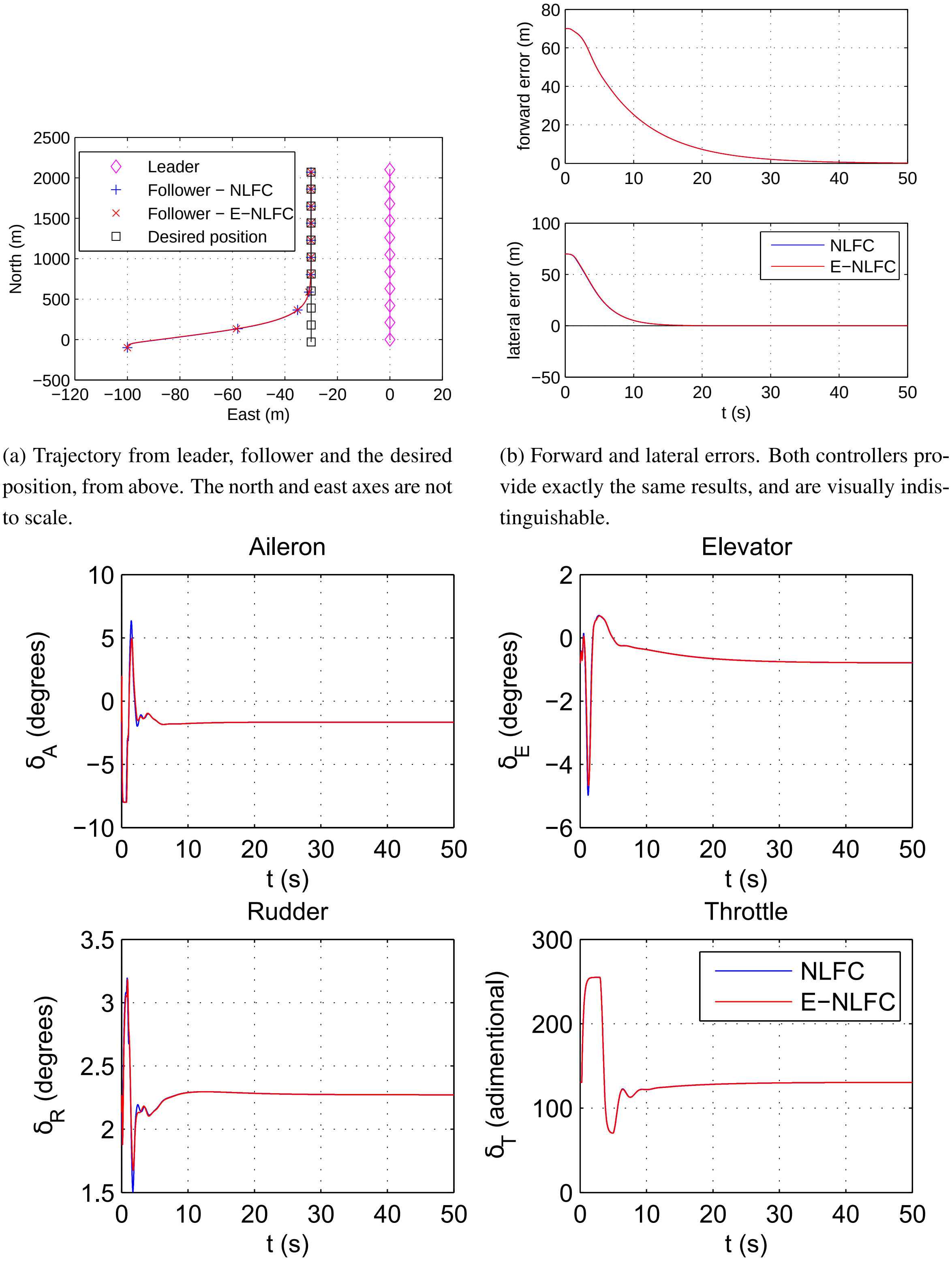
Forward control gains	k_{pfi} = 0.2419, k_{dfi} = 2.0560
Lateral control gains	k_{pli} = 0.2027, k_{dli} = 0.8894
Forward poles	-1.9307, -0.1253
Lateral poles	-0.4447 $\pm 0.0703j$

As explained in Subsection 3.5.1, the scenario extension considered in E-NLFC increases its precision compared to the NLFC if the leader is maneuvering and/or the formation has time-varying shape. The simulation scenarios are chosen to show both effects. In the simplest case, simulated in scenario *A*, neither the leader maneuvers, nor the formation varies, and the controllers must perform the same. In scenario *B*, the formation is variable, but the leader still does not maneuver. In scenario *C*, the variable-formation tracking capability of the controller is used to achieve a controller approach maneuver generated by the proposed path planner algorithm presented in Section 3.6. In scenario *D*, the leader makes an aggressive maneuver and the path planner generates a time-varying formation to achieve an approaching maneuver. Finally, the proof-of-concept collision avoidance strategy from Section 3.6 is evaluated in scenario *E*, in with two followers must approach to the leader.

3.8.1.1 Straight level flight, fixed shape formation

The scenario A is the most simple case. The leader is at a non-maneuvering straight level flight, at 42 m/s, flying to north direction, starting at position $\boldsymbol{p}_{\text{NED},0}(0)$ = [0 0 -120]$^{\text{T}}$ m. The follower has the same parameters, except its position, which is, in S_{NED}, $\boldsymbol{p}_{\text{NED},1}$ = [-100 -100 -120]$^{\text{T}}$ m, or, in $S_{\chi 0}$, $\Delta\boldsymbol{p}_{\chi 0,1}$ = [100 100 0]$^{\text{T}}$. The desired position $\Delta\boldsymbol{p}^{d}_{\chi 0,1}(0)$ is [30 30 0]$^{\text{T}}$ m. The desired position is immediately commanded, i.e., $f_{d,1}(t) = 30$ m and $l_{d,1}(t) = 30$ m for all $t > 0$. There is a small undesired deviation from the constant speed and direction movement of the leader at the start of simulation because there is an initial fine adjusting of the controller output.

Figure 3.4a shows the obtained trajectory from above, including the results from the leader, from the follower using both NLFC and the proposed E-NLFC, and the desired follower position. A symbol is used to show each aircraft position in the trajectory at 5 seconds intervals. If the follower achieves the desired position, its symbol, plus or cross, will exactly fit the square from the desired position. The north and east axes are not to scale, which magnifies the apparent lateral error. Figure 3.4b shows the forward and lateral error for both controllers. Both AFF controllers present similar results, achieving the desired position after



(a) Trajectory from leader, follower and the desired position, from above. The north and east axes are not to scale.

(b) Forward and lateral errors. Both controllers provide exactly the same results, and are visually indistinguishable.

(c) Follower actuator response.

Figure 3.4: Results from NLFC and the proposed E-NLFC controllers in scenario A.

some time. Figure 3.4c shows the actuators response, which is similar to both controllers. There is a small difference in actuator response because of the undesired leader movement at the start of the simulation, which excite the additional elements present in the proposed controller compared to the literature. This difference does not generate a significant difference in the trajectory of the follower aircraft.

3.8.1.2 Straight level flight, variable shape formation

The scenario B shows a time-varying shape formation flight, which the NLFC from literature is not projected for, but the proposed controller is. The follower must rotate around the leader. The desired position is defined by

$$\begin{aligned} l_{d,1}(t) &= 50\sin(-10\frac{\pi}{180}t), \\ f_{d,1}(t) &= 50\cos(-10\frac{\pi}{180}t). \end{aligned} \tag{3.64}$$

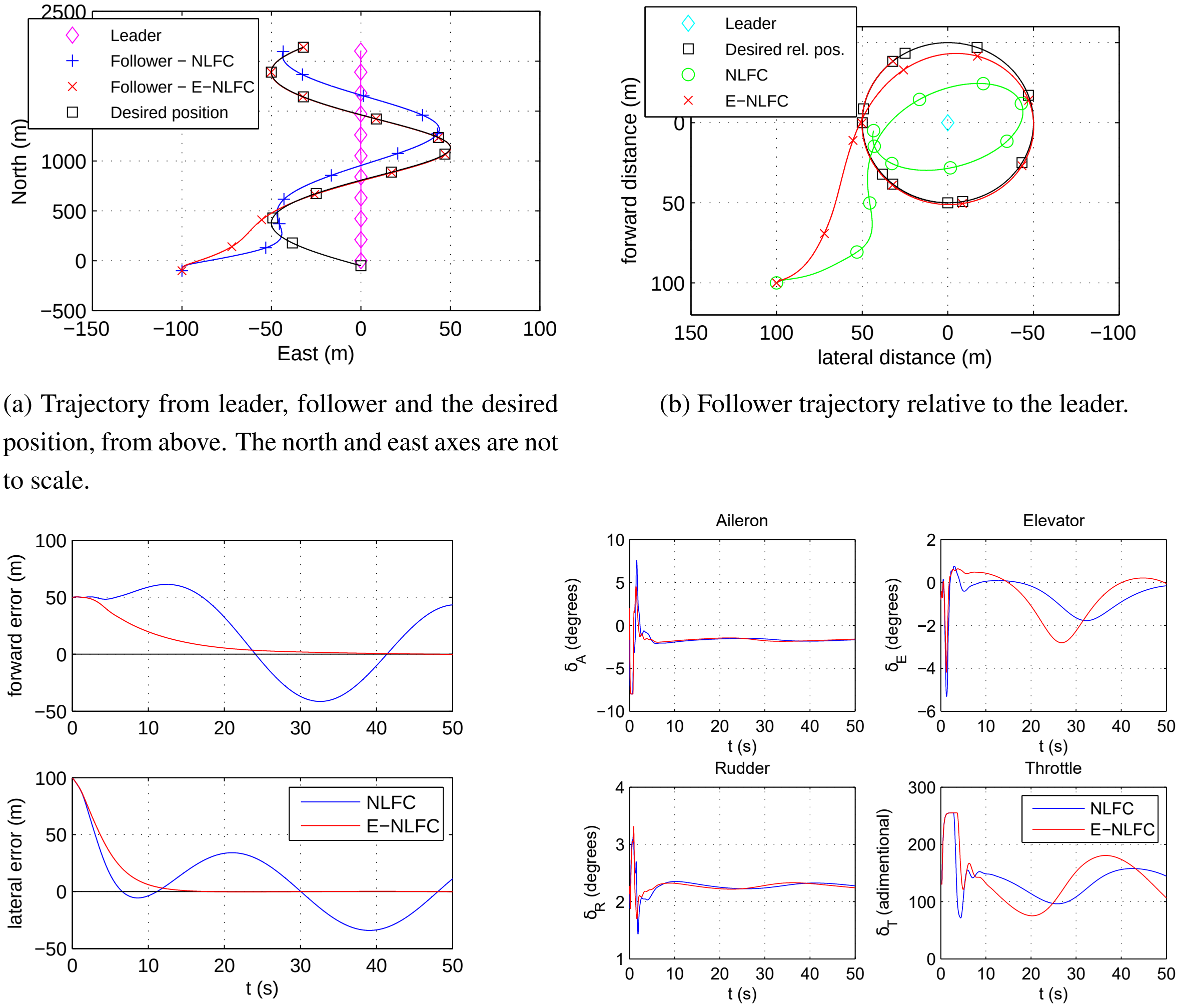
Figure 3.5a shows the trajectory from above, Fig. 3.5b shows the relative position between follower and leader, and Fig. 3.5c shows the forward and lateral error. It can be seen that the proposed E-NLFC improves the AFF performance compared to NLFC, tracking the variations in formation with a very low error. Figure 3.5d shows the actuator response in this scenario.

3.8.1.3 Straight level flight, planned approach path

The scenario C shows a case in which the variable-shape formation tracking ability is used as a way to the follower achieve a pre-determined approach maneuver.

The maneuver is described as a desired position, which is generated by the path planner presented in Section 3.6. It is configured that the path planner generates a trajectory that have both the maximum forward and lateral relative accelerations as 1.2 m/s^2, and both the maximum forward and lateral relative speed as 5 m/s. It is worth to note that this restriction is applied only in the trajectory generation, and not in the aircraft itself. The aircraft flies for 5 seconds before engage in the approach maneuver. Figure 3.6 shows the planned forward desired position f_d. The lateral planned path is identical and is omitted.

Figure 3.7a shows the trajectory from above, and Fig. 3.7b shows the forward and lateral error. The proposed algorithm tracks the approach trajectory with a low error, whereas the NLFC, not projected to work in variable formation, presents errors near 35 m in the forward direction and 25 m in the backward direction. Figure 3.7c shows the actuator response. It can be seen that the abruptly changes in $\ddot{f}_c(t)$ generates peaks in $\delta_a(t)$ and $\delta_r(t)$. It is worth to note that the simulation includes actuator dynamics and limitations so, at least theoretically, the real actuators in a real scenario are capable to respond as fast as the simulated actuators which response is shown in Fig. 3.7c. Also, being these peaks only an artifact of the way that the path planner generates trajectory, and since it occurs in very short instants of time, the performance is not significantly degraded if the real actuator output does not achieve exactly the simulated actuator output.



(a) Trajectory from leader, follower and the desired position, from above. The north and east axes are not to scale.

(b) Follower trajectory relative to the leader.

(c) Forward and lateral errors.

(d) Follower actuator response.

Figure 3.5: Results from NLFC and the proposed E-NLFC controllers in scenario B.

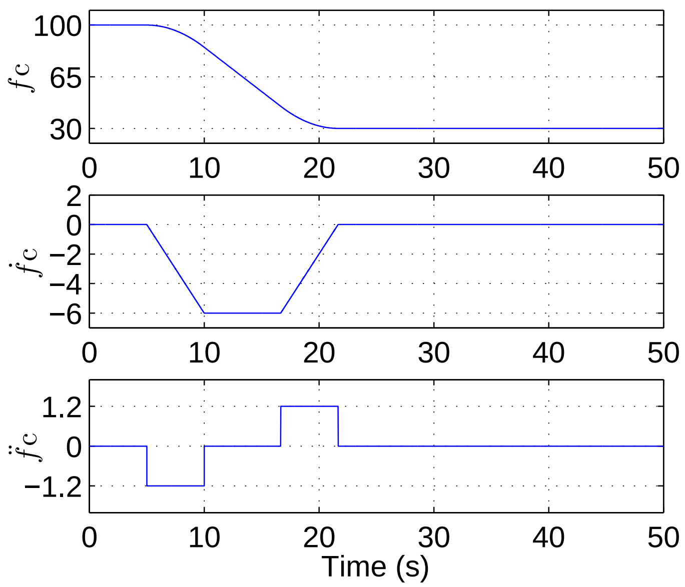


Figure 3.6: Approach trajectory $f_d(t)$ and its first and second derivative. The $l_d(t)$ is identical and is omitted.

3.8.1.4 Maneuvering, variable shape formation

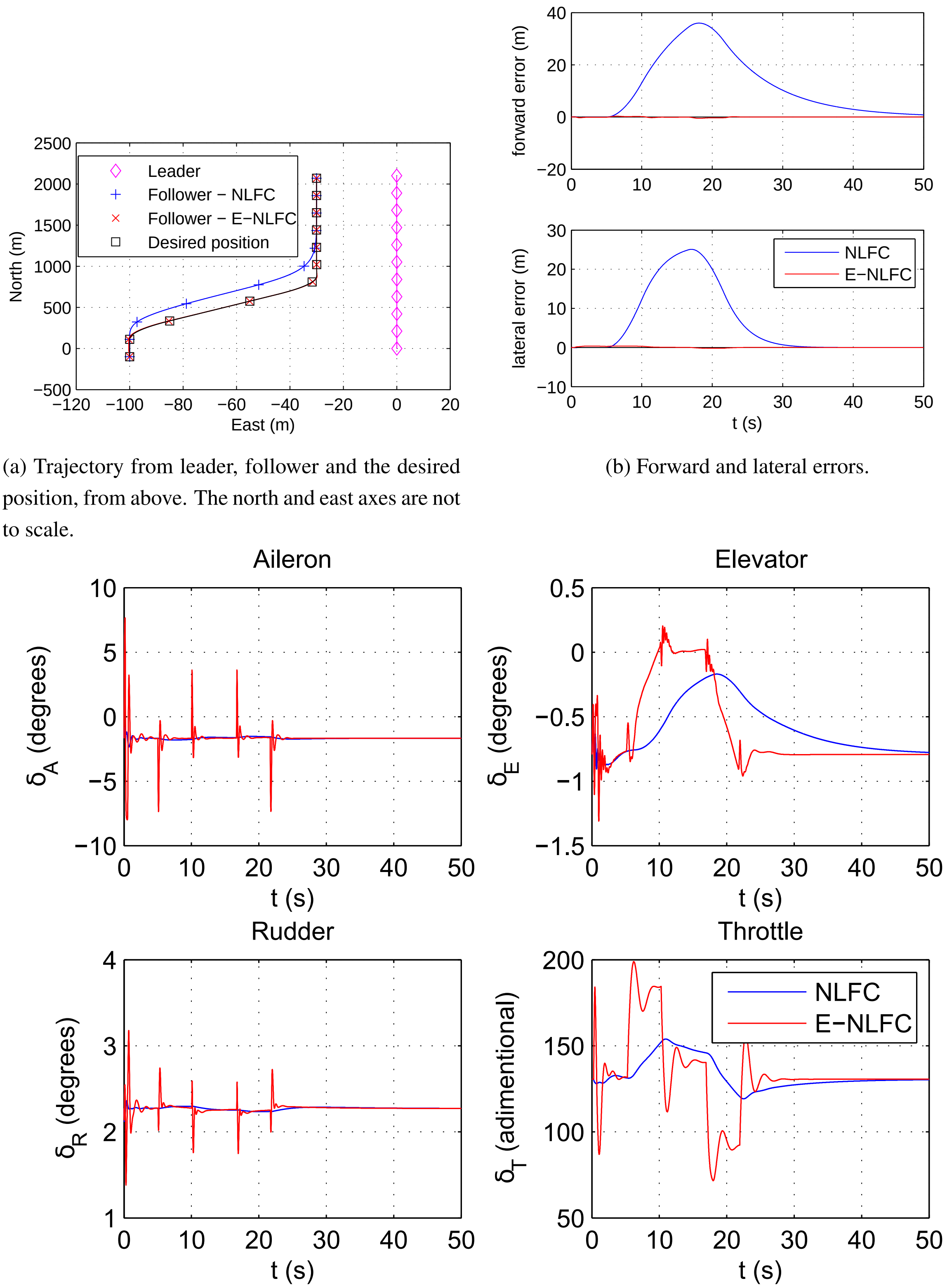
Another situation in which the NLFC algorithm is not projected for, is a high maneuvering leader combined with a distant desired position. In the scenario D, the initial conditions are the same from previous scenarios. However, the leader rate of turn $\dot{\chi}_0(t)$, speed $V_0(t)$ and altitude $h_0(t)$ are variable. $\dot{\chi}_0$ is a cosine with amplitude 5 $^\circ/s$ and frequency 0.1 Hz. The speed is a 42 m/s plus a sine of amplitude 5 m/s and frequency 1/50 Hz. h_0 = 120 + 5t m. The desired follower position is described the the planned approach path from Fig. 3.6.

Figure 3.8a shows the 3D trajectory from leader and follower, and Fig. 3.8b shows the forward, lateral and vertical errors. It can be seen that the E-NLFC converges faster to a lower error than NLFC. The actuators response are shown in Fig. 3.8c.

3.8.1.5 Collision avoidance

This scenario evaluates the proposed collision avoidance algorithm. In this scenario, the leader behaves identically to the straight level scenario. There are now two followers, both have the same parameters as the leader, except its initial position. The first follower is in $\Delta\boldsymbol{p}_{\chi0,1}$ = [100 50 0]$^{\mathrm{T}}$, and its desired position $\delta\boldsymbol{p}^d_{\chi0,1}$ is [30 -30 0]$^{\mathrm{T}}$ m. The second follower is in $\Delta\boldsymbol{p}_{\chi0,2}$ = [100 -50 0]$^{\mathrm{T}}$, and its desired position $\delta\boldsymbol{p}^d_{\chi0,2}$ is [30 30 0]$^{\mathrm{T}}$ m. Since one follower must change from left to the right of the leader, and the other must do the opposite, a collision may occur.

The initial planned trajectories are shown in Fig. 3.9. It can be seen that the paths intersect with each other, which means that the followers pass through the same point. Distances show that, in this initial planned trajectory, the aircraft achieves the same point at same time, colliding. So, another trajectory must

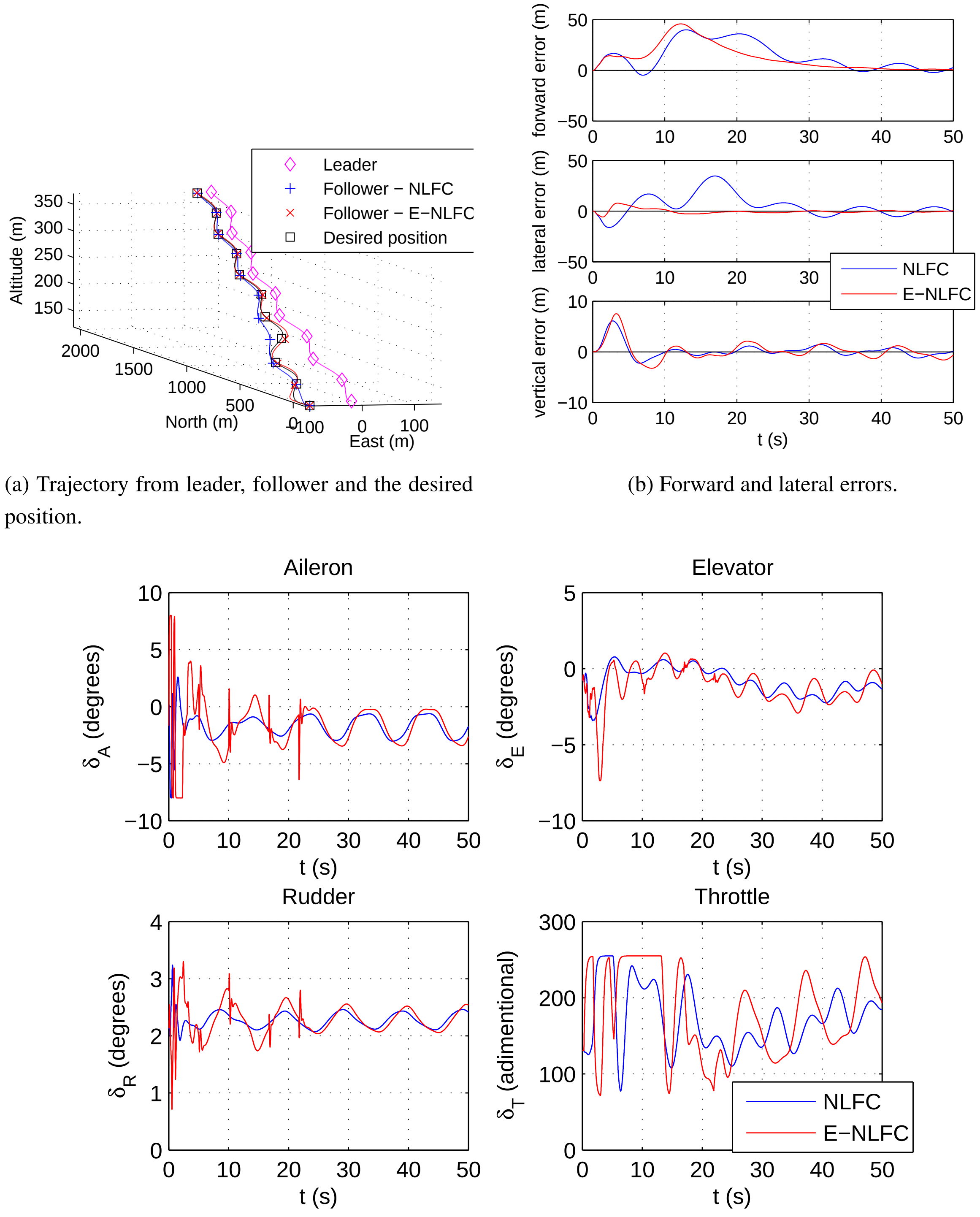


(a) Trajectory from leader, follower and the desired position, from above. The north and east axes are not to scale.

(b) Forward and lateral errors.

(c) Follower actuator response.

Figure 3.7: Results from NLFC and the proposed E-NLFC controllers in scenario C.



(a) Trajectory from leader, follower and the desired position.

(b) Forward and lateral errors.

(c) Follower actuator response.

Figure 3.8: Results from NLFC and the proposed E-NLFC controllers in scenario D.

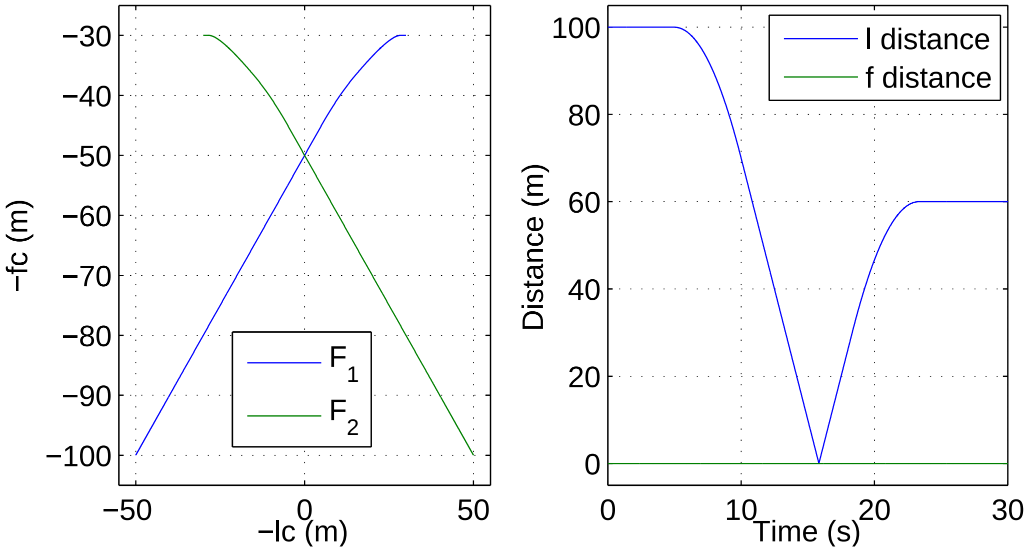


Figure 3.9: Initial planned trajectory for followers 1 (F_1) and 2 (F_2), shown in the S_χ frame, and distance between followers on forward and lateral directions, with collision. Signals reversed in trajectory to make a forward movement to go up in the plot.

be made to one of the followers.

Also, Fig. 3.9 shows that both UAVs are always with the same forward distance to the leader, resulting in null forward distance between then. Because of this, delaying the point in which the lateral distance is null does not help. It is necessary to change the forward movement to include the minimum safety margin. The second aircraft is arbitrary chosen to delay the start of its forward movement in $t_{\text{delay}} = v_{f,\max}/10$ which puts the first aircraft 10 m ahead of the second one during the constant speed approximation.

Figure 3.10 shows that the aircraft does not collide, since the forward distance is 10 m when the lateral distance is null. Figure 3.11 shows the simulation of the two followers correctly tracking the planned non-colliding track.

3.8.2 NED-NLDI Formation control approach

The S_{NED}-NLDI variant outer loop uses the $\boldsymbol{k}_{di} = \boldsymbol{k}_{pi} = \boldsymbol{I}_3$ as tuning. It is worth to note that the gains define the behavior in north, east and down directions, diferently to the $S_{\chi 0}$-NLDI variants, in which the gains are defined relatively to the forward and lateral direction. In this way, the gains can not be tuned to the engine and aileron dynamics. Movements in north direction, for example, can be achieved by the engine, aileron, or a combination of both, depending on the aircraft course direction. The evaluated gain puts the poles in a compromise between the poles presented in Table 3.1.

The same scenario B from Subsection 3.8.1.2 is simulated, but for the S_{NED}-NLDI variant controller. The only difference, besides the controller, is the use of a virtual leader instead of an leader aircraft.

Figure 3.12a shows the trajectory from above, Fig. 3.12b shows the forward and lateral error and Fig. 3.12c shows the actuator response. The proposed algorithm tracks the proposed trajectory. The forward distance has degraded performance, since the engine command presents saturation. This is caused because

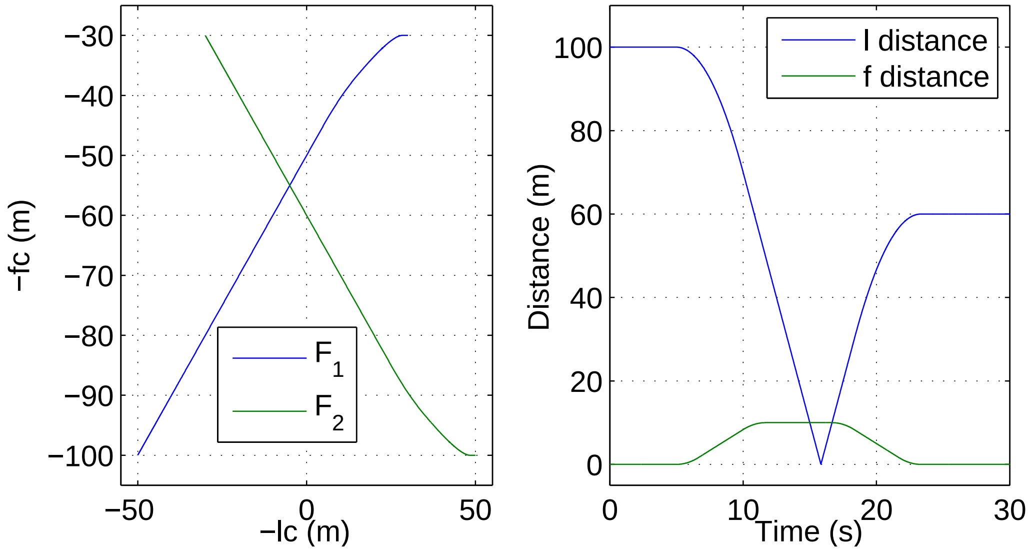


Figure 3.10: Non-colliding planned trajectory for followers 1 (F_1) and 2 (F_2), shown in the S_χ frame and distance between followers on forward and lateral directions.

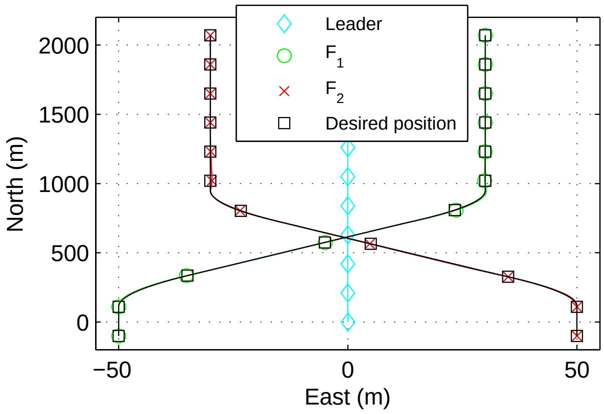


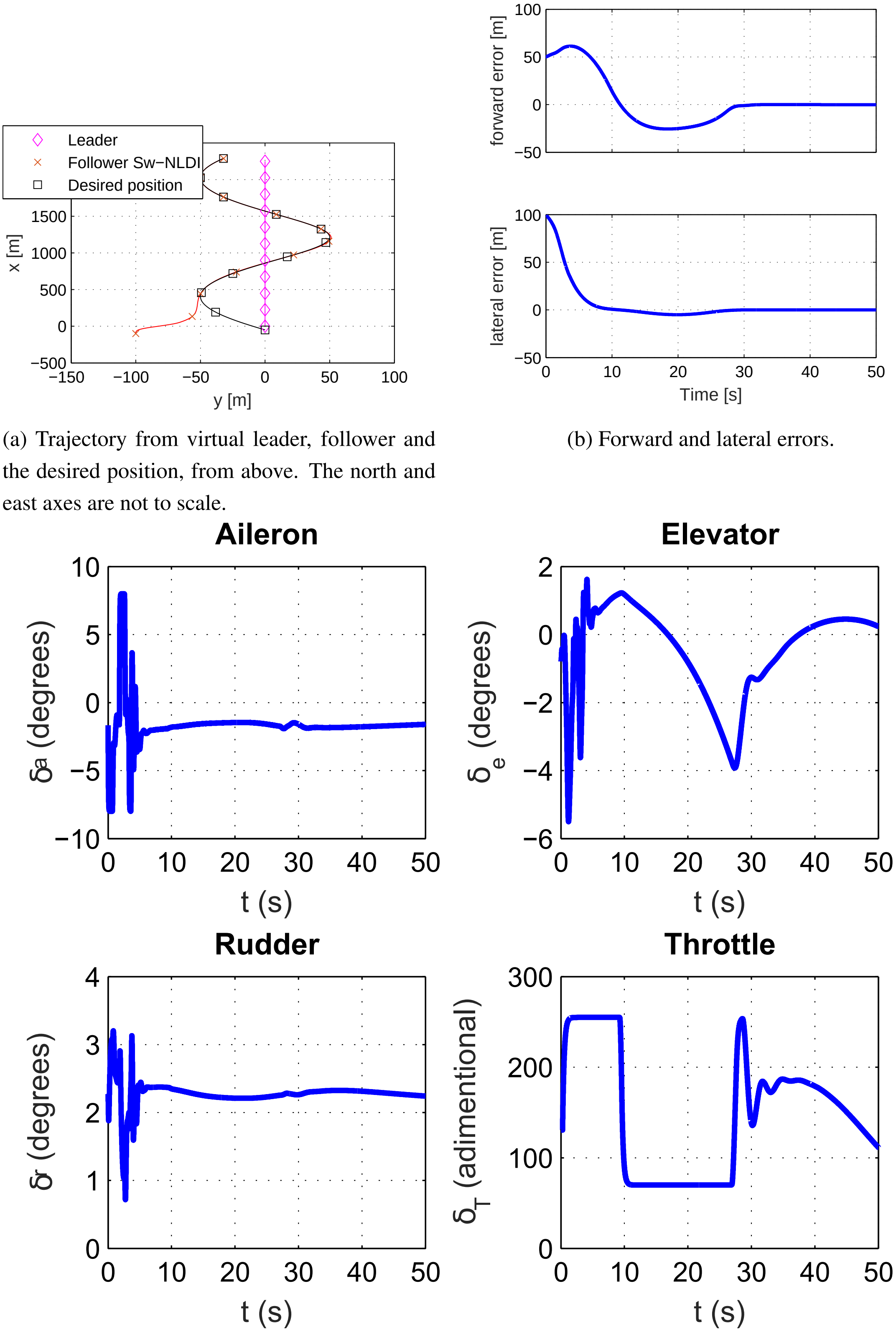
Figure 3.11: Trajectory of leader and two followers, both using the proposed improved controller, tracking a planned non-colliding track.

the projected pole of the linearized system is too fast. To avoid the saturation, a slower pole can be chosen, but this degrades the speed of convergence of the lateral error.

3.9 CONCLUSIONS

Here are presented two outer loop nonlinear dynamic inversion formation flight controller styles, named here as S_{NED} and the $S_{\chi 0}$ variants. Both variants work as expected, generating three axial double integrators dynamics. Both styles assume an inner loop that is faster than the outer loop, so the poles from the outer loop must be allocated appropriately.

In the $S_{\chi 0}$ variant, the proposed E-NLFC presented increased performance compared to the NLFC presented in the literature. It provides a low tracking error in a broader set of scenarios, including maneu-



(a) Trajectory from virtual leader, follower and the desired position, from above. The north and east axes are not to scale.

(b) Forward and lateral errors.

(c) Follower actuator response.

Figure 3.12: Results from the S_{NED}-NLDI AFF controller variant in scenario C.

vering leader and time-varying shape. The S_{NED} present in literature is already correct, and present similar performance to the proposed $S_{\chi 0}$ variant. The main advantage of the proposed variant is that it is easier to develop path planners. Also, it provides a way to tune separately the forward and lateral movements, providing a finer tuning with extra performance.

4 SLIDING MODE CONTROL

4.1 INTRODUCTION

In Chapter 3 two NLDI AFF approaches were presented, obtaining satisfactory results under the hypothesis that the UAV model is perfectly known and that there are no disturbances, such as wind. In practice, these disturbances and uncertainties, if unaccounted, can degrade or even destabilize the controller. The uncertainties and disturbances can be treated by using a robust control approach. Another approach to achieve extra robustness include cooperation between the followers agents. Both modifications in the control law are discussed here.

The synchronous [24, 15, 14] and distributed [25, 26, 22] autonomous formation control are generalizations of the leader-follower and virtual leader approaches, in which each agent not only aims to reduce its error relatively to the (virtual) leader, but also aims to reduce the error difference between itself and its set of neighbors. In the synchronous approach, which assumes that all agents have access to the leader's data, these tracking errors synchronization allows the controller designer to weight between two control objectives: 1) each agent individually achieves its own desired position in the formation, without cooperation and 2) the agents move collectively to achieve the desired relative position between agents, i.e., the desired formation shape, even if this means that some agents will temporarily distance itself from their own desired individual position. In the distributed approach, some of the agents does not have access to the leader's data. In this case, reducing the relative error between neighbors achieves a reduction in the the relative error between the agents and the (virtual) leader, even to the agents that does not directly communicates with the leader.

As explained in Chapter 1, the model obtained from NLDI approach is, in an ideal scenario, linear. This maintains true even when the leader-follower model is modified to a more general model such as synchronous or distributed, as explained in this Chapter. The NLDI, however, does not provide a perfectly linear model if there are model uncertainties. The sliding mode controller (SMC) can be seen as a modification in the NLDI to achieve robustness to model uncertainties and bounded disturbances. The controller, however, produces a chattering control output, which is undesired. A solution to chattering, evaluated in this chapter, is to use the low-pass-filter SMC variant.

Recently, one low-pass-filter SMC architecture was proposed specifically for synchronous formation of UAVs [14]. It achieves a robust controller, without presenting actuator chattering. Also, the formation can be time-varying, and the formation can be described in any frame, such as in inertial or in the leader's frame. It is the robust version the S_{NED}-NLDI presented in Section 3.4. It is, allegedly, a decentralized approach, which means that each UAV can implement locally its controller by using only information about itself, from its neighbors via the communication channel and/or from the virtual leader via computation of the embedded virtual leader's model.

In this chapter, as contribution, it is shown that the controller presented in [14] does not achieve the decentralization alleged by the article's author. The reason is that the local controller implemented in each follower UAV requires access to data from outside its neighborhood, which means that the controller does not respect a given communication topology. In the sequel, a decentralized synchronous AFF SMC that correct these problems is proposed. A second variant is after proposed, which includes an improvement in the low pass filter chattering reduction technique, which reduces the mathematical and computational complexity of the controller and improves the controller performance. The proposed second variant is an adaptation of single-agent SMC presented in [41] to the synchronous multi-agent scenario.

The finite-time convergence to a linear sliding surface is proven for both proposed approaches by introduction of appropriated Lyapunov function candidates. Simulation results show the effectiveness of the proposed control architecture.

In this chapter, only the S_{NED}-NLDI variant is evaluated, since it is the usual approach in literature [15, 14, 26, 38]. However, since both the S_{NED} and $S_{\chi 0}$ NLDI variants reduce the system to double integrators, the developed robust controllers can be adapted to the $S_{\chi 0}$-NLDI variant without significant modification.

The remainder of this chapter is organized as follows. Section 4.2 briefly describes the SMC and presents the mathematical models for formation flight and communication graph. Section 4.3 presents the controller found in literature, and demonstrates that it has some problems in its design. Section 4.4 presents a proposed controller that correct these problems and proves its stability. Section 4.5 presents a new proposed controller that include an improvement over the proposed controller from the previous section. 4.6 evaluates the proposed controller by simulation and Section 4.7 concludes this chapter.

4.2 PRELIMINARIES

In this section, the sliding mode controller, the disturbance model and fleet communication model are presented.

4.2.1 Sliding Mode Control

Here the SMC is briefly reviewed. The review is based in [42, 60].

As in the NLDI, the tracking error is defined as $\boldsymbol{e}(t) = \boldsymbol{\xi}(t) - \boldsymbol{\xi}_d(t)$, where $\boldsymbol{\xi}(t)$ is the state vector of a nonlinear system represented in the companion form (3.1) and $\boldsymbol{e}(t) = [e(t)\ \dot{e}(t)\ ...\ e^{(n-1)}(t)]^T$.

Aiming to reduce the n-th tracking problem of the vector $\boldsymbol{e}(t)$ to a first order tracking problem, the scalar function $s(t)$ is defined as

$$s(t) = e^{(n-1)}(t) + k_{n-2}e^{(n-2)}(t) + ... + k_1\dot{e}(t) + k_0 e(t). \tag{4.1}$$

It can be seen that if $s(t)$ is maintained zero by choosing the appropriated control signal $u(t)$, it is obtained

$$0 = e^{(n-1)}(t) + k_{n-2}e^{(n-2)}(t) + ... + k_1\dot{e}(t) + k_0 e(t), \tag{4.2}$$

and the tracking error remains in a subset of the state space known as surface s. When $s(t) = 0$, it is said that the system is sliding in the surface. Being the surface a linear system, it is exponentially stable and converges to $e(t) = 0$ if all poles are strictly in the left half complex plane. The sliding surface dynamics must be adequate to the system dynamics, respecting for example its control bandwidth.

It can be shown that if a control law is chosen such as

$$\frac{1}{2}\frac{d}{dt}s^2(t) \leq -\epsilon|s(t)|, \tag{4.3}$$

where $\epsilon > 0$ is constant, the function $s(t)$ converges to zero in finite time t_{reach} given by

$$t_{\text{reach}} \leq |s(t=0)|/\epsilon. \tag{4.4}$$

4.2.1.1 Model uncertainty and disturbance rejection

When modeling a system using the companion form (3.1), the functions $f(\boldsymbol{\xi}(t))$ and $b(\boldsymbol{\xi}(t))$ are not perfectly known. In the modeling, estimates $\hat{f}(\boldsymbol{\xi}(t))$ and $\hat{b}(\boldsymbol{\xi}(t))$ are used instead. There can be also an unmeasured disturbance $u_d(t)$. It is assumed, however, that there are known bounds in the magnitude of the error created by using these estimated functions and in the magnitude of the disturbance.

$$|\hat{f}(\boldsymbol{\xi}(t)) - f(\boldsymbol{\xi}(t))| \leq F, \tag{4.5a}$$

$$\beta^{-1} \leq \frac{\hat{b}(\boldsymbol{\xi}(t))}{b(\boldsymbol{\xi}(t))} \leq \beta, \tag{4.5b}$$

$$|u_d(t)| \leq U_d. \tag{4.5c}$$

The offset error caused by using $\hat{f}(\boldsymbol{\xi}(t))$ instead of $f(\boldsymbol{\xi}(t))$, the scaling factor error that multiplies the control input that is caused by using $\hat{b}(\boldsymbol{\xi}(t))$ instead of $b(\boldsymbol{\xi}(t))$, and can be grouped in a single unknown bounded disturbance $d(t)$, where

$$|d(t)| \leq \Delta, \tag{4.6}$$

resulting in

$$\xi^{(n)}(t) = f(\boldsymbol{\xi}(t)) + b(\boldsymbol{\xi}(t))u(t) + u_d(t) = \hat{f}(\boldsymbol{\xi}(t)) + \hat{b}(\boldsymbol{\xi}(t))u(t) + d(t). \tag{4.7}$$

By deriving (4.1)

$$\begin{aligned} \dot{s}(t) &= e^{(n)}(t) + k_{n-2}e^{(n-1)}(t) + ... + k_1\ddot{e}(t) + k_0\dot{e}(t) \\ &= \xi^{(n)}(t) - \xi_d^{(n)}(t) + k_{n-2}e^{(n-1)}(t) + ... + k_1\ddot{e}(t) + k_0\dot{e}(t) \\ &= \hat{f}(\boldsymbol{\xi}(t)) + \hat{b}(\boldsymbol{\xi}(t))u(t) + d(t) - \xi_d^{(n)}(t) + k_{n-2}e^{(n-2)}(t) + ... + k_1\ddot{e}(t) + k_0\dot{e}(t). \end{aligned} \tag{4.8}$$

Defining $u(t)$ as

$$u(t) = \hat{b}^{-1}(\boldsymbol{\xi}(t))\left(u_a(t) + u_b(t)\right), \tag{4.9a}$$

$$u_a(t) = -\hat{f}(\boldsymbol{\xi})(t) + \xi_d^{(n)}(t) - k_{n-2}e^{(n-1)}(t) - ... - k_1\ddot{e}(t) - k_0\dot{e}(t), \tag{4.9b}$$

$$u_b(t) = -\eta\ \text{sgn}(s(t)), \tag{4.9c}$$

it can be seen that $u_a(t)$ would achieve $\dot{s}(t) = 0$ if $d(t) = 0$. To account the uncertainties and disturbances, a discontinuous term is added via the control input component $u_b(t)$

Using the proposed $u(t)$ from (4.9) and the upper bound from (4.6)

$$\begin{aligned} \frac{1}{2}\frac{d}{dt}s^2(t) &= \dot{s}(t)s(t) \\ &= [d(t) - \eta \operatorname{sgn}(s(t))]\, s(t) \\ &= d(t)s(t) - \eta\, |s(t)| \\ &\leq [d(t) - \eta]\ \ |s(t)| \\ &\leq [\Delta - \eta]\ \ |s(t)|. \end{aligned} \tag{4.10}$$

By choosing $\eta = \Delta + \epsilon$, it is obtained (4.3), which means that the system achieves the sliding surface in finite time and, after, the error asymptotically converges to zero with linear dynamics given by (4.2).

The SMC can be understand as: if there is error, apply a sufficiently high input in opposite direction, which is an approach that works in first order error models, such as defined by $s(t)$, since the control input appears in $\dot{s}(t)$. From this, it can be naively concluded that η can be chosen as a very high arbitrary value, and that greater values provides better results. However, the greater is η, the greater is the input value, and the greater is the discontinuity when $s(t)$ changes sign. In other words, if η is much greater than needed a high intensity chattering effect will appear. Chattering is the actuator abruptly changing its output in a high frequency. Therefore, it is good design to choose η not greater than needed. It is important to note that the greater the uncertainties, the greater must be η.

4.2.1.2 Avoiding chattering

Even a good choice of η will present some chattering, especially if the uncertainties are high. If the chattering effect is undesired, and some error is acceptable, the discontinuous $\operatorname{sgn}(s(t))$ function can be changed to the continuous one $\operatorname{sat}(s(t)/\Phi)$, where Φ is a scale factor parameter and

$$\operatorname{sat}(x) = \begin{cases} x, & \text{if } |x| \leq 1, \\ 1, & \text{otherwise.} \end{cases} \tag{4.11}$$

In this case, [42] shows that the controller maintains $s(t) \leq \Phi$. However, $s(t)$ does not necessary converge to zero, which results in the error being limited, but not converging to zero. Examples of formation controller that follow this approach are [25, 40]. A variant of this approach is used in [27], in which a hyperbolic tangent function is used instead of the sign function.

Another approach is to understand that the chattering comes from the way that the disturbance is modeled. The magnitude of the disturbance is limited, but not its derivative. In this way, if it is assumed that $d(t)$ can change from $-\Delta$ to $+\Delta$ instantaneously, this means that the controller must change instantaneously from $-\eta$ to $+\eta$, $\eta > \Delta$, if the disturbance must be instantaneously compensated.

The n-th order SMC is a group of variants of the original SMC that assumes that are bounds up to the $(N-1)$-th derivative of the disturbance [60]. Under this assumption, the obtained control signal has up to the $(N-1)$-th derivative bounded, which results in a continuous control signal.

The low-pass-filter (LPF) SMC, proposed in [43], and used for example in [24, 41, 44, 14, 45] is a second-order SMC. It inserts a low-pass-filter between the control signal and the plant to be controlled, as depict in Fig. 4.1. The low-pass-filter plus plant is seen as an augmented plant. The SMC is then projected as usual, but considering the model of the augmented plant. Since the model of the LPF is included in the project of the SMC, the resulting controller is still robust, even being the filtered control signal continuous. The LPF SMC applied in a MIMO system is further explained in Section 4.4, where a synchronous formation flight controller is proposed.

4.2.2 Disturbance model

Measurement or calculation errors and the effect of non modeled dynamics are incorporated in the dynamics model, given by Eq. (3.17), as a disturbance signal described in the reference frame, $\boldsymbol{d}_i = [\,d_{xi} \quad d_{yi} \quad d_{zi}\,]^T$. It is supposed that the controller has no access to $\boldsymbol{d}_i$ but there are known upper bounds Δ_{xi}, Δ_{yi} and Δ_{zi} on the magnitude of the components of $\boldsymbol{d}_i$ and upper bounds Δ_{dxi}, Δ_{dyi}, and Δ_{dzi} on the derivatives of the components of $\boldsymbol{d}_i$, i.e.

$$|d_{xi}(t)| \leq \Delta_{xi}, \quad |d_{yi}(t)| \leq \Delta_{yi}, \quad |d_{zi}(t)| \leq \Delta_{zi}, \tag{4.12}$$

$$|\dot{d}_{xi}(t)| \leq \Delta_{dxi}, \quad |\dot{d}_{yi}(t)| \leq \Delta_{dyi}, \quad |\dot{d}_{zi}(t)| \leq \Delta_{dzi}. \tag{4.13}$$

The upper bounds on the components in the reference frame coordinates can be calculated from the upper bounds δ_{ti}, δ_{pi} and δ_i on the components of the disturbance signal in the wind frame $\boldsymbol{b}_i(t)$ from (2.46),

$$|b_{ti}(t)| \leq \delta_{ti}, \quad |b_{pi}(t)| \leq \delta_{pi}, \quad |b_{yi}(t)| \leq \delta_{yi}, \tag{4.14}$$

and from the upper bounds δ_{dti}, δ_{dpi} and δ_{dyi} for the

$$|\dot{b}_{ti}(t)| \leq \delta_{dti}, \quad |\dot{b}_{pi}(t)| \leq \delta_{dpi}, \quad |\dot{b}_{yi}(t)| \leq \delta_{dyi}. \tag{4.15}$$

The wind frame components of the disturbances are more naturally obtained, for example, in description of imprecision in the calculation of drag or thrust forces. Assume that there is an upper bound $\boldsymbol{\Omega}_i$ for the i-th UAV angular velocity $\boldsymbol{\omega}_{swi}^{swi,\mathrm{NED}}(t)$ and define the bounds vectors $\boldsymbol{\delta}_i \triangleq [\,\delta_{ti} \quad \delta_{pi} \quad \delta_{yi}\,]^T$ and $\boldsymbol{\delta}_{di} \triangleq [\,\delta_{dti} \quad \delta_{dpi} \quad \delta_{dyi}\,]^T$. From Eq. (3.16), it can be seen that

$$|d_{xi}(t)| \leq \|\boldsymbol{d}_i(t)\|, \tag{4.16a}$$

$$\|\boldsymbol{d}_i(t)\| = \|\boldsymbol{D}_{\mathrm{NED}}^{swi}(t)\boldsymbol{b}_i(t)\|, \tag{4.16b}$$

$$\|\boldsymbol{D}_{\mathrm{NED}}^{swi}(t)\boldsymbol{b}_i(t)\| = \|\boldsymbol{b}_i(t)\|, \tag{4.16c}$$

$$\|\boldsymbol{b}_i(t)\| \leq \|\boldsymbol{\delta}_i\|, \tag{4.16d}$$

where (4.16a) means that the magnitude of a component of the disturbance vector is limited by the norm of the disturbance vector, (4.16b) means that since the disturbance vector in the S_{NED} frame is vector in the S_{swi} frame rotated by the rotational matrix $\boldsymbol{D}_{\mathrm{NED}}^{swi}(t)$, the norm of $\boldsymbol{d}_i(t)$ is the norm of the rotated vector, (4.16c) means that a rotation transformation does not change the vector magnitude or, alternatively, that $\|\boldsymbol{D}_{\text{NED}}^{swi}(t)\| = 1$, which means that the norm of the rotated and the non-rotated vector is the same, and (4.16d) is obntained from (4.14).

From (4.16), the upper bounds of each component of $\boldsymbol{d}_i$ are

$$\Delta_{xi} = \Delta_{yi} = \Delta_{zi} = \|\boldsymbol{\delta}_i\|. \tag{4.17}$$

The derivative of Eq. (3.16) is obtained by using the Theorem of Coriolis (2.12)

$$\dot{\boldsymbol{d}}_i(t) = \boldsymbol{D}_{\text{NED}}^{swi}(t)\left(\dot{\boldsymbol{b}}_i(t) + \boldsymbol{\omega}_{swi}^{swi,\text{NED}}(t) \times \boldsymbol{b}_i(t)\right). \tag{4.18}$$

With respect to the bounds $\boldsymbol{\delta}_i$, $\boldsymbol{\delta}_{di}$ and $\boldsymbol{\Omega}_i$, it is obtained

$$\begin{aligned}
|\dot{d}_{xi}(t)| &\le \|\dot{\boldsymbol{d}}_i(t)\|, && \text{(4.19a)}\\
\|\dot{\boldsymbol{d}}_i(t)\| &= \left\|\boldsymbol{D}_{\text{NED}}^{swi}(t)\left(\dot{\boldsymbol{b}}_i(t) + \boldsymbol{\omega}_{swi}^{swi,\text{NED}}(t) \times \boldsymbol{b}_i(t)\right)\right\|, && \text{(4.19b)}\\
&= \|\dot{\boldsymbol{b}}_i(t) + \boldsymbol{\omega}_{swi}^{swi,\text{NED}}(t) \times \boldsymbol{b}_i(t)\|, && \text{(4.19c)}\\
&\le \|\dot{\boldsymbol{b}}_i(t)\| + \|\boldsymbol{\omega}_{swi}^{swi,\text{NED}}(t) \times \boldsymbol{b}_i(t)\|, && \text{(4.19d)}\\
\|\dot{\boldsymbol{b}}_i(t)\| + \|\boldsymbol{\omega}_{swi}^{swi,\text{NED}}(t) \times \boldsymbol{b}_i(t)\| &\le \|\dot{\boldsymbol{b}}_i(t)\| + \|\boldsymbol{\omega}_{swi}^{swi,\text{NED}}(t)\|\,\|\boldsymbol{b}_i(t)\|, && \text{(4.19e)}\\
\|\dot{\boldsymbol{b}}_i(t)\| + \|\boldsymbol{\omega}_{swi}^{swi,\text{NED}}(t)\|\,\|\boldsymbol{b}_i(t)\| &\le \|\boldsymbol{\delta}_{di}\| + \|\boldsymbol{\Omega}_i\|\|\boldsymbol{\delta}_i\|, && \text{(4.19f)}
\end{aligned}$$

where (4.19d) is obtained from the triangle inequality and (4.19e) obtained from the computation of the magnitude of the cross product $\|\boldsymbol{a} \times \boldsymbol{b}\| = \|\boldsymbol{a}\|\|\boldsymbol{b}\| \sin\theta$, assuming the highest magnitude possible, which is achieved if the vectors are orthonormal, meaning that $\theta = 90^o$. In this way,

$$\Delta_{dxi} = \Delta_{dyi} = \Delta_{dzi} = \|\boldsymbol{\delta}_{di}\| + \|\boldsymbol{\Omega}_i\|\|\boldsymbol{\delta}_i\|. \tag{4.20}$$

4.2.3 Formation tracking and synchronization errors

The tracking error of each aircraft $\boldsymbol{e}_i(t)$ is defined by (3.19). The synchronization error $\Delta\boldsymbol{e}_{ij}(t) = [\,\Delta e_{xij}(t) \quad \Delta e_{yij}(t) \quad \Delta e_{zij}(t)\,]^T \in \mathbb{R}^3$, which can be seen as a relative position error between the UAVs, is defined as

$$\Delta\boldsymbol{e}_{ij}(t) \triangleq \boldsymbol{e}_i(t) - \boldsymbol{e}_j(t) = \boldsymbol{p}_i(t) - \boldsymbol{p}_j(t) - \left(\tilde{\boldsymbol{p}}_i(t) - \tilde{\boldsymbol{p}}_j(t)\right). \tag{4.21}$$

It can be seen that $\Delta\boldsymbol{e}_{ij}(t)$ can be calculated without knowing the leader's position. However, since the computation of $\tilde{\boldsymbol{p}}_i(t)$ and $\tilde{\boldsymbol{p}}_j(t)$ in Eq. (3.23) can be chosen to be dependent of the leader's flight direction or attitude angles, it is assumed here that the leader's data is available to all UAVs. Also, it can be seen that $\Delta\boldsymbol{e}_{ij}(t)$ can be obtained by knowing the relative position between aircraft i and j, which is useful if some follower UAV does not have a GPS sensor and does not know its own position $\boldsymbol{p}_i(t)$.

It is assumed that each i-th UAV can communicate only with a correspondent set of neighbor UAVs, $\mathcal{N}_i \subset \{1, 2, \ldots, n\}$. The communication graph is assumed to be undirected, connected, not change with time, and previously known. Each UAV receives the tracking error information of other UAVs in the fleet only through its neighbors (for example, see Fig. 4.4). The virtual leader can be seen as an extra node in the graph, that connects to every other UAV in a directed way, from leader to each follower.

The coupled error at i-th UAV is defined as the weighted sum of its tracking error and the synchronization error with respect to its neighbors, that is,

$$\boldsymbol{e}_i^c(t) = [\, e_{xi}^c(t) \quad e_{yi}^c(t) \quad e_{zi}^c(t)\,]^T \triangleq \lambda_i \boldsymbol{e}_i(t) + \sum_{j\in\mathcal{N}_i} a_{ij}\Delta\boldsymbol{e}_{ij}(t) = \lambda_i \boldsymbol{e}_i(t) + \sum_{j=1}^{n} a_{ij}\Delta\boldsymbol{e}_{ij}(t), \tag{4.22}$$

in which $\lambda_i > 0$ weights its own tracking error and $a_{ij} > 0$ weights the error difference between the neighbor UAV j of the UAV i. In the last equality in Eq. (4.22), if $j \notin \mathcal{N}_i$ then $a_{ij} = 0$. The synchronization control objective is to make the coupled errors approach to zero.

4.2.4 A componentwise formation description

Following [14], it is supposed that each component of $\boldsymbol{d}_i(t)$ is independent from each other which implies that each component of $\ddot{\boldsymbol{p}}_i(t)$ is independent from each other. In this way, the controller design is simplified since the description of only one axis is sufficient. A controller policy can be developed to a single axis and then it can be directly applied to the other two.

The one-dimensional dynamics from the x axis of reference frame is obtained from Eq. (3.17) as

$$\ddot{x}_i(t) = \tau_{xi}(t) + d_{xi}(t). \tag{4.23}$$

Accordingly, the coupled tracking-synchronization error is obtained from Eq. (4.22) as

$$e_{xi}^c(t) = \lambda_i e_{xi}(t) + \sum_{j\in\mathcal{N}_i} a_{ij}\left[e_{xi}(t) - e_{xj}(t)\right]. \tag{4.24}$$

4.2.5 Fleet dynamics

In order to analyze the overall fleet behavior, all local variables must be concatenated in vectors. Concatenating the positions x_i, virtual control inputs $\tau_{xi}(t)$, and disturbances $d_{xi}(t)$ from all UAVs of the fleet results in respectively $\boldsymbol{P}_x(t) = [\, x_1(t) \ \dots \ x_n(t)\,]^T$, $\boldsymbol{\tau}_x(t) = [\, \tau_{x1}(t) \ \dots \ \tau_{xn}(t)\,]^T$, and $\boldsymbol{D}_x(t) = [\, d_{x1}(t) \ \dots \ d_{xn}(t)\,]^T$, all $\mathbb{R}^n$ vectors. In this way, the dynamics of the fleet of UAVs in the x axis is given by concatenating Eq. (4.23) as

$$\ddot{\boldsymbol{P}}_x(t) = \boldsymbol{\tau}_x(t) + \boldsymbol{D}_x(t). \tag{4.25}$$

Similarly, the error and coupled error in x axis are $\mathbb{R}^n$ vectors given by $\boldsymbol{E}_x(t) = [\, e_{x1}(t) \ \dots \ e_{xn}(t)\,]^T$ and $\boldsymbol{E}_x^c(t) = [\, e_{x1}^c(t) \ \dots \ e_{xn}^c(t)\,]^T$ which are related by

$$\boldsymbol{E}_x^c(t) = \boldsymbol{H}\boldsymbol{E}_x(t), \tag{4.26}$$

where

$$\boldsymbol{H} = \boldsymbol{\Lambda} + \boldsymbol{L}, \tag{4.27}$$

$$\boldsymbol{\Lambda} = \text{diag}([\,\lambda_1\ \dots\ \lambda_n\,]), \tag{4.28}$$

and the Laplacian matrix $\boldsymbol{L}$ is obtained by

$$\boldsymbol{L}_{ij} = \begin{cases} -a_{ij}, & \text{if } j \neq i \text{ and } j \in \mathcal{N}_i, \\ \sum_{k \in \mathcal{N}_i} a_{ik}, & \text{if } j = i, \\ 0, & \text{otherwise.} \end{cases} \tag{4.29}$$

Note that since $\lambda_1, \dots, \lambda_n > 0$ and $\boldsymbol{L}$ is semidefinite positive, the matrix $\boldsymbol{H}$ is invertible.

4.3 DECENTRALIZED SLIDING MODE CONTROL FROM LITERATURE

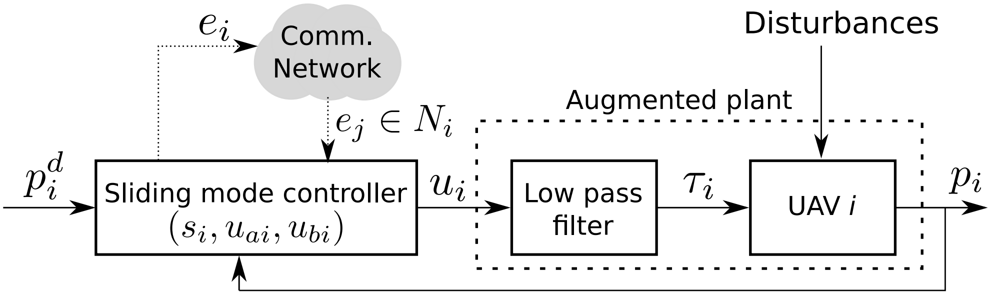


Figure 4.1: The block diagram of the control structure from [14]

In order to achieve synchronization, each UAV uses tracking errors of its neighbors to compute a sliding surface in the coupled error space. The sliding surface at the i-th UAV for the x axis is defined by [14] as

$$\bar{s}_{xi}(t) = \ddot{e}_{xi}^c(t) + k_{dxi}\dot{e}_{xi}^c(t) + k_{pxi}e_{xi}^c(t). \tag{4.30}$$

As usual to sliding mode controllers, the author shows that $s_{xi}(t)$ converges to zero in finite time, which then starts the sliding mode. When in sliding mode the coupled error behaves according to the linear system

$$\ddot{e}_{xi}^c(t) + k_{dxi}\dot{e}_{xi}^c(t) + k_{pxi}e_{xi}^c(t) = 0, \tag{4.31}$$

which is exponentially asymptotically stable for project parameters $k_{dxi}, k_{pxi} > 0$.

The controller present in [14] achieves and maintain the sliding mode defined in (4.31), and is given by

$$\boldsymbol{U}_x(t) = \boldsymbol{U}_{xa}(t) + \boldsymbol{U}_{xb}(t), \tag{4.32a}$$

$$\boldsymbol{U}_{xa}(t) = \ddot{\boldsymbol{P}}_x(t) + \boldsymbol{\Xi}_x^{-1} \left(\dddot{\boldsymbol{P}}_x^d(t) - \boldsymbol{k}'_{dx} \ddot{\boldsymbol{E}}_1^c(t) - \boldsymbol{k}'_{px} \dot{\boldsymbol{E}}_1^c(t) \right), \tag{4.32b}$$

$$\boldsymbol{U}_{xb}(t) = -\boldsymbol{\Xi}_x^{-1} \eta_x \frac{\boldsymbol{H}^T \bar{\boldsymbol{S}}_x(t) \|\bar{\boldsymbol{S}}_x^T(t)\| \|\boldsymbol{H}\|}{\|\bar{\boldsymbol{S}}_x^T(t) \boldsymbol{H}\|^2}, \tag{4.32c}$$

$$\dot{\boldsymbol{\tau}}_x(t) = -\boldsymbol{\Xi}_x \boldsymbol{\tau}_x(t) + \boldsymbol{\Xi}_x \boldsymbol{U}_x(t), \tag{4.32d}$$

where $\boldsymbol{U}_x(t) = [u_{x1}(t) \ \ldots \ u_{xn}(t)]^T \in \mathbb{R}^n$ is the control vector, achieved by stacking each of u_{xi} control signal related to the i-th UAV, $\boldsymbol{U}_{xa}(t) = [u_{xa1}(t) \ \ldots \ u_{xan}(t)]^T \in \mathbb{R}^n$ is the smooth component of the control vector signal and $\boldsymbol{U}_{xb}(t) = [u_{xb1}(t) \ \ldots \ u_{xbn}(t)]^T \in \mathbb{R}^n$ is the discontinuous (chattering) component of the control vector signal. The parameter η_{xi} must be chosen by the designer to guarantee the stability of overall system. The sliding vector $\boldsymbol{S}_x(t) = [s_{x1}(t) \ \ldots \ s_{xn}(t)]^T$ is the vector of sliding mode surfaces described by

$$\bar{\boldsymbol{S}}_x(t) = \ddot{\boldsymbol{E}}_x^c(t) + \boldsymbol{k}_{dx} \dot{\boldsymbol{E}}_x^c(t) + \boldsymbol{k}_{px} \boldsymbol{E}_x^c(t), \tag{4.33}$$

with $\boldsymbol{k}_{dx} = \text{diag}([k_{dx1} \ \ldots \ k_{dxn}])$, $\boldsymbol{k}_{px} = \text{diag}([k_{px1} \ \ldots \ k_{pxn}])$, and

$$\boldsymbol{k}'_{dx} = \boldsymbol{H}^{-1} \boldsymbol{k}_{dx}, \qquad \boldsymbol{k}'_{px} = \boldsymbol{H}^{-1} \boldsymbol{k}_{px}. \tag{4.34}$$

The diagonal matrix $\boldsymbol{\Xi}_x = \text{diag}([\xi_{x1} \ \ldots \ \xi_{xn}]) \in \mathbb{R}^{n \times n}$ defines a set of stacked first order differential equations

$$\dot{\tau}_{xi}(t) = -\xi_{xi} \tau_{xi}(t) + \xi_{xi} u_{xi}(t), \tag{4.35}$$

that define low pass filters with cutoff frequency $\xi_{xi} > 0$. This is used to convert the chattering signals $u_{xi}(t)$ to smooth signals $\tau_{xi}(t)$. It is worth to note that is the filtered control signal $\boldsymbol{\tau}_x(t)$ that is applied to the fleet dynamics in (4.25)

4.3.1 Decentralization

It is claimed by [14] that the control law (4.32a)-(4.32d) is decentralized, which means that each UAV can implement locally its controller by using only information about itself, from its neighbors via the communication channel and/or from the virtual leader via computation of the embedded virtual leader's model. In other words, this means that the control law must not require to be computed by a central entity that has acess to all available data.

However, as it is shown in the following counter-example, this is not true for the controller proposed in [14], because each of its local controller needs to access data from outside its neighborhood.

Example: Assume a fleet of 3 UAVs, with communication links and data flow shown in Fig. 4.2.

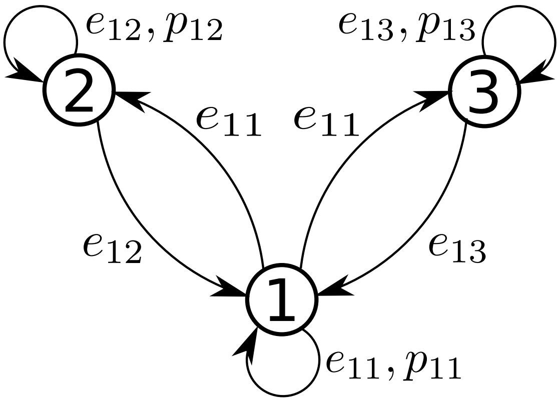


Figure 4.2: 3 UAVs and their undirected communication links.

From Fig. 4.2 and assuming unitary weight in all existing communications nodes and unitary weight in the own error, the Laplacian $\boldsymbol{L}$, the own error gain $\boldsymbol{\Lambda}$ and $\boldsymbol{H}$ matrices are

$$\boldsymbol{L} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{I}_3, \quad \boldsymbol{H} = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{bmatrix}. \tag{4.36}$$

Choosing $\boldsymbol{k}_{dx} = \text{diag}([k_{dx} \ \dots \ k_{dx}]) = k_{dx}\boldsymbol{I}_n$ and $\boldsymbol{k}_{px} = \text{diag}([k_{px} \ \dots \ k_{px}]) = k_{dx}\boldsymbol{I}_n$, a specific case of the sliding vector (4.33) is obtained

$$\boldsymbol{S}_x(t) = \ddot{\boldsymbol{E}}_x^c(t) + k_{dx}\dot{\boldsymbol{E}}_x^c(t) + k_{px}\boldsymbol{E}_x^c(t) = \boldsymbol{H}\left(\ddot{\boldsymbol{E}}_x(t) + k_{dx}\dot{\boldsymbol{E}}_x(t) + k_{px}\boldsymbol{E}_x(t)\right). \tag{4.37}$$

where $\boldsymbol{S}_x(t) = [\, s_{x1}(t) \ \dots \ s_{xn}(t)\,]^T$ is the concatenation of the n UAVs sliding surfaces

$$s_{xi}(t) = \ddot{e}_{xi}^c(t) + k_{dx}\dot{e}_{xi}^c(t) + k_{px}e_{xi}^c(t). \tag{4.38}$$

It is worth to note that $\boldsymbol{S}_x(t)$ is a more restrict sliding surface than $\bar{\boldsymbol{S}}_x(t)$. If the controller does not work for $\boldsymbol{S}_x(t)$, it also does not work for $\bar{\boldsymbol{S}}_x(t)$.

Choosing $k_{dx} = k_{px} = 1$, the component u_{xb2} of the controller (4.32c) for the 2nd UAV is

$$u_{xb2} = -\xi_{x2}^{-1}\frac{(-s_{x1} + 2s_{x2})\left(s_{x1}^2 + s_{x2}^2 + s_{x3}^2\right)^{1/2} 4\eta_1}{11s_{x1}^2 + 5s_{x2}^2 + 5s_{x3}^2 - 10s_{x1}s_{x2} - 10s_{x1}s_{x3} + 2s_{x2}s_{x3}}. \tag{4.39}$$

It follows that in order to implement the control signal (4.39) it is necessary to have access to p_{x2}, p_{x2}^d, e_{x1}^c, e_{x2}^c, e_{x3}^c, s_{x1}, s_{x2} and s_{x3}. By hypothesis (see also Fig. 4.1), the 2nd UAV receives the desired position p_{x2}^d. From Fig. 4.1 and Fig. 4.2, the 2nd UAV has access to its own position p_{x2} and receives the error e_{x1} from the 1st UAV and uses it together its own error e_{x2} to calculate its own coupled error e_{x2}^c using (4.22) and, after, its own sliding surface s_{x2} by using (4.30). That is, the 2nd UAV has access only to the locally computed own coupled error e_{x2}^c and sliding surface s_{x2} and it should not depend on s_{x1}, s_{x3}, e_{x1}^c, and e_{x3}^c since it has not access to them. Thus, the control signal obtained in (4.39), which is obtained by using the control law presented in [14] and the Laplacian matrix $\boldsymbol{L}$ that represents the communication topology of Figure 4.2 as input, does not respect the communication topology used to generate the controller.

The explanation is the control signal $\boldsymbol{U}_{xb}$, calculated by (4.32c). To compute $\|\bar{\boldsymbol{S}}_x^T(t)\|$, all $\bar{s}_{xj}$, $j \in \{1, 2..., n\}$ must be known, even $j \neq \mathcal{N}_i$. A second source is the term $\boldsymbol{H}^T\bar{\boldsymbol{S}}_x(t)$. Using the stricter $\boldsymbol{S}_x(t)$ instead of $\bar{\boldsymbol{S}}_x(t)$ and knowing that $\boldsymbol{H}^T = \boldsymbol{H}$, it can be seen that

$$\boldsymbol{H}^T\boldsymbol{S}_x(t) = \boldsymbol{H}^2\left(\ddot{\boldsymbol{E}}_x(t) + k_{dx}\dot{\boldsymbol{E}}_x(t) + k_{px}\boldsymbol{E}_x(t)\right) = \ddot{\boldsymbol{E}}_x^{cc}(t) + k_{dx}\dot{\boldsymbol{E}}_x^{cc}(t) + k_{px}\boldsymbol{E}_x^{cc}(t), \tag{4.40}$$

where the *bi-composed* error $\boldsymbol{E}_x^{cc}(t)$ is defined as

$$\boldsymbol{E}_x^{cc}(t) \triangleq \boldsymbol{H}^2\boldsymbol{E}_x(t). \tag{4.41}$$

Using the numerical example (4.36), it is obtained

$$\boldsymbol{H}^2 = \begin{bmatrix} 11 & -5 & -5 \\ -5 & 5 & 1 \\ -5 & 1 & 5 \end{bmatrix}, \tag{4.42}$$

which can be seen, in this example, that the *bi-composed* error is function of all errors, and not only the neighborhood error, since all elements of the matrix are non-null.

4.4 FIRST PROPOSED CONTROL ARCHITECTURE

Here it is proposed a controller that enhances the previously presented controller to achieve the same control objectives but by using only available data, i.e., the local, the neighborhood and the virtual leader data. It is assumed the same general setup of [14] presented in the previous section, but uses the more restrict $s_{xi}(t)$ defined in (4.38). In this case, the error dynamics in sliding mode is given by

$$\ddot{e}_{xi}^c(t) + k_{dx}\dot{e}_{xi}^c(t) + k_{px}e_{xi}^c(t) = 0, \tag{4.43}$$

which is exponentially asymptotically stable for project parameters $k_{dxi}, k_{pxi} > 0$.

The proposed control law for i-th UAV is

$$u_{xi}(t) = u_{xai}(t) + u_{xbi}(t), \tag{4.44a}$$

$$u_{xai}(t) = \xi_{xi}^{-1}\left(\xi_{xi}\ddot{x}_i(t) + \dddot{p}_{xi}^d(t) - k_{dx}\ddot{e}_{xi}(t) - k_{px}\dot{e}_{1i}(t)\right), \tag{4.44b}$$

$$u_{xbi}(t) = -\xi_{xi}^{-1}\text{sign}(s_{xi}(t))\eta_{xi}, \tag{4.44c}$$

$$\dot{\tau}_{xi}(t) = -\xi_{xi}\tau_{xi}(t) + \xi_{xi}u_{xi}(t), \tag{4.44d}$$

which contains only information from the virtual leader, the local i-th UAV and its neighborhood $\mathcal{N}_i$. The neighborhood information is contained in $s_{xi}(t)$, defined in (4.38), which is function of $e_{xi}^c(t)$ from (4.22), which is function of the own local error $e_i(t)$ and the neighborhood errors $e_j(t)$, $j \in \mathcal{N}_i$.

Equations (4.44a)-(4.44d) can be vectorized as

$$\begin{aligned}
\boldsymbol{U}_x(t) &= \boldsymbol{U}_{xa}(t) + \boldsymbol{U}_{xb}(t), && (4.45)\\
\boldsymbol{U}_{xa}(t) &= \ddot{\boldsymbol{P}}_x(t) + \Xi_x^{-1}\left(\dddot{\boldsymbol{P}}_x^d(t) - k_{dx}\ddot{\boldsymbol{E}}_x(t) - k_{px}\dot{\boldsymbol{E}}_x(t)\right), && (4.46)\\
\boldsymbol{U}_{xb}(t) &= -\Xi_x^{-1}\mathrm{diag}\left(\frac{\eta_{xi}}{|s_{xi}(t)|}\right)\boldsymbol{S}_x(t). && (4.47)\\
\dot{\boldsymbol{\tau}}_x(t) &= -\Xi_x\boldsymbol{\tau}_x(t) + \Xi_x\boldsymbol{U}_x(t). && (4.48)
\end{aligned}$$

Remark 3 *Restricting $k_{dxi} = k_{dx}$ and $k_{pxi} = k_{px}$ means that all UAVs have sliding surfaces that share the same control bandwidth. This is reasonable if all UAVs have similar physical, actuator, and aerodynamic characteristics. However, if there are distinct UAVs, the constants must be chosen to respect the control bandwidth of the UAV with slowest dynamics.*

4.4.1 Stability proof

The stability proof is adapted from the proof presented in [14]. The following Lyapunov functional candidate is proposed

$$\mathcal{V}_x(t) = \frac{1}{2}\boldsymbol{S}_x^T(t)\boldsymbol{H}^{-1}\boldsymbol{S}_x(t). \qquad (4.49)$$

Note that, being $\boldsymbol{H}$ a positive definite matrix, $\boldsymbol{H}^{-1}$ is also positive definite matrix, so $\mathcal{V}_x(t)$ is always positive for $\boldsymbol{S}_x(t) \neq 0$.

Since $\boldsymbol{H}^{-1}$ is constant, the derivative of $\mathcal{V}_x(t)$ is

$$\dot{\mathcal{V}}_x(t) = \boldsymbol{S}_x^T(t)\boldsymbol{H}^{-1}\dot{\boldsymbol{S}}_x(t). \qquad (4.50)$$

By using (4.26), (3.19), (4.25), the equation (4.37) is rewritten as

$$\begin{aligned}
\boldsymbol{S}_x(t) &= \boldsymbol{H}\left(\ddot{\boldsymbol{P}}_x(t) - \ddot{\boldsymbol{P}}_x^d(t) + k_{dx}\dot{\boldsymbol{E}}_x(t) + k_{px}\boldsymbol{E}_x(t)\right)\\
&= \boldsymbol{H}\left(\boldsymbol{\tau}_x(t) + \boldsymbol{D}_x(t) - \ddot{\boldsymbol{P}}_x^d(t) + k_{dx}\dot{\boldsymbol{E}}_x(t) + k_{px}\boldsymbol{E}_x(t)\right).
\end{aligned} \qquad (4.51)$$

Deriving $\boldsymbol{S}_x(t)$ and, after, using (4.48),(4.25), (4.45), (4.46)

$$\begin{aligned}
\dot{\boldsymbol{S}}_x(t) &= \boldsymbol{H}\left(\dot{\boldsymbol{\tau}}_x(t) + \dot{\boldsymbol{D}}_x(t) - \dddot{\boldsymbol{P}}_x^d(t) + k_{dx}\ddot{\boldsymbol{E}}_x(t) + k_{px}\dot{\boldsymbol{E}}_x(t)\right)\\
&= \boldsymbol{H}\left(\Xi_x\left(\boldsymbol{U}_x(t) - \boldsymbol{\tau}_x(t)\right) + \dot{\boldsymbol{D}}_x(t) - \dddot{\boldsymbol{P}}_x^d(t) + k_{dx}\ddot{\boldsymbol{E}}_x(t) + k_{px}\dot{\boldsymbol{E}}_x(t)\right)\\
&= \boldsymbol{H}\left(\Xi_x\left(\boldsymbol{U}_{xa}(t) + \boldsymbol{U}_{xb}(t) - \ddot{\boldsymbol{P}}_x(t) + \boldsymbol{D}_x(t)\right) - \dddot{\boldsymbol{P}}_x^d(t) + \dot{\boldsymbol{D}}_x(t) + k_{dx}\ddot{\boldsymbol{E}}_x(t) + k_{px}\dot{\boldsymbol{E}}_x(t)\right)\\
&= \boldsymbol{H}\left(\Xi_x\boldsymbol{U}_{xb}(t) + \dot{\boldsymbol{D}}_x(t) + \Xi_x\boldsymbol{D}_x(t)\right).
\end{aligned} \qquad (4.52)$$

Using (4.52) in (4.50) and, after, (4.47)

$$\begin{aligned}\dot{\mathcal{V}}_x(t) &= \boldsymbol{S}_x^T(t)\left(\Xi_x\boldsymbol{U}_{xb}(t) + \dot{\boldsymbol{D}}_x(t) + \Xi_x\boldsymbol{D}_x(t)\right) \\ &= \boldsymbol{S}_x^T(t)\left(\dot{\boldsymbol{D}}_x(t) + \Xi_x\boldsymbol{D}_x(t)\right) + \boldsymbol{S}_x^T(t)\Xi_x\boldsymbol{U}_{xb}(t) \\ &= \boldsymbol{S}_x^T(t)\left(\dot{\boldsymbol{D}}_x(t) + \Xi_x\boldsymbol{D}_x(t)\right) - \boldsymbol{S}_x^T(t)\text{diag}\left(\frac{\eta_{xi}}{|s_{xi}(t)|}\right)\boldsymbol{S}_x(t) \\ &= \sum s_{xi}(t)\left(\dot{d}_{xi}(t) + \xi_{xi}d_{xi}(t)\right) - \sum \frac{s_{xi}^2(t)}{|s_{xi}(t)|}\eta_{xi} \\ &= \sum\left(s_{xi}(t)\left(\dot{d}_{xi}(t) + \xi_{xi}d_{xi}(t)\right) - |s_{xi}(t)|\eta_{xi}\right) \\ &\leq \sum |s_{xi}(t)|\left(|\dot{d}_{xi}(t)| + \xi_{xi}|d_{xi}(t)| - \eta_{xi}\right).\end{aligned} \tag{4.53}$$

The upper bounds to the disturbance and its derivative are given by respectively $\Delta_{xi} \geq |d_{xi}(t)|$ and $\Delta_{dxi} \geq |\dot{d}_{xi}(t)|$, which are calculated by, respectively, Eqs. (4.17) and (4.20). By choosing η_{xi} satisfying

$$\eta_{xi} > \Delta_{dxi} + \xi_{xi}\Delta_{xi} + \epsilon, \quad \forall i \in \{1, ..., n\}, \tag{4.54}$$

for some arbitrarily chosen constant $\epsilon > 0$, it is obtained

$$\dot{\mathcal{V}}_x(t) \leq -\sum_{i=1}^{n} |s_{xi}(t)|\epsilon = -\epsilon\sum_{i=1}^{n} |s_{xi}(t)| = -\epsilon\left\|\boldsymbol{S}_x(t)\right\|_1, \tag{4.55}$$

where $\|\boldsymbol{S}_x(t)\|_1$ is the 1-norm of $\boldsymbol{S}_x(t)$. Using the fact that the 1-norm is greater than the Euclidean norm of the same vector, then

$$\dot{\mathcal{V}}_x(t) \leq -\epsilon\|\boldsymbol{S}_x(t)\|, \tag{4.56}$$

which means that $\mathcal{V}_x(t)$ and, therefore, $\boldsymbol{S}_x(t)$ go to zero in finite time [42]. On the sliding surface, the system behaves as a stable linear system given by Eq. (4.43) and the error converges asymptotically to zero.

4.5 SECOND PROPOSED CONTROL ARCHITECTURE

Here the controller proposed in Section 4.4 is improved. As made in [41] to a single-agent problem, the control, only the chattering u_{xbi} is filtered by the low-pass filter, whereas the non-chattering component of the controller is renamed to τ_{xai} and reprojected to be applied directly in the plant, bypassing the low-pass filter. The proposed control structure is depicted in Fig. 4.3.

The proposed control law for i-th UAV is

$$\tau_{xi}(t) = \tau_{xai}(t) + \tau_{xbi}(t), \tag{4.57}$$

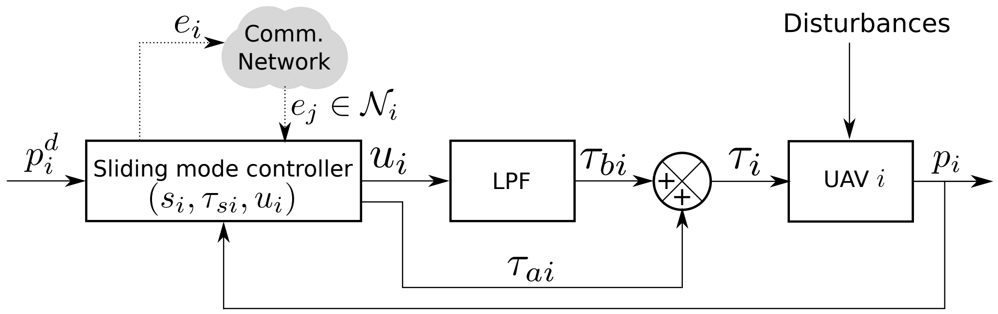


Figure 4.3: Block diagram of the control structure.

where $\tau_{xai}(t)$ and $\tau_{xbi}(t)$ are, respectively, a smooth signal and a filtered signal of the control law, computed by

$$\tau_{xai}(t) = \ddot{p}_{xi}^d(t) - k_{dx}\dot{e}_{xi}(t) - k_{px}e_{xi}(t), \tag{4.58}$$

$$\dot{\tau}_{xbi}(t) = -\xi_{xi}\tau_{xbi}(t) + u_{xi}(t), \tag{4.59}$$

$$u_{xi}(t) = -\text{sign}(s_{xi}(t))\eta_{xi}. \tag{4.60}$$

It is worth to note that the low pass filter (4.59) has distinct gain from (4.44d). This allows to define $\xi_{xi} = 0$ if desired, which transforms the low-pass filter SMC to the more specific 2nd order SMC.

It is also worth to note that if $\tau_{xbi}(t)$ is defined as null, it is achieved the NLDI controller presented in Section 3.4.

As in the previous proposed controller, the proposed control law given by Eqs. (4.57)–(4.60) contains only information from the virtual leader (or from a broadcasting non-virtual leader), from the own i-th UAV, and from its neighborhood $\mathcal{N}_i$. The neighborhood information is contained in $s_{xi}(t)$, defined in Eq. (4.38), which is function of $e_{xi}^c(t)$ from Eq. (4.24), which is function of the own local error $e_i(t)$ and the neighborhood errors $e_j(t)$, $j \in \mathcal{N}_i$.

The proposed sliding mode control law can be vectorized as

$$\boldsymbol{\tau}_x(t) = \boldsymbol{\tau}_{xa}(t) + \boldsymbol{\tau}_{xb}(t), \tag{4.61}$$

where $\boldsymbol{\tau}_{xa}(t)$ and $\boldsymbol{\tau}_{xb}(t)$ are computed by

$$\boldsymbol{\tau}_{xa}(t) = \ddot{\boldsymbol{P}}_x^d(t) - k_{dx}\dot{\boldsymbol{E}}_x(t) - k_{px}\boldsymbol{E}_x(t), \tag{4.62}$$

$$\dot{\boldsymbol{\tau}}_{xb}(t) = -\boldsymbol{\Xi}_x\boldsymbol{\tau}_{xb}(t) + \boldsymbol{U}_x(t), \tag{4.63}$$

$$\boldsymbol{U}_x(t) = -\text{diag}\left(\frac{\eta_{xi}}{|s_{xi}(t)|}\right)\boldsymbol{S}_x(t). \tag{4.64}$$

Remark 4 *It is worth to note that the proposed controller uses only up to the second derivative of the desired position, whereas in the controller from [14] presented in Section 4.3 and in the proposed correction presented in Section 4.4, the third one is also used.*

Remark 5 *The $\tau_{xai}(t)$ component of the control signal $\tau_{xi}(t)$, described in* (4.58)*, bypasses the filter, as shown in Fig. 4.3. Being applied directly in the UAV, the bandwidth of this signal is not restricted by the*

LPF. As result, this approach provides a faster transient response when compared to the approach from Section 4.4, as can be seen in the simulation results in Section 4.6.

4.5.1 Stability proof

The Lyapunov functional candidate and its derivative are the same from the first proposed controller, and are given by, respectively, (4.49) and (4.50)

By using Eqs. (4.25), (4.61), and (4.62), the sliding surface given by Eq. (4.37) can be rewritten as

$$\begin{aligned}\boldsymbol{S}_x(t) &= \boldsymbol{H}\left(\ddot{\boldsymbol{P}}_x(t) - \ddot{\boldsymbol{P}}_x^d(t) + k_{dx}\dot{\boldsymbol{E}}_x(t) + k_{px}\boldsymbol{E}_x(t)\right)\\ &= \boldsymbol{H}\left(\boldsymbol{\tau}_{xa}(t) + \boldsymbol{\tau}_{xb}(t) + \boldsymbol{D}_x(t) - \ddot{\boldsymbol{P}}_x^d(t) + k_{dx}\dot{\boldsymbol{E}}_x(t) + k_{px}\boldsymbol{E}_x(t)\right)\\ &= \boldsymbol{H}\left(\boldsymbol{D}_x(t) + \boldsymbol{\tau}_{xb}(t)\right).\end{aligned} \tag{4.65}$$

By deriving Eq. (4.65) and after using Eq. (4.63), $\dot{\mathcal{V}}_x(t)$ is rewritten to

$$\begin{aligned}\dot{\mathcal{V}}_x(t) &= \boldsymbol{S}_x^T(t)\left(\dot{\boldsymbol{D}}_x(t) + \dot{\boldsymbol{\tau}}_{xb}(t)\right)\\ &= \boldsymbol{S}_x^T(t)\left(\dot{\boldsymbol{D}}_x(t) + \boldsymbol{U}_x(t) - \boldsymbol{\Xi}_x\boldsymbol{\tau}_{xb}(t)\right).\end{aligned} \tag{4.66}$$

By substituting (4.61) in (4.25) and, after, by using (4.37), it is obtained

$$\begin{aligned}\boldsymbol{\tau}_{xb}(t) &= \ddot{\boldsymbol{P}}_x - \boldsymbol{\tau}_{xa}(t) - \boldsymbol{D}_x(t)\\ &= \ddot{\boldsymbol{P}}_x - \ddot{\boldsymbol{P}}_x^d(t) + k_{dx}\dot{\boldsymbol{E}}_x(t) + k_{px}\boldsymbol{E}_x(t) - \boldsymbol{D}_x(t)\\ &= \boldsymbol{H}^{-1}\boldsymbol{S}_x(t) - \boldsymbol{D}_x(t).\end{aligned} \tag{4.67}$$

By using (4.67) in (4.66), and by assuming that $\boldsymbol{\Xi}_x = \text{diag}([\xi_x \ \dots \ \xi_x]) = \xi_x\boldsymbol{I}_n$, i.e., that the filter has the same cutoff frequency to all UAVS, it is obtained

$$\begin{aligned}\dot{\mathcal{V}}_x(t) &= \boldsymbol{S}_x^T(t)\left(\dot{\boldsymbol{D}}_x(t) + \boldsymbol{U}_x(t) - \boldsymbol{\Xi}_x\boldsymbol{H}^{-1}\boldsymbol{S}_x(t) + \boldsymbol{\Xi}_x\boldsymbol{D}_x(t)\right)\\ &= \boldsymbol{S}_x^T(t)\dot{\boldsymbol{D}}_x(t) + \boldsymbol{S}_x^T(t)\boldsymbol{\Xi}_x\boldsymbol{D}_x(t) + \boldsymbol{S}_x^T(t)\boldsymbol{U}_x(t) - \boldsymbol{S}_x^T(t)\boldsymbol{\Xi}_x\boldsymbol{H}^{-1}\boldsymbol{S}_x(t)\\ &= \sum_{i=1}^{n}\left(s_{xi}(t)\dot{d}_{xi}(t) - s_{xi}(t)\xi_x d_{xi}(t) - |s_{xi}(t)|\eta_{xi}\right) - \xi_x\boldsymbol{S}_x^T(t)\boldsymbol{H}^{-1}\boldsymbol{S}_x(t)\\ &\leq \sum_{i=1}^{n}|s_{xi}(t)|\left(|\dot{d}_{xi}(t)| + \xi_x|d_{xi}(t)| - \eta_{xi}\right),\end{aligned} \tag{4.68}$$

where $-\xi_x\boldsymbol{S}_x^T(t)\boldsymbol{H}^{-1}\boldsymbol{S}_x(t) \leq 0$. The upper bounds to the disturbance and its derivative are given by respectively $\Delta_{xi} \geq |d_{xi}(t)|$ and $\Delta_{dxi} \geq |\dot{d}_{xi}(t)|$, which are calculated by, respectively, Eqs. (4.17) and (4.20). By using these upper bounds in Eq. (4.68), it can be seen that

$$\dot{\mathcal{V}}_x(t) \leq \sum_{i=1}^{n}|s_{xi}(t)|\left(\Delta_{dxi} + \xi_x\Delta_{xi} - \eta_{xi}\right). \tag{4.69}$$

By choosing η_{xi} satisfying

$$\eta_{xi} \geq \Delta_{dxi} + \xi_x \Delta_{xi} + \epsilon, \quad \forall i \in \{1, ..., n\}, \tag{4.70}$$

for some arbitrarily chosen constant $\epsilon > 0$, it is obtained

$$\begin{aligned} \dot{\mathcal{V}}_x(t) &\leq -\sum_{i=1}^{n} |s_{xi}(t)|\epsilon = -\epsilon \sum_{i=1}^{n} |s_{xi}(t)| = -\epsilon \, \|\boldsymbol{S}_x(t)\|_1 \\ &\leq -\epsilon \|\boldsymbol{S}_x(t)\|, \end{aligned} \tag{4.71}$$

which means that $\mathcal{V}_x(t)$ and, therefore, $\boldsymbol{S}_x(t)$ go to zero in finite time [42]. On the sliding surface, the system behaves as a stable linear system given by Eq. (4.43) and the error converges asymptotically to zero.

The above demonstration restricted the low-pass-filter to use the same cutoff frequency in all UAVs. To remove this restriction, an alternative stability proof is presented. From (4.66) it is obtained

$$\begin{aligned} \dot{\mathcal{V}}_x(t) &= \boldsymbol{S}_x^T(t)\dot{\boldsymbol{D}}_x(t) - \boldsymbol{S}_x^T(t)\boldsymbol{\Xi}_x\boldsymbol{\tau}_{xb}(t) + \boldsymbol{S}_x^T(t)\boldsymbol{U}_x(t) \\ &= \sum_{i=1}^{n} \Big(s_{xi}(t)\dot{d}_{xi}(t) - s_{xi}(t)\xi_{xi}\tau_{xbi}(t) - |s_{xi}(t)|\eta_{xi} \Big) \\ &\leq \sum_{i=1}^{n} |s_{xi}(t)| \left(|\dot{d}_{xi}(t)| + \xi_{xi}|\tau_{xbi}(t)| - \eta_{xi} \right). \end{aligned} \tag{4.72}$$

The obtained result is similar to (4.68), but has $\xi_{xi}|\tau_{xbi}(t)|$ instead of $\xi_{xi}|d_{xi}(t)|$. IT can be defined that $\tau_{xbi}(0) = 0$, i.e., the filter starts with a null output, which means that, initially, $\xi_{xi}|\tau_{xbi}(0)| = 0$. Assuming that is proven that $\dot{\mathcal{V}}_x(t) \leq -\epsilon\|\boldsymbol{S}_x(t)\|$, this means that the magnitude of $S_x(t)$ gets smaller with time which, by (4.65), means that the sum $\boldsymbol{D}_x(t) + \boldsymbol{\tau}_{xb}(t)$ approaches zero, which means that $\|\boldsymbol{\tau}_{xb}(t)\|$ approaches $\|\boldsymbol{D}_x(t)\|$. Since $\tau_{xbi}(0) = 0$ and it reaches $\tau_{xbi}(t) = d_{xi}(t)$ when $S_x(t)$ reaches zero, then

$$|\tau_{xbi}(t)| \leq |d_{xi}(t)| \leq \Delta_{xi}. \tag{4.73}$$

By using the upper bounds in Eq. (4.72), it can be seen that

$$\dot{\mathcal{V}}_x(t) \leq \sum_{i=1}^{n} |s_{xi}(t)| \left(\Delta_{dxi} + \xi_{xi}\Delta_{xi} - \eta_{xi}\right). \tag{4.74}$$

By choosing η_{xi} satisfying

$$\eta_{xi} \geq \Delta_{dxi} + \xi_{xi}\Delta_{xi} + \epsilon, \quad \forall i \in \{1, ..., n\}, \tag{4.75}$$

for some arbitrarily chosen constant $\epsilon > 0$, it is obtained

$$\begin{aligned} \dot{\mathcal{V}}_x(t) &\leq -\sum_{i=1}^{n} |s_{xi}(t)|\epsilon = -\epsilon \sum_{i=1}^{n} |s_{xi}(t)| = -\epsilon \, \|\boldsymbol{S}_x(t)\|_1 \\ &\leq -\epsilon \|\boldsymbol{S}_x(t)\|, \end{aligned} \tag{4.76}$$

which means that $\mathcal{V}_x(t)$ and, therefore, $\boldsymbol{S}_x(t)$ go to zero in finite time [42]. On the sliding surface, the system behaves as a stable linear system given by Eq. (4.43) and the error converges asymptotically to zero.

Remark 6 *Note that the sliding surface given by Eq.* (4.37)*, when rewritten in Eq.* (4.65)*, is function only of the disturbance* $\boldsymbol{D}_x(t)$ *and the output of the filter* $\boldsymbol{\tau}_{xb}(t)$*. This has two main implications:*

1. *Since it is shown here that* $\boldsymbol{S}_x(t) \to \boldsymbol{0}$*, it follows that* $\boldsymbol{\tau}_{xb}(t) \to -\boldsymbol{D}_x(t)$*. In this way,* $\boldsymbol{\tau}_{xb}(t)$ *estimates and compensates disturbances. Since the effect of airflow being not aligned to the fuselage is a disturbance, the presence of a disturbance compensation shows that the wind effect can be neglected in the initial model if this effect has known bounds.*
2. *If the disturbance is null at* $t = 0$*,* $\boldsymbol{S}(0) = \boldsymbol{0}$ *if* $\boldsymbol{\tau}_{xb}(0) = \boldsymbol{0}$ *and the system already starts in sliding condition. Similarly, if the known disturbance upper bound is relatively small, the system starts near the sliding surface and converges fast to the sliding surface.*

4.6 SIMULATION

In this section, a simulation is made to show the effectiveness of the proposed controllers. First, in Subsection 4.6.1, the UAVs as modeled as simple double integrators, as in [14], since the objective is to compare the proposed controllers to the controller from the literature. After, in Subsection 4.6.2, the second proposed control architecture is evaluated using the realistic aircraft nonlinear model.

4.6.1 Double integrator simulation

It is used a scenario of 5 UAVs with communication links described by Fig. 4.4.

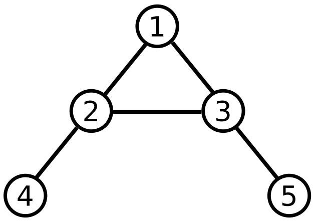
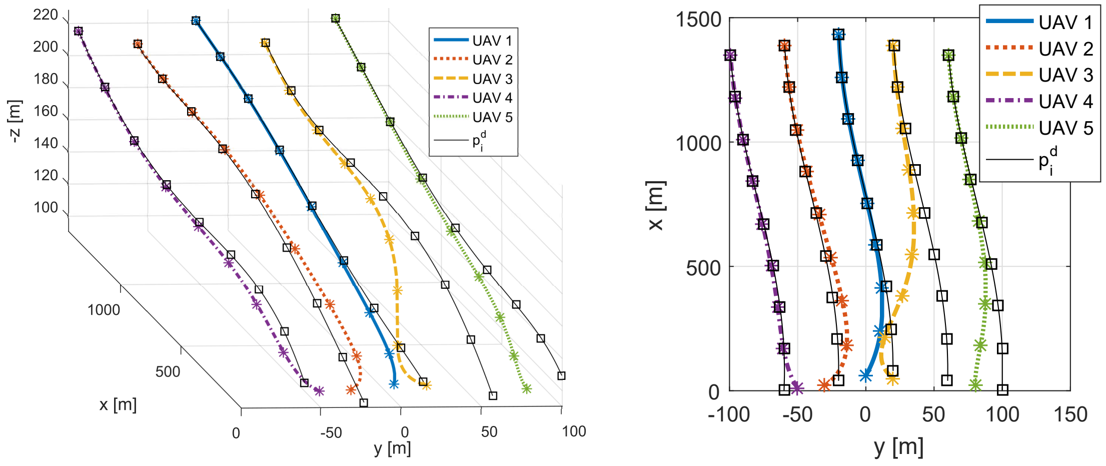


Figure 4.4: 5 UAVs and their undirected communication links. The virtual leader is not shown here. All UAVs has access to the virtual leader's trajectory information.

The matrices

$$\boldsymbol{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \tag{4.77}$$

and $\boldsymbol{\Lambda} = \boldsymbol{I}_5$ are chosen in order to give the same weight for the UAV own error and for each of its relative errors. The choice $k_{px} = k_{py} = k_{pz} = 0.5$ and $k_{dx} = k_{dy} = k_{dz} = 0.0625$ provides a critically damped sliding surface with natural frequency $\omega_n = 0.25$ rad/s. These gains are chosen relatively small, as a way to limit the maximum commanded acceleration, even if the UAVs are initially far from their desired position. The low pass filters are settled such that $\boldsymbol{\Xi} = \boldsymbol{I}_5$.



(a) Trajectory (3D view)

(b) Trajectory (above view)

Figure 4.5: Desired trajectory and UAV position achieved when using the Proposed Controller II.

It is considered a fleet with a non-rectilinear 3D trajectory which is defined by the virtual leader path given by

$$\begin{cases} x_0(t) = 80 + 45t \quad [\mathrm{m}], \\ y_0(t) = 20\cos(0.1t) \quad [\mathrm{m}], \\ \gamma_0(t) = \frac{\pi}{36}\ \mathrm{rad}, \quad (z_0(0) = -100\ \mathrm{m}). \end{cases} \tag{4.78}$$

For easy visualization, it is considered a time-varying formation whose horizontal projection in the reference frame has a V-shape and the altitude has time-varying oscillation. Accordingly, the formation rotation matrix $\boldsymbol{D}^L_{\mathrm{NED}}(t)$ in (3.23) is defined as $\boldsymbol{D}^{\chi 0}_{\mathrm{NED}}$ (Eq. (2.8)) and the clearance vectors $\tilde{\boldsymbol{p}}_{Li}(t)$ related to the virtual leader are

$$\tilde{\boldsymbol{p}}_{L1}(t) = \begin{bmatrix} 0 \\ 0 \\ 10\sin(0.1t) \end{bmatrix}, \quad \tilde{\boldsymbol{p}}_{L2}(t) = \begin{bmatrix} -40 \\ -40 \\ 10\sin(0.1t + 2\pi/5) \end{bmatrix}, \quad \tilde{\boldsymbol{p}}_{L3}(t) = \begin{bmatrix} -40 \\ 40 \\ 10\sin(0.1t + 4\pi/5) \end{bmatrix},$$
$$\tilde{\boldsymbol{p}}_{L4}(t) = \begin{bmatrix} -80 \\ -80 \\ 10\sin(0.1t + 6\pi/5) \end{bmatrix}, \quad \tilde{\boldsymbol{p}}_{L5}(t) = \begin{bmatrix} -80 \\ 80 \\ 10\sin(0.1t + 8\pi/5) \end{bmatrix}. \tag{4.79}$$

The initial position of each UAV is defined as

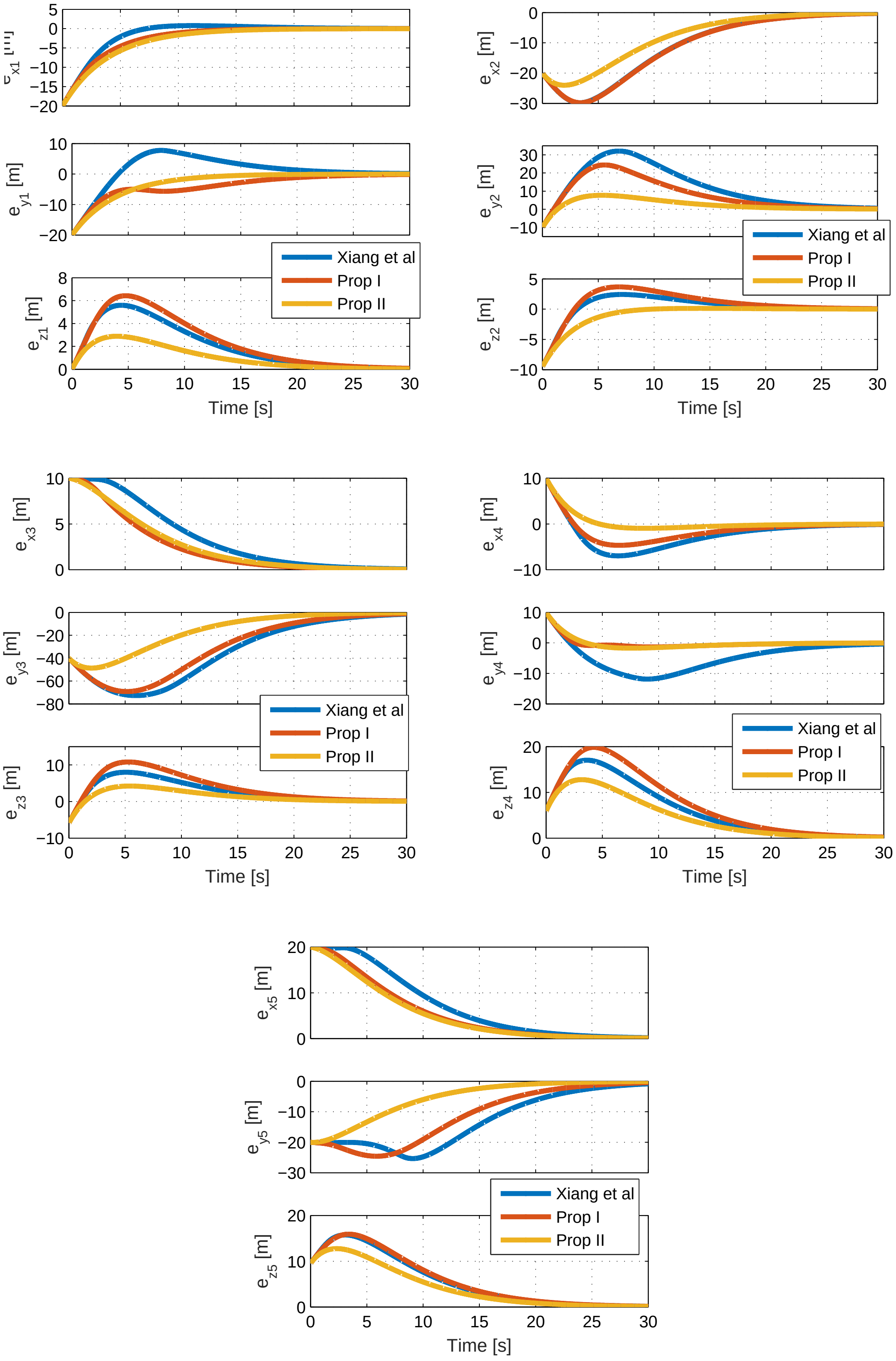


Figure 4.6: Position error.

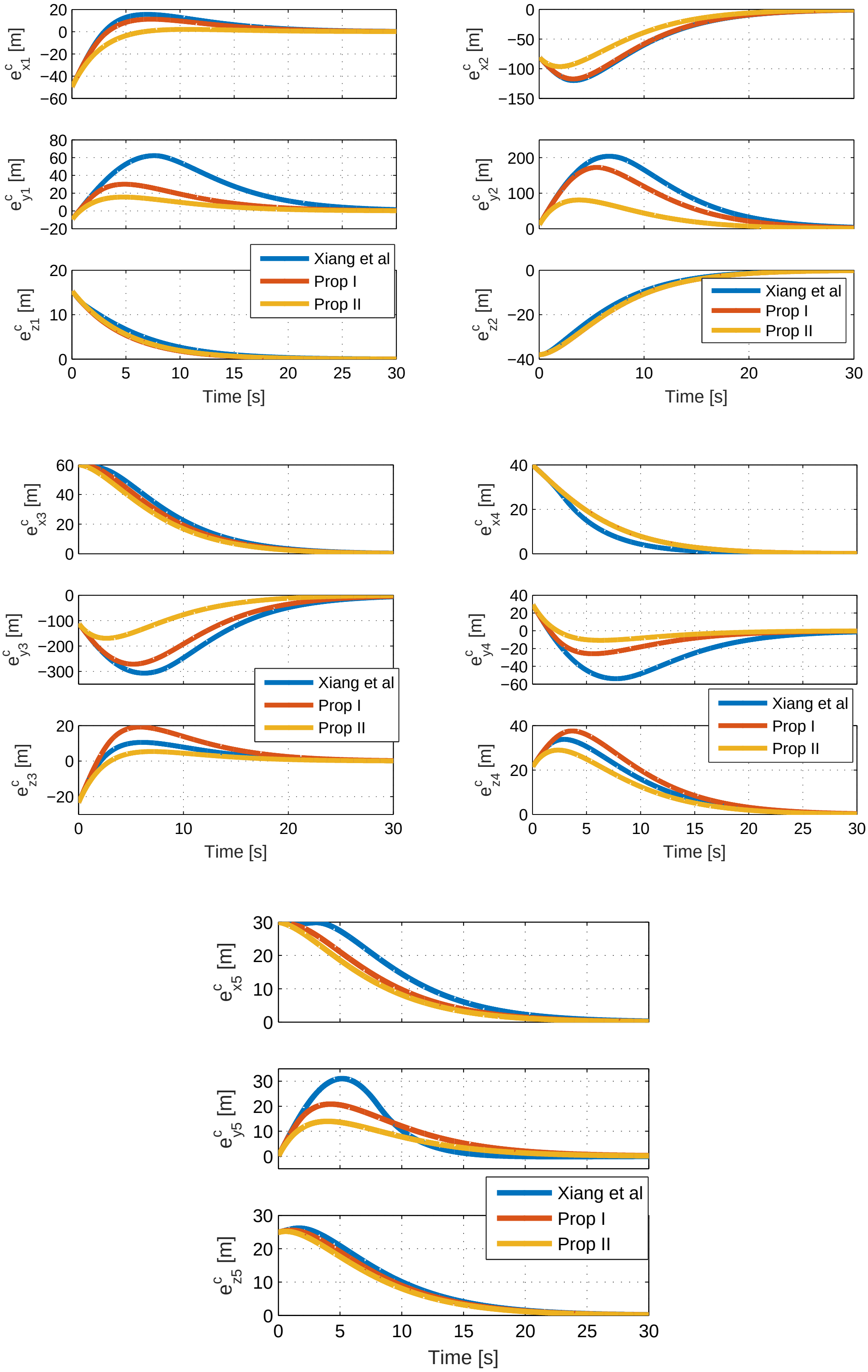
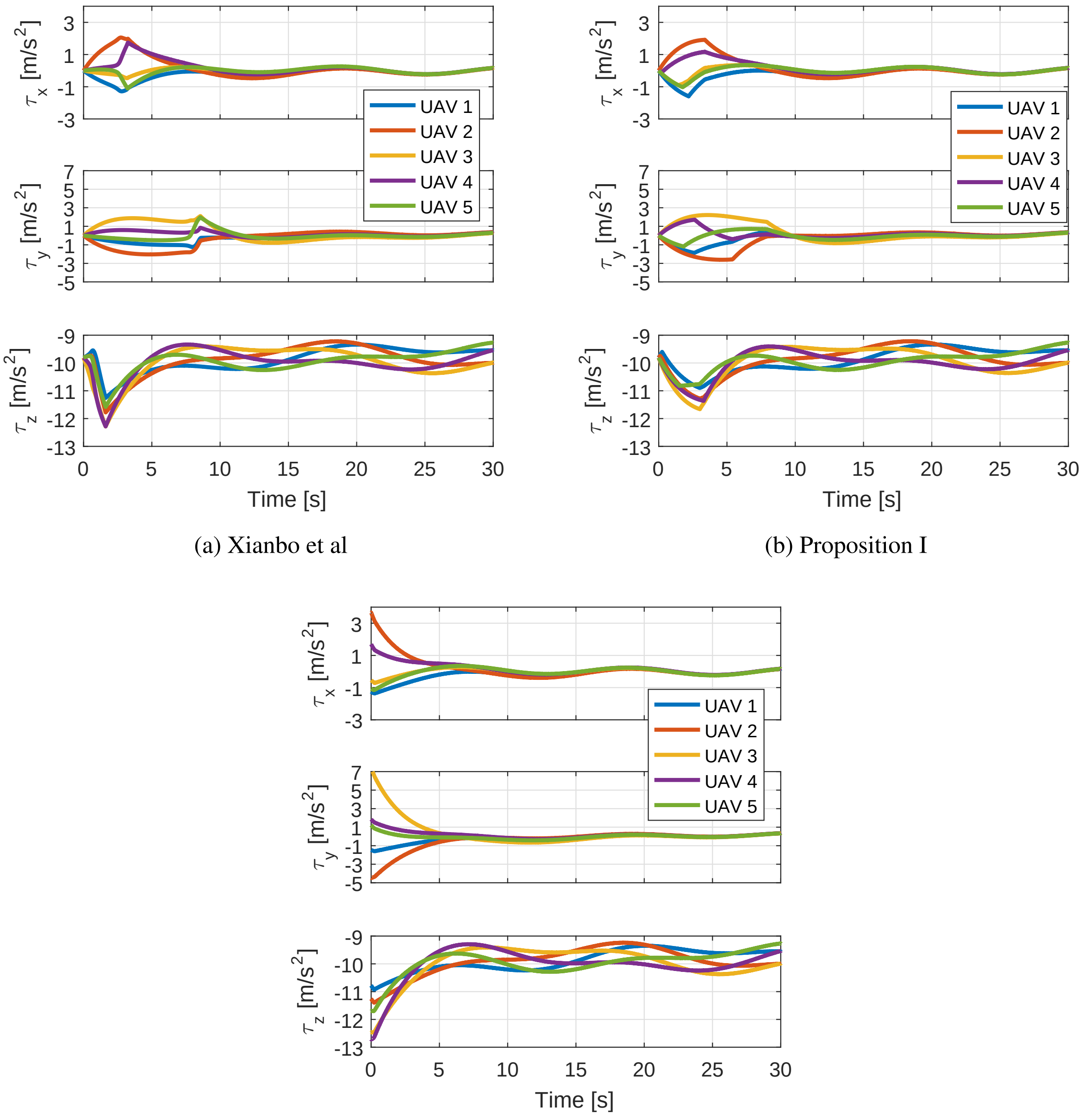


Figure 4.7: Coupled error.



(a) Xianbo et al

(b) Proposition I

(c) Proposition II

Figure 4.8: The filtered controller output

$$\boldsymbol{p}_{\text{NED},1}(0) = \begin{bmatrix} 60 \\ 0 \\ -100 \end{bmatrix}, \quad \boldsymbol{p}_{\text{NED},2}(0) = \begin{bmatrix} 20 \\ -30 \\ -100 \end{bmatrix}, \quad \boldsymbol{p}_{\text{NED},3}(0) = \begin{bmatrix} 50 \\ 20 \\ -100 \end{bmatrix},$$
$$\boldsymbol{p}_{\text{NED},4}(0) = \begin{bmatrix} 10 \\ -50 \\ -100 \end{bmatrix}, \quad \boldsymbol{p}_{\text{NED},5}(0) = \begin{bmatrix} 20 \\ 80 \\ -100 \end{bmatrix}. \tag{4.80}$$

The initial velocity of each UAV is defined as

$$\dot{\boldsymbol{p}}_{\text{NED},1}(0) = \begin{bmatrix} 50 \\ 5 \\ 0 \end{bmatrix}, \quad \dot{\boldsymbol{p}}_{\text{NED},2}(0) = \begin{bmatrix} 40 \\ 10 \\ 0 \end{bmatrix}, \quad \dot{\boldsymbol{p}}_{\text{NED},3}(0) = \begin{bmatrix} 45 \\ -10 \\ 0 \end{bmatrix},$$
$$\dot{\boldsymbol{p}}_{\text{NED},4}(0) = \begin{bmatrix} 40 \\ -5 \\ 0 \end{bmatrix}, \quad \dot{\boldsymbol{p}}_{\text{NED},5}(0) = \begin{bmatrix} 45 \\ 0 \\ 0 \end{bmatrix}. \tag{4.81}$$

Figure 4.5 shows the desired trajectory to each UAV in black and, to illustrate, the trajectory achieved by each UAV by using the second proposed control architecture in distinct colors. Square and '*' markers show respectively the desired and achieved positions in specific and equally spaced time instants. When a '*' is inside the square, the UAV is in its desired position.

The disturbance is simulated as

$$\boldsymbol{b}_i(t) = 0.2 \begin{bmatrix} \cos(0.5t) & \cos(0.5t) & \cos(0.5t) \end{bmatrix}^T, \qquad \forall i \in \{1,2,3,4,5\}. \tag{4.82}$$

From Eq. (4.82), the magnitude of the upper bound vector $\boldsymbol{\delta}_i$ of $\boldsymbol{b}_i(t)$ is calculated as $\|\boldsymbol{\delta}_i\| = 0.35$. The magnitude of the upper bound vector of $\dot{\boldsymbol{\delta}}_i$ is calculated also from Eq. (4.82) as $\|\boldsymbol{\delta}_{di}\| = 0.17$. The upper bound of each component of $\boldsymbol{d}_i(t)$ is computed by Eq. (4.17) resulting in $\Delta_{xi} = \Delta_{yi} = \Delta_{zi} = 0.35$. By simulation experiments it is verified that $\Omega_i = 0.17$ rad/s is an upper bound for the angular velocity amplitude; the upper bound in $\dot{\boldsymbol{d}}_i(t)$ is computed by Eq. (4.20), resulting in $\Delta_{dxi} = \Delta_{dyi} = \Delta_{dzi} = 0.23$. By choosing $\epsilon = 0.42$, it is obtained from Eq. (4.70) that $\eta_{xi} = \eta_{yi} = \eta_{zi} = 1$.

The system is implemented using an ode4 Runge-Kutta solver, with a fixed-step size of 0.1 ms. Since it is impossible to perfectly simulate the effect of a chattering input signal in a continuous differential equation, the controller output is evaluated at 1 ms time steps, and maintained constant between time intervals.

Figure 4.6 show the formation flight error components e_{xi}, e_{yi} and e_{zi} for each i-th UAV, for each control approach. Figure 4.7, shows the coupled error of each i-th UAV, which is given by Eq. (4.26). It can be seen that all approaches work, converging fast to a null error.

It is worth to note that the controller from literature and the first proposed controller has similar performance. However, as discussed in 4.3.1, the controller from literature is conceptually wrong, ignoring the limitations imposed by the communication graph. The proposed controller is correct, respecting the communication graph.

The second proposed controller is faster than the others. Figure 4.8 shows why. The second proposed controller has a controller component that bypasses the low pass filter. This means that the controller responds immediately, whereas the other two starts with null control command and must wait the filtered output grow over time. The instantaneous response, however, is more intense. A less intense sliding surface must be chosen to attenuate the initial peak in the control signal. Also from 4.8, it can be seen that all controllers present smooth, non-chattering outputs.

4.6.2 Nonlinear aircraft model simulation

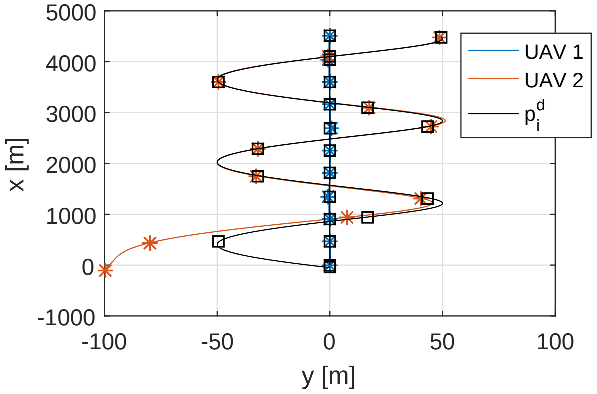
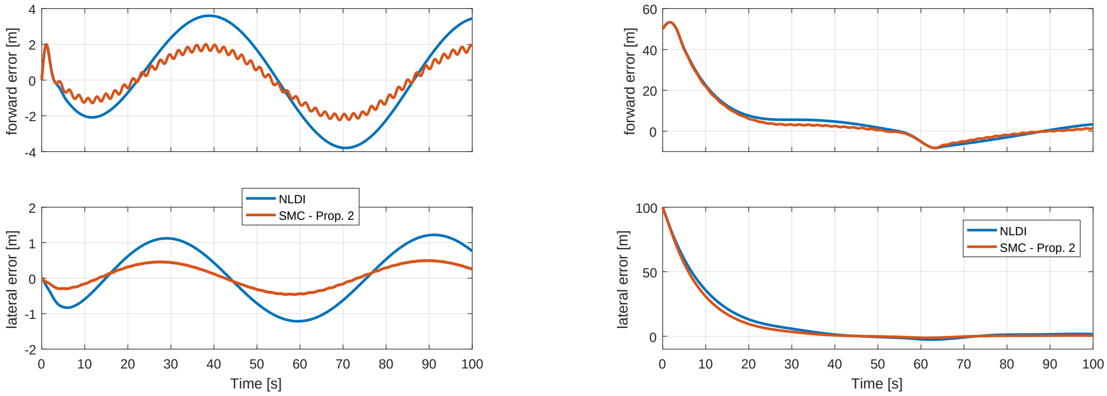


Figure 4.9: Trajectory, as seen from above, for the second proposed robust controller.



(a) UAV 1

(b) UAV 2

Figure 4.10: Position error using a robust and a non robust approaches.

Here the second proposed controller is simulated using the nonlinear aircraft model. The main objective is to evaluate the robust approach versus the non-robust one from Chapter 3, and to evaluate the effect of using a high bandwidth control signal generated by the sliding mode controller in the inner loop controlled aircraft. The scenario is the following: there are two UAVs, and they made the same maneuver as in Subsection 3.8.2. Being only two aircraft, the synchronous part of the controller is not evaluated, which

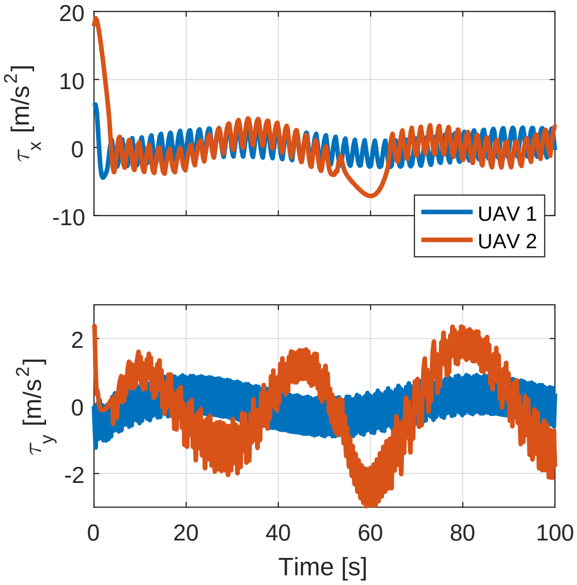


Figure 4.11: The filtered outer loop controller output from Proposition II.

means that $\boldsymbol{L} = \mathbf{0}_{2\times 2}$ and $\boldsymbol{\Lambda} = \boldsymbol{I}_2$. It is chosen $k_{px} = k_{py} = k_{pz} = 0.2419$ and $k_{dx} = k_{dy} = k_{dz} = 2.0560$ to provide a slow and super damped dynamics. Also, the linear controller defined by (3.50) is used to maintain the aircraft in a constant altitude.

The disturbance is applied in the aerodynamic forces as

$$\begin{bmatrix} D_{\text{dist},i}(t) \\ Y_{\text{dist},i}(t) \\ L_{\text{dist},i}(t) \end{bmatrix} = \begin{bmatrix} D_i(t) \\ Y_i(t) \\ L_i(t) \end{bmatrix} + m_i \begin{bmatrix} \cos(0.1t) \\ \cos(0.1t) \\ \cos(0.1t) \end{bmatrix} \tag{4.83}$$

where $D_{\text{dist},i}(t)$, $Y_{\text{dist},i}(t)$, $L_{\text{dist},i}(t)$, the disturbed aerodynamic forces, are used instead of $D_i(t)$, $Y_i(t)$, $L_i(t)$ in the nonlinear aircraft model defined by (2.21). The minus sign in $b_{ti}(t)$ and $b_{pi}(t)$ is to compensate the fact that, by convention, the scalar values $D_i(t)$ and $L_i(t)$ are positive when the vectors $\boldsymbol{D}_i(t)$ and $\boldsymbol{L}_i(t)$ has opposite direction to, respectively, $\boldsymbol{x}_{wi}$ and $\boldsymbol{z}_{wi}$ basis vectors from the S_{wi} (see Fig. 2.1b). The disturbance has higher amplitude and lower frequency when compared to the used in Subsection 4.6.1. The higher magnitude helps to highlight the difference between the robust and non robust approaches, and the lower frequency is more appropriate to the dynamics of the inner loop controller. The constants $\eta_{xi}, \eta_{yi}, \eta_{zi}$ are changed to 2.5 instead of 1, since in this example the magnitude of the disturbance is grater.

As can be seen in Fig. 4.10, the robust controller presents increased performance compared to the non-robust variant. However, by evaluating the actuator response in Figs. 4.11 and 4.12, it can be seen that a significant chattering is present. In Fig. 4.11, it is shown the output of the robust outer loop controller for the north and east directions. The vertical command is omitted, since the aircraft altitude is maintained constant by a linear controller. It can be seen that there is chattering. It also can be seen that there is a small frequency and amplitude sinusoidal component in the output, that compensates the sinusoidal disturbance. In Fig. 4.12, it is shown the actuator response from both UAVs using the robust controller. It can be seen in the aileron, rudder and throttle that there is a high frequency and a small frequency sinusoidal components.

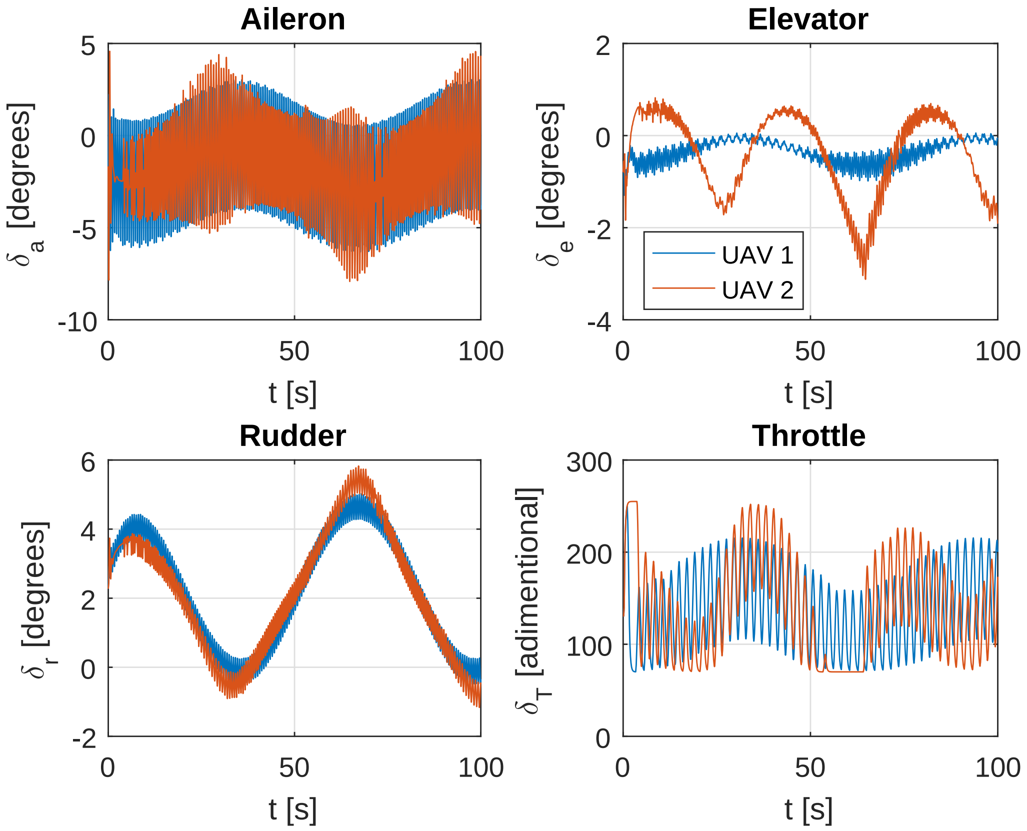


Figure 4.12: Actuator response from Proposition II.

The high frequency is the undesired chattering. The small frequency is the disturbance component being compensated. The elevator has a smaller chattering, because it is being commanded by a linear and non-robust outer loop controller.

4.7 CONCLUSION

The S_{NED}-NLDI controller presented in Chapter 3 is made robust in this chapter by adding the LPF SMC and the synchronous approaches to it. These additions results in two proposed control architectures. The proposed robust controllers are similar, but the second is mathematically and computationally simpler, and presented faster transient response.

It can be seen that the formation error asymptotically approaches zero in the simplified double integrator simulation. The formation error converges to a small error in the more realistic simulation that uses the nonlinear aircraft model plus the inner loop described in Chapter 2.

The communication graph component of the robust controller provides a way to maintain the formation shape more or less aggressively than the individual position of the aircraft, meaning that an UAV can temporarily depart from its desired position to approach other UAVs that are significantly away from the desired position. This can be useful in situations such as autonomous aerial refueling, in which the relative distance between the aircraft is more important than the position of each one related to the ground.

The low pass filter component of the SMC removes almost all the chattering in the simplified simulation, but not in the more realistic simulation. The main reason can be the inner loop, which does not respond fast enough. The outer loop is projected under the hypothesis that the inner loop dynamics can be neglected, which is necessary to the system to behave as the point-mass model. However, by definition, the sliding mode controller has very fast dynamics, even when including the low pass filter. The fast dynamics is necessary to perfectly track fast changing unknown disturbances.

There are two solutions that can reduce the chattering. The first one is to improve the inner loop speed convergence by, for example, choosing a more sophisticated controller. Increasing the inner loop bandwidth increases the allowed bandwidth in which the outer loop can operate.

The second one is to chose a $\boldsymbol{\eta}_i$ with smaller magnitude. This reduces reduces the aggressiveness of the controller, because it provides a $u_{xbi}(t)$ (Eq. (4.44c)) or $u_{xi}(t)$ (Eq. (4.60)) with smaller amplitude, which results in a $\dot{\tau}_{xi}(t)$ (Eq. (4.44d)) or $\dot{\tau}_{xbi}(t)$ (Eq. (4.59)) with smaller amplitude. This means that the control signal varies with less intensity, presenting a smoother control signal.

Reducing $\|\boldsymbol{\eta}_i\|$ can be made by two distinct approaches. The first is to try increase the modeling precision and to measure part of the disturbances. Decreasing the uncertainty decreased the required $\|\boldsymbol{\eta}_i\|$ that still provides fully disturbance rejection capability. A second approach is to give up rejecting the occasional extreme magnitude disturbance, and choosing $\|\boldsymbol{\eta}_i\|$ big enough to reject most of the disturbances, but not all of them.

A significant source of uncertainty is caused by the fact that the disturbance is estimated in the S_{NED} frame, but it is originated mostly in the $S_{\chi i}$ frame. For example, in (4.16), each component of the S_{NED}-described disturbance $\boldsymbol{d}_i(t)$ is bounded by the magnitude of the whole $S_{\chi i}$-described $\boldsymbol{b}_i(t)$, which is a very pessimistic approach. It can be seen that, besides $\|\boldsymbol{d}_i(t)\| = \|\boldsymbol{b}_i(t)\|$ (Eq. (4.16)) and being $\boldsymbol{\Delta}_i$ and $\boldsymbol{\delta}_i$ the upper bounds of, respectively, $\boldsymbol{d}_i(t)$ and $\boldsymbol{b}_i(t)$ (Eqs. (4.14) and (4.12)), by using (4.16), it is obtained

$$\|\boldsymbol{\Delta}_i\| = \sqrt{3}\|\boldsymbol{\delta}_i\|. \tag{4.84}$$

It is possible that the uncertainty can be reduced by implementing the LPF SMC in the $S_{\chi 0}$-NLDI variant, because the uncertainty would be described in $S_{\chi 0}$ frame. It is not the $S_{\chi i}$ frame in which the disturbance is described but, in usual formation flight, they are similar enough, since the UAVs usually flies to similar directions. Other approach would be try to develop a controller in which the disturbance rejection mechanism is placed after the NLDI. In this case, the rejection is made in the $S_{\chi i}$ frame. Both approaches can be evaluated as future work.

5 CONCLUSIONS

As the main contribution of this work, three leader-follower autonomous formation flight controllers are developed. As the first step of the development, all of the approaches use the nonlinear dynamic inversion of one of two possible formation flight modeling to achieve a linear double integrator model. Then, the obtained model can be controlled by a linear or by the nonlinear sliding mode control techniques. The stability of all approaches are mathematically demonstrated and simulations using a realistic nonlinear aircraft model demonstrate the feasibility of the proposed controllers. Others contributions include the comparison between two formation flight modeling styles and the development of a proof of concept path planner that shows the formation flight, when described in a leader's frame, can use simpler path planners. The main conclusions and contributions are detailed below.

Chapter 3 shows, by reviewing the literature, that there are two main groups of formation flight modeling styles. The first one describe all the UAVs positions and movements in a inertial or global frame as, for example, the S_{NED}. The second approach describes all positions and movements of the followers UAVs in a leader's frame, such as $S_{\chi 0}$ or the S_{sw0}. Applying the NLDI in each approach generate a distinct controller. It is proposed a new controller based in the $S_{\chi 0}$ frame approach, which performs better than an approach from literature when is a variable shape formation or the fleet maneuvers. Both the proposed approach and a controller from literature based in the S_{NED} perform similarly, but the proposed approach can be better tuned, because the dynamics of the outer loop can be better related to the dynamics of the inner loop and actuators.

In Chapter 4, it is shown that a NLDI obtained controller can be made robust to model uncertainties and disturbances by the SMC technique. The SMC generates a chattering output, which is undesired. A possible solution, evaluated in this work, is the use of a variant of the SMC that includes a low pass filter, which smooths the original chattering signal, providing a continuous output. The robustness of the formation shape can be increased by the use of the synchronous formation flight, in which the relative errors between the followers UAVs are also included in the control law.

The inertial frame S_{NED}-NLDI variant is chosen to be improved by the synchronous LPF SMC. The main reason is to compare to recent works that use this approach. However, the developed controllers can be applied in the leader's frame variant without significant modifications.

In the development of the proposed controller, first a recent AFF that uses the synchronous LPF SMC is reviewed. It is shown that this controller, proposed in [14], has problems, which results in the local controllers of each follower requiring data from outside its neighborhood. The first robust controller proposed in this thesis correct these problems. A second proposition is made, in which the original controller is simplified without compromising the performance. The stability of both controllers are proven by Lyapunov functions candidates. It is shown that the controllers achieve the sliding surface in finite time and, after, behave as a linear system that asymptotically converges to null error. The controller from literature and the two proposed controllers are simulated in a simplified scenario, which assumes that each UAV behaves exactly as double integrators. This shows that all controllers perform as expected. It worth to note that, however, only the proposed controllers are consistent with the given communication graph.

5.1 FUTURE WORKS

There are several theoretical and practical work that can be developed from this thesis:

- To adapt the robust controllers developed in Chapter 4 to the $S_{\chi 0}$ approach presented in Chapter 3. It is expected that, by adjusting distinct dynamics to the forward and lateral movements, and by describing the disturbances in a frame that is similar to the follower's frame, the controller performance wold be increased.
- To develop a S_{NED}-NLDI based LPF SMC that applies the low pass filter in the followers frame instead of the global frame. In this case, the filtered control signal is generated where the disturbance and actuators operate. This would reduce the model uncertainty by removing the rotating component of the uncertainty calculation in (4.19).
- To evaluate other 2nd order SMC, such as the super twisting approach [60] or nonlinear sliding surfaces that, for example, provides finite time convergence of the error [41].
- To implement a better inner loop controller, with faster response time. It is assumed that a better inner loop would reduce the chattering presented in the realistic simulation.
- To implement an inner loop that converts vertical acceleration to elevator command. This would allow a realistic 3D simulation of the robust controllers.
- To evaluate non-UAV state-of-art path planners in the formation flight described in the leader's frame.
- To evaluate other communication graphs, such as directed, time-varying or with communication delay.
- To evaluate the effect of sensor noise.
- To implement the proposed controllers in real model aircraft.

5.2 SCIENTIFIC PUBLICATIONS

Partial results of this thesis were published in scientific conferences. The published work is listed below:

- [3] CORDEIRO, T. F. K.; FERREIRA, H. C.; ISHIHARA, J. Y. Non linear controller and path planner algorithm for an autonomous variable shape formation flight. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS). [S.l.: s.n.], 2017. p. 1493-1502.
- [59] CORDEIRO, T. F. K.; FERREIRA, H. C.; ISHIHARA, J. Y. Controle Não Linear de Voo em Formação de Veículos Aéreos Não Tripulados de Asa Fixa. In: XIII Simpósio Brasileiro de Automação Inteligente (SBAI). [S.l.: s.n.], 2017. p. 1480-1485.

There is also a conference and a journal publications that were published during the PhD as a final evaluation exam of two disciplines taken during the PhD. They are listed below.

- [61] CORDEIRO, T. F. K. et al. Kalman-based attitude estimation for an UAV via an antenna array. In: Signal Processing and Communication Systems (ICSPCS), 2014 8th International Conference on. [S.l.: s.n.], 2014. p. 1-10.
- [52] CORDEIRO, T. F. K. et al. Improved Kalman-based attitude estimation framework for UAVs via an antenna array. Digital Signal Processing, Elsevier, v. 59, p. 49-65, 2016.

BIBLIOGRAPHY

[1] Agência Nacional de Aviação Civil. *Resolução n^o 419, de 2 de maio de 2017, Requisitos Gerais para Aeronaves Não Tripuladas de Uso Civil.* Disponível em: <http://www.anac.gov.br/assuntos/legislacao/legislacao-1/rbha-e-rbac/rbac/rbac-e-94-emd-00>.

[2] WIKIPEDIA. *Unmanned combat aerial vehicle*. [Online; accessed 04-November-2018]. Disponível em: <https://en.wikipedia.org/wiki/Unmanned_combat_aerial_vehicle>.

[3] CORDEIRO, T. F. K.; FERREIRA, H. C.; ISHIHARA, J. Y. Non linear controller and path planner algorithm for an autonomous variable shape formation flight. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.: s.n.], 2017. p. 1493–1502.

[4] WILSON, D. B. *Guidance, Navigation and Control for UAV Close Formation Flight and Airborne Docking*. Tese (Doutorado) — University of Sydney, 2015.

[5] MACHARET, D. G. et al. Efficient target visiting path planning for multiple vehicles with bounded curvature. In: IEEE. *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. [S.l.], 2013. p. 3830–3836.

[6] RASMUSSEN, S. J.; SHIMA, T. Tree search algorithm for assigning cooperating uavs to multiple tasks. *International Journal of Robust and Nonlinear Control*, Wiley Online Library, v. 18, n. 2, p. 135–153, 2008.

[7] HAN, J.; CHEN, Y. Multiple uav formations for cooperative source seeking and contour mapping of a radiative signal field. *Journal of Intelligent & Robotic Systems*, Springer, v. 74, n. 1-2, p. 323–332, 2014.

[8] KOLLING, A.; KLEINER, A.; RUDOL, P. Fast guaranteed search with unmanned aerial vehicles. In: IEEE. *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. [S.l.], 2013. p. 6013–6018.

[9] SHAFERMAN, V.; SHIMA, T. Unmanned aerial vehicles cooperative tracking of moving ground target in urban environments. *Journal of guidance, control, and dynamics*, v. 31, n. 5, p. 1360–1371, 2008.

[10] TAL, E.; SHIMA, T. Differential games based autonomous rendezvous for aerial refueling. In: *Advances in Aerospace Guidance, Navigation and Control*. [S.l.]: Springer, 2015. p. 167–185.

[11] CAMPA, G. et al. Design and flight-testing of non-linear formation control laws. *Control Engineering Practice*, Elsevier, v. 15, n. 9, p. 1077–1092, 2007.

[12] LEE, D.; KIM, S.-K.; SUK, J. Design of a track guidance algorithm for formation flight of UAVs. In: *AIAA Guidance, Navigation, and Control Conference*. [S.l.: s.n.], 2015. p. 1315.

[13] SINGH, S. N. et al. Decentralized nonlinear robust control of UAVs in close formation. *International Journal of Robust and Nonlinear Control*, Wiley Online Library, v. 13, n. 11, p. 1057–1078, 2003.

[14] XIANG, X. et al. On decentralized adaptive full-order sliding mode control of multiple UAVs. *ISA Transactions*, v. 71, p. 196 – 205, 2017. ISSN 0019-0578.

[15] REZAEE, H.; ABDOLLAHI, F.; TALEBI, H. A. $\mathcal{H}_\infty$ based motion synchronization in formation flight with delayed communications. *IEEE Transactions on Industrial Electronics*, IEEE, v. 61, n. 11, p. 6175–6182, 2014.

[16] VACHON, M. J. et al. F/A-18 aircraft performance benefits measured during the autonomous formation flight project. In: *AIAA Atmospheric Flight Mechanics Conference and Exhibit*. [S.l.: s.n.], 2002.

[17] DIBLEY, R.; ALLEN, M.; NABAA, N. Autonomous airborne refueling demonstration phase i flight-test results. In: *AIAA Atmospheric Flight Mechanics Conference and Exhibit*. [S.l.: s.n.], 2007. p. 6639.

[18] NICHOLS, J. W. et al. Aerial rendezvous of small unmanned aircraft using a passive towed cable system. *Journal of Guidance, Control, and Dynamics*, American Institute of Aeronautics and Astronautics, v. 37, n. 4, p. 1131–1142, 2014.

[19] HADAEGH, F. Y.; LU, W.-M.; WANG, P. K. Adaptive control of formation flying spacecraft for interferometry. *IFAC Proceedings Volumes*, Elsevier, v. 31, n. 20, p. 117–122, 1998.

[20] REN, W.; BEARD, R. W. Formation feedback control for multiple spacecraft via virtual structures. *IEE Proceedings-Control Theory and Applications*, IET, v. 151, n. 3, p. 357–368, 2004.

[21] OH, K.-K.; PARK, M.-C.; AHN, H.-S. A survey of multi-agent formation control. *Automatica*, Elsevier, v. 53, p. 424–440, 2015.

[22] YU, J. et al. Time-varying formation tracking for high-order multi-agent systems with switching topologies and a leader of bounded unknown input. *Journal of the Franklin Institute*, Elsevier, v. 355, n. 5, p. 2808–2825, 2018.

[23] LIU, Z. et al. Fault-tolerant formation control of unmanned aerial vehicles in the presence of actuator faults and obstacles. *Unmanned Systems*, World Scientific, v. 4, n. 03, p. 197–211, 2016.

[24] ZHAO, D.; LI, C.; ZHU, Q. Low-pass-filter-based position synchronization sliding mode control for multiple robotic manipulator systems. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, SAGE Publications Sage UK: London, England, v. 225, n. 8, p. 1136–1148, 2011.

[25] LI, Z.; XING, X.; YU, J. Decentralized output-feedback formation control of multiple 3-DOF laboratory helicopters. *Journal of the Franklin Institute*, Elsevier, v. 352, n. 9, p. 3827–3842, 2015.

[26] HAN, T. et al. Three-dimensional containment control for multiple unmanned aerial vehicles. *Journal of the Franklin Institute*, Elsevier, v. 353, n. 13, p. 2929–2942, 2016.

[27] GHAPANI, S. et al. Fully distributed flocking with a moving leader for Lagrange networks with parametric uncertainties. *Automatica*, Elsevier, v. 67, p. 67–76, 2016.

[28] SINGH, S. N. et al. Input–output invertibility and sliding mode control for close formation flying of multiple uavs. *International Journal of Robust and Nonlinear Control*, Wiley Online Library, v. 10, n. 10, p. 779–797, 2000.

[29] YU, X.; LIU, Z.; ZHANG, Y. Fault-tolerant formation control of multiple UAVs in the presence of actuator faults. *International Journal of Robust and Nonlinear Control*, Wiley Online Library, 2015.

[30] WILBURN, B. K.; PERHINSCHI, M. G.; WILBURN, J. N. A modified genetic algorithm for UAV trajectory tracking control laws optimization. *International Journal of Intelligent Unmanned Systems*, Emerald Group Publishing Limited, v. 2, n. 2, p. 58–90, 2014.

[31] PALUMBO, N. F.; BLAUWKAMP, R. A.; LLOYD, J. M. Basic principles of homing guidance. *Johns Hopkins APL Technical Digest*, v. 29, n. 1, p. 25–41, 2010.

[32] WATANABE, Y. *Stochastically optimized monocular vision-based navigation and guidance*. Tese (Doutorado) — Georgia Institute of Technology, 2008.

[33] YAMASAKI, T.; BALAKRISHNAN, S. Sliding mode based pure pursuit guidance for uav rendezvous and chase with a cooperative aircraft. In: IEEE. *American Control Conference (ACC), 2010*. [S.l.], 2010. p. 5544–5549.

[34] ELKAIM, G. H.; LIE, F. A. P.; GEBRE-EGZIABHER, D. Principles of guidance, navigation, and control of UAVs. In: *Handbook of Unmanned Aerial Vehicles*. [S.l.]: Springer, 2015. p. 347–380.

[35] SU, Z. et al. Exact docking flight controller for autonomous aerial refueling with back-stepping based high order sliding mode. *Mechanical Systems and Signal Processing*, Elsevier, v. 101, p. 338–360, 2018.

[36] CURRY, R. et al. L_2^+, an improved line of sight guidance law for UAVs. In: IEEE. *American Control Conference (ACC), 2013*. [S.l.], 2013. p. 1–6.

[37] WILSON, D. B.; GÖKTOGAN, A.; SUKKARIEH, S. Guidance and navigation for UAV airborne docking. In: *Robotics: Science and Systems*. [S.l.: s.n.], 2015.

[38] LIN, W. Distributed UAV formation control using differential game approach. *Aerospace Science and Technology*, Elsevier, v. 35, p. 54–62, 2014.

[39] CAMPA, G. et al. Design of formation control laws for manoeuvred flight. *The Aeronautical Journal*, Cambridge University Press, v. 108, n. 1081, p. 125–134, 2004.

[40] SUMANTRI, B.; UCHIYAMA, N.; SANO, S. Least square based sliding mode control for a quadrotor helicopter and energy saving by chattering reduction. *Mechanical Systems and Signal Processing*, Elsevier, v. 66, p. 769–784, 2016.

[41] CHONG, S. et al. An improved chattering-free sliding mode control with finite time convergence for reentry vehicle. In: *IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*. [S.l.: s.n.], 2016. p. 69–74.

[42] SLOTINE, J.-J. E.; LI, W. *Applied nonlinear control*. [S.l.]: Prentice Hall Englewood Cliffs, NJ, 1991.

[43] PARK, K.-B.; LEE, J.-J. Sliding mode controller with filtered signal for robot manipulators using virtual plant/controller. *Mechatronics*, Elsevier, v. 7, n. 3, p. 277–286, 1997.

[44] BALAMURUGAN, S.; VENKATESH, P.; VARATHARAJAN, M. Fuzzy sliding-mode control with low pass filter to reduce chattering effect: an experimental validation on Quanser SRIP. *Sādhanā*, Springer, v. 42, n. 10, p. 1693–1703, 2017.

[45] PHUKAN, S.; MAHANTA, C. Position synchronization control of multiple robotic manipulator systems using low pass filter based integral sliding mode. In: *15th International Workshop on Variable Structure Systems and Sliding Mode Control (VSS 2018)*. [S.l.: s.n.], 2018.

[46] DUBINS, L. E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, JSTOR, v. 79, n. 3, p. 497–516, 1957.

[47] SHANMUGAVEL, M. et al. Co-operative path planning of multiple UAVs using dubins paths with clothoid arcs. *Control Engineering Practice*, Elsevier, v. 18, n. 9, p. 1084–1092, 2010.

[48] LIN, Y.; SARIPALLI, S. Path planning using 3d dubins curve for unmanned aerial vehicles. In: IEEE. *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. [S.l.], 2014. p. 296–304.

[49] ANDERSON JR, J. D. *Introduction to Flight*. 3. ed. [S.l.]: McGraw-Hill Book Company, NY, 1989. ISBN 0-07-001641-0.

[50] STEVENS, B. L.; LEWIS, F. L. *Aircraft Control and Simulation*. [S.l.]: John Wilew & Sons, Inc, NY, 1992. ISBN 0-471-61397-5.

[51] ETKIN, B. *Dynamics of atmospheric flight*. [S.l.]: Courier Corporation, 2012.

[52] CORDEIRO, T. F. K. et al. Improved Kalman-based attitude estimation framework for UAVs via an antenna array. *Digital Signal Processing*, Elsevier, v. 59, p. 49–65, 2016.

[53] GU, Y. et al. *Autonomous formation flight: Design and experiments*. [S.l.]: INTECH Open Access Publisher, 2009.

[54] PAW, Y. C. *Synthesis and validation of flight control for UAV*. Tese (Doutorado) — University of Minnesota, 2009.

[55] MENON, P.; SWERIDUK, G.; SRIDHAR, B. Optimal strategies for free-flight air traffic conflict resolution. *Journal of Guidance, Control, and Dynamics*, v. 22, n. 2, p. 202–211, 1999.

[56] AZAM, M.; SINGH, S. N. Invertibility and trajectory control for nonlinear maneuvers of aircraft. *Journal of Guidance, Control, and Dynamics*, v. 17, n. 1, p. 192–200, 1994.

[57] WILSON, D. B. et al. Real-time rendezvous point selection for a nonholonomic vehicle. In: IEEE. *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. [S.l.], 2013. p. 3941–3946.

[58] OWEN, M.; BEARD, R. W.; MCLAIN, T. W. Implementing dubins airplane paths on fixed-wing uavs. In: *Handbook of Unmanned Aerial Vehicles*. [S.l.]: Springer, 2015. p. 1677–1701.

[59] CORDEIRO, T. F. K.; FERREIRA, H. C.; ISHIHARA, J. Y. Controle não linear de voo em formação de veículos aéreos não tripulados de asa fixa. In: *XIII Simpósio Brasileiro de Automação Inteligente (SBAI)*. [S.l.: s.n.], 2017. p. 1480–1485.

[60] SHTESSEL, Y. et al. *Sliding Mode Control and Observation*. [S.l.]: Birkhäuser Basel, 2014. ISBN 978-0-8176-4893-0.

[61] CORDEIRO, T. F. K. et al. Kalman-based attitude estimation for an UAV via an antenna array. In: *Signal Processing and Communication Systems (ICSPCS), 2014 8th International Conference on*. [S.l.: s.n.], 2014. p. 1–10.

Appendix

A. SIMULATION IMPLEMENTATION DETAILS

Here are included details about how the aircraft model is implemented in the Simulink toolbox from MATLAB and how the trimming is obtained.

A.1 AIRCRAFT MODEL IMPLEMENTATION

The aircraft model is implemented in the Simulink toolbox from MATLAB 2013b. The nonlinear equations (2.21) are calculated by using a *MATLAB function* block, and the results are integrated by using *Integrator* blocks. The Simulink solver is configured to use a fixed-step of 0.01 s, and to use the Runge-Kutta (ODE45) numerical integrator. The integrator blocks are initialized with values obtained from a previously executed trim algorithm that sets the aircraft to fly a straight level flight.

The surface actuators and engine are implemented including their dynamic model and delay, and respecting the lower and upper limits, as shown in Fig. A.1. The commanded surface position $\delta_{\text{commanded}}$ and the commanded throttle values are limited by a saturation block before being applied in the actuator model, to prevent an unrealistic fast response generated by an unrealistic high input. The transfer functions (2.17) and (2.15) are redesigned as a feedback of an integrator block instead of a simple transfer function, because the integrator block can be initialized to the trim conditions of the actuator. The integrator limits the output to adequate bounds.

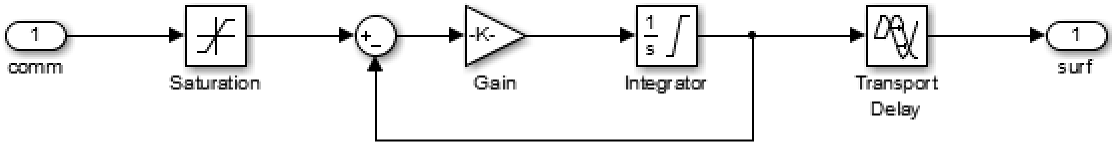


Figure A.1: Actuator dynamic model in Simulink. Gain K = $1/\tau_a$ (control surfaces) or $1/\tau_T$ (engine). The *Transport Delay* block inserts a $t_{d,a}$ or $t_{d,T}$ delay.

Several extra outputs are calculated from the state vector and/or its derivative. These extra outputs are used 1) as input of controllers, or 2) to help evaluate the control or estimator performance. It is worth to note that these equations are not necessary meant to be used in real scenarios with noisy measurements or estimations, but to be used to generate precise simulated outputs generated from the assumed perfect state space and the state space derivative equations.

The velocity vector in body frame $\boldsymbol{V}_{bi}(t)$

$$\boldsymbol{V}_{bi}(t) = \boldsymbol{D}_{bi}^{wi}(t)\boldsymbol{V}_{wi}(t) = \begin{bmatrix} \cos\alpha_i(t)\cos\beta_i(t) \\ \sin\beta_i(t) \\ \sin\alpha_i(t)\cos\beta_i(t) \end{bmatrix} V(t). \tag{A.1}$$

The inertial acceleration vector $\bar{\boldsymbol{a}}_{bi}(t)$ measured in body frame, i.e., the second derivative of the position described in S_{NED}, represented in the body frame is calculated by first calculating the airspeed derivative represented in the S_{bi} frame $[\boldsymbol{a}_{wi}]_{bi}(t)$. After, the Theorem of Coriolis (2.12) is used twice. First to obtain $\dot{\boldsymbol{V}}_{bi}(t)$, which removes the effect of the rotation $\boldsymbol{\omega}^{wi,bi}(t)$ between S_{wi} and S_{bi} that is present in $[\boldsymbol{a}_{wi}]_{bi}(t)$. After, to obtain the inertial acceleration $\bar{\boldsymbol{a}}_{bi}(t)$, which removes the effect of rotation $\boldsymbol{\omega}_{bi}^{bi,\text{NED}}(t)$ = $[P(t)$ $Q(t)$ $R(t)]^T$ between S_{bi} and S_{NED}.

$$[\boldsymbol{a}_{wi}]_{bi}(t) = \boldsymbol{D}_{bi}^{wi}(t)\dot{\boldsymbol{V}}_{wi}(t) = \begin{bmatrix} \cos\alpha_i(t)\cos\beta_i(t) \\ \sin\beta_i(t) \\ \sin\alpha_i(t)\cos\beta_i(t) \end{bmatrix} \dot{V}_i(t), \tag{A.2a}$$

$$\dot{\boldsymbol{V}}_{bi}(t) = [\boldsymbol{a}_{wi}]_{bi}(t) - \boldsymbol{\omega}^{wi,bi}(t) \times \boldsymbol{V}_{bi}(t), \tag{A.2b}$$

$$\bar{\boldsymbol{a}}_{bi}(t) = \dot{\boldsymbol{V}}_{bi}(t) - \boldsymbol{\omega}_{bi}^{bi,\text{NED}}(t) \times \boldsymbol{V}_{bi}(t), \tag{A.2c}$$

where $\boldsymbol{\omega}^{wi,bi}(t)$ is calculated as

$$\boldsymbol{\omega}^{wi,bi}(t) = \begin{bmatrix} \dot{\beta}_i(t)\sin\alpha_i(t) & \dot{\alpha}_i(t) & -\dot{\beta}_i(t)\cos\alpha_i(t) \end{bmatrix}^T. \tag{A.3}$$

The accelerometer measured acceleration $\boldsymbol{a}_{bi}$ does not include the gravity field generated acceleration, which is represented by $\boldsymbol{g}_{\text{NED}}$ = [0 0 g_0]^T in S_{NED}

$$\boldsymbol{a}_{bi}(t) = \bar{\boldsymbol{a}}_{bi}(t) - \boldsymbol{D}_{bi}^{\text{NED}}(t)\boldsymbol{g}_{\text{NED}}. \tag{A.4}$$

The velocity vector represented in S_{NED}, $\boldsymbol{V}_{\text{NED},i}(t)$, is obtained directly from the state space equations

$$\boldsymbol{V}_{\text{NED},i}(t) \triangleq [V_{N,i}(t)\ V_{E,i}(t)\ V_{D,i}(t)]^T = [\dot{x}_i(t)\ \dot{y}_i(t)\ \dot{z}_i(t)]^T. \tag{A.5}$$

The S_{NED} described trajectory information ground speed $V_{gi}(t)$, course angle $\chi_i(t)$ and flight path angle $\gamma_i(t)$ are calculated by

$$V_{gi}(t) = \|\boldsymbol{V}_{\text{NED},i}(t)\|, \tag{A.6}$$

$$\chi_i(t) = \text{atan2}(V_{E,i}(t), V_{N,i}(t)), \tag{A.7}$$

$$\gamma_i(t) = \text{asin}(-V_{D,i}(t)/V_{gi}(t)), \tag{A.8}$$

where $\text{atan2}(a, b)$ is the 4-quadrant arc tangent. The time derivatives of χ and γ are also obtained

$$\dot{\chi}_i(t) = \frac{\bar{a}_{E,i}(t)V_{N,i}(t) - \bar{a}_{N,i}(t)V_{E,i}(t)}{V_{N,i}^2(t)\,(1 + (V_{E,i}(t)/V_{N,i}(t))^2)} = \frac{\bar{a}_{E,i}(t)V_{N,i}(t) - \bar{a}_{N,i}(t)V_{E,i}(t)}{V_{N,i}^2(t) + V_{E,i}^2(t)}, \tag{A.9}$$

$$\dot{\gamma}_i(t) = -\frac{\bar{a}_{D,i}(t)V_{gi}(t) - \dot{V}_{gi}(t)V_{D,i}(t)}{V_{gi}^2(t)\sqrt{1 - (V_{D,i}(t)/V_{gi})^2}}, \tag{A.10}$$

where

$$\bar{\boldsymbol{a}}_{\text{NED},i}(t) = [\bar{a}_{N,i}(t)\ \bar{a}_{E,i}(t)\ \bar{a}_{D,i}(t)]^T = \boldsymbol{D}_{\text{NED}}^{bi}(t)\bar{\boldsymbol{a}}_{bi}(t) = \boldsymbol{D}_{\text{NED}}^{bi}(t)\boldsymbol{a}_{bi}(t) + \boldsymbol{g}_E, \tag{A.11}$$

$$\dot{V}_{gi}(t) = \boldsymbol{V}_{\text{NED},i}(t) \cdot \bar{\boldsymbol{a}}_{\text{NED},i}/V_{gi}(t). \tag{A.12}$$

A.2 MODEL TRIMMING

When simulating the system in open-loop, i.e., without a control loop, it is desired to be able to simulate the system making simple maneuvers, such as a straight level flight. It is impossible to guess, by trial and error, the correct initial state and the correct input values that achieves these maneuvers. A solution is to implement an algorithm that tries several combinations of initial state space values and actuator values, adjusting these values based in the previous obtained results. This is a trimming algorithm, and it detailed in [50]. There is a pair of MATLAB functions, *operspec* and *findop*, that achieves the trimmed condition.

The *operspec* function receives a Simulink model and returns an object that is used to specify the desired operation point, i.e., the trim conditions. This object contains references to the inputs and states, where the states are all integrators found in the model. The used Simulink model is the one detailed in previous subsection, but removing the actuator dynamics and time delay. The actuator dynamics insert extra state and dynamics to the problem, which increase the difficult of obtaining a solution. The dynamics, however, does not change the steady state results of the system.

It must be provided an initial value to each input, and set a flag that indicates if the provided value is a guess or if it is a known, immutable input. All inputs are maintained constant during each simulation.

The state configuration is similar. It must be provided an initial value, which must be defined by a flag as a guess or known value. An extra flag defines the state as steady state, i.e., constant during simulation, or variable. As example, the initial position of the aircraft is known, but varies with the time, since the aircraft has non-null velocity.

The flags must be defined to describe the operating point, i.e., straight level flight, while avoiding to over restrict the state and inputs. An initial guess, obtained from [11], is provided to obtain a fast convergence of the algorithm. Table A.1 shows the configuration used.

Table A.1: Trim setup

Input	Guess	Known?	
δ_E	$-1°$	0	
δ_A	0	0	
δ_R	0	0	
δ_T	128.97 ($\approx 54.62\ N$)	0	
State	Guess	Known?	Steady state?
Airspeed	42 m/s	1	1
α, β	3, 0 (°)	0, 0	1, 1
p, q, r	0, 0, 0 (rad/s)	0, 0, 0	1, 1, 1
ϕ, θ, ψ	0, 3, 0 (°)	1, 0, 0	1, 1, 1
x_N, x_E, x_D	0, 0, -120 (m)	1, 1, 1	0, 1, 1

After configuring the input and states, the function *findop* is called and finds the trimmed state, returning the obtained input and state values.

B. RESUMO ESTENDIDO EM PORTUGUÊS

Figure B.1: Veículos aéreos não tripulados presentes no Laboratório de Robótica Aérea - UnB.

Veículos aéreos não tripulados (VANTs) são veículos que se sustentam no ar via reações aerodinâmicas e em que a pilotagem é feita de modo remoto ou completamente autônoma [1]. A definição inclui veículos de asa fixa, como aviões, e asa rotativa, como helicópteros e quadrirrotores.

Existe um interesse civil, militar e acadêmico bastante intenso no assunto. Como exemplo de interesse civil, no Brasil a Agência Nacional Aviação Civil (ANAC) publicou recentemente um regulamento especial (de caráter provisório) para aeronaves remotamente pilotadas de uso não-recreacional. Já ocorre o uso militar, seja para combate ou inteligência [2]. Do ponto de vista acadêmico, destaca-se como exemplo a bibliografia desse trabalho, que contém diversos materiais recentes sobre o assunto.

Um dos tópicos de interesse para VANTs são os sistemas multi agente, em que diversas aeronaves interagem entre si. Essa interação pode ser classificada como [4]

- **Cooperativa**: ambos os veículos manobram buscando atingir certo objetivo. Exemplos: uma frota tem que cumprir uma lista de tarefas no menor tempo possível, ou com menor consumo de combustível [5, 6].
- **Evasiva**: um veículo manobra ativamente para evitar interagir com o outro, ou manobra de forma brusca e imprevisível [10].
- **Não-cooperativa**: um dos veículos manobra de forma suave, mas sem cooperar ou evadir ativamente do objetivo. É o caso de muitas abordagens de formação líder-seguidor [11, 12, 13, 14, 15], em que um líder voa livremente, e seguidores mantém uma distância pré-determinada desse líder.

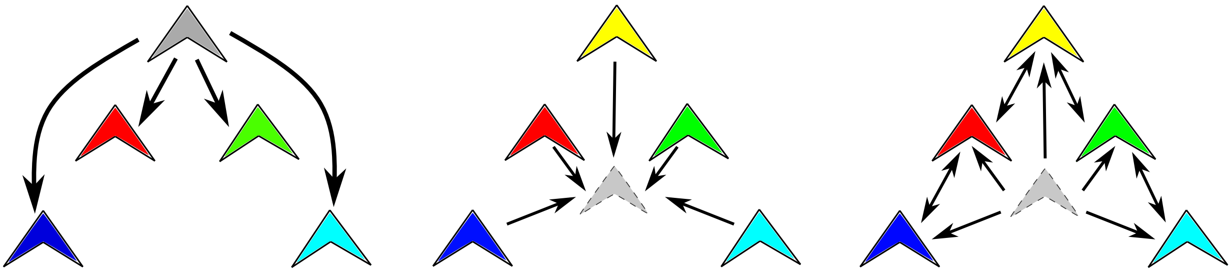
Ressalta-se que a classificação de grau de cooperação se refere às manobras especificamente. Por exemplo, uma aeronave não-cooperativa pode transmitir seus dados a aeronaves próximas, cooperando no sentido de fornecer informações.

B.1 VOO EM FORMAÇÃO AUTÔNOMO

Um caso específico de sistemas multi agente é o voo em formação. Dentre as motivações para o voo em formação incluem a redução do consumo de combustível da frota [13, 16], como uma etapa intermediária para interações físicas entre aeronaves, como reabastecimento aéreo [17, 4, 10] ou pouso e decolagem de pequenas aeronaves em uma aeronave mãe em pleno voo [18], ou para gerar uma rede móvel de sensores, como por exemplo a detecção de uma fonte radiativa (radiação sonora, luminosa ou radioativa por exemplo) via mensuração do gradiente de radiação [7].

Existem diversas abordagens para se obter a formação autônoma de agentes, que aplicam-se também ao voo em formação autônomo. Algumas delas estão ilustradas pela Fig. B.2 e são descritas abaixo [20, 21, 14, 15, 22]

- **Líder-seguidor** [11, 12, 23]: um veículo é definido como líder, e os outros como seguidores. O líder movimenta-se em uma trajetória que pode ou não ser influenciada pelos seguidores, ou seja, pode ser cooperativo ou não-cooperativo. Os seguidores devem rastrear uma trajetória definida em relação ao líder, de modo que o movimento dos seguidores é afetado pelo movimento do líder. Usualmente o líder transmite a todos (*broadcast*) suas informações. É uma das soluções mais simples de se implementar, e permite a descrição de formatos precisos.
- **Estrutura virtual / líder virtual** [20]: uma entidade virtual é definida como líder. Todos os veículos tem acesso a suas informações porque, por exemplo, a entidade é um software embarcado em cada veículo. Como vantagem, não exige a existência de um líder que consegue se comunicar com todos ao mesmo tempo. O projeto do sistema de controle da formação se faz de forma idêntica ao da abordagem líder-seguidor. A abordagem líder virtual também pode ser cooperativa ou não-cooperativa.
- **Síncrona** [24, 15, 14]: generalização da abordagem líder-seguidor (ou líder virtual), em que seguidores comunicam-se entre si além de receber informações do líder. Como objetivo de controle, buscam não só se posicionar em relação ao líder (virtual ou não), mas também em relação uns aos outros. Ao controlarem as posições relativas entre si, as aeronaves trabalham em modo cooperativo, em que as aeronaves trabalham juntas para manter o formato correto da formação, mesmo que o conjunto de veículos, como um todo, esteja na posição incorreta.



(a) Líder-seguidor

(b) Líder virtual

(c) Síncrono com líder virtual

Figure B.2: Algumas estilos de formação baseados em líder.

A formação também pode ser classificada como formato fixo ou variável no tempo. O segundo caso, mais geral, permite que haja movimento entre os agentes.

Por fim, o projeto do controle de formação também depende do sistema cartesiano de referência utilizado para descrever o problema [21]. Para aeronaves em interação não evasivo, são relevantes principalmente a abordagem por posição e abordagem por deslocamento. Na abordagem por posição, cada aeronave sabe sua posição em relação a um referencial global, obtido por um receptor de GPS, por exemplo, e a trajetória que a aeronave deve seguir também é descrita em relação a esse referencial global. Na abordagem por deslocamento, as aeronaves conhecem apenas as distâncias relativas entre si, sendo essa distância um vetor descrito em um referencial comum, como por exemplo, o do líder.

B.2 ARQUITETURA DE CONTROLE DA AERONAVE

Um sistema de controle permite que um VANT faça manobras, voe em direção de *waypoints* ou siga trajetórias desejados, de forma automática. No voo em formação autônomo, é o sistema de controle quem mantém a formação.

É comum implementar o sistema de controle de um VANT separado em dois ou três laços encadeados [11, 4, 28, 14], o laço interno, o laço externo e o planejador de trajetórias. Laços mais internos simplificam a dinâmica para os laços mais externos, funcionando como camadas de abstração, sendo que o laço interno (*inner loop*) age diretamente com a aeronave e seus atuadores.

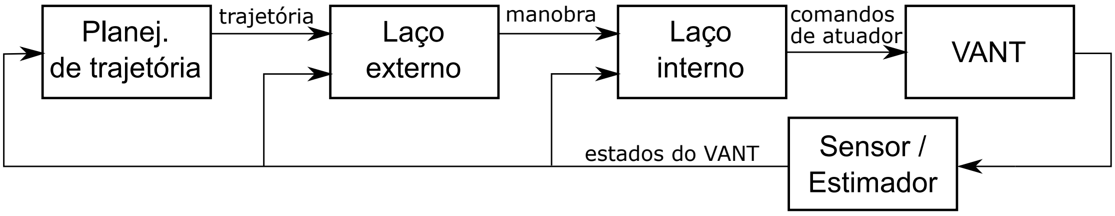


Figure B.3: Três laços encadeados controlando um VANT.

Sendo o foco da tese o projeto de controladores para o voo em formação, e este controle ser feito principalmente pela malha externa (*outer loop*), apenas essa malha é discutida aqui.

B.2.1 Controlador de malha externa

A malha externa recebe como entrada uma trajetória desejada descrita, por exemplo, como uma posição e/ou velocidade desejadas no atual instante de tempo. Avaliando o erro entre a trajetória desejada e a atual, a malha interna gera comandos de manobras para a malha interna. Assume-se, para simplificar o projeto, que a malha interna é capaz de responder instantaneamente às manobras prescritas. Apesar de não realista, essa simplificação funciona bem se a malha interna consegue realizar as manobras significativamente mais rápido do que a malha externa modifica a manobra comandada.

Dentre as diversas aplicações da malha externa, destaca-se nesse trabalho o uso para voo em formação autônomo.

B.3 VOO EM FORMAÇÃO AUTÔNOMO

Pode-se modelar o problema do voo em formação de diversas formas. Nessa tese, agruparam-se os diversos controladores em duas categorias: abordagem do sistema de referência inercial e abordagem do sistema de referência do líder.

Na abordagem do sistema de referência inercial, cada VANT segue uma trajetória que é descrita no sistema inercial. Dessa forma, o problema fica bastante similar ao problema de rastreamento de trajetórias, o que permite aproveitar soluções já existentes para esse caso. Destaca-se que a trajetória continua sendo afetada em tempo real pelo líder. Essa abordagem exige o uso de sensores de posição (GPS) de alta precisão e de uma sincronia na medição do tempo entre as aeronaves.

Na abordagem do sistema de referência do líder, a trajetória é descrita em algum referencial do líder, de forma que a trajetória rotaciona junto do líder. É uma abordagem de posição relativa, o que dispensaria o uso de receptor GPS nas aeronaves seguidoras, trocando-o por algum sensor que meça distância relativa, como por exemplo uma câmera de vídeo cujos dados são tratados por algum algoritmo de visão computacional.

Independentemente da abordagem escolhida, o voo em formação autônomo pode ser visto como um sistema não-linear. Uma abordagem possível para o projeto de um controlador para voo em formação é utilizar controladores lineares, como o PID (proporcional-integral-derivativo), controlando diretamente o sistema não linear [12, 39]. Entretanto, o desempenho de controladores lineares depende do sistema se manter perto de um ponto de operação.

Outra possibilidade é o uso de uma técnica, chamada de inversão não linear, antes de efetuar o controle, que consiste em aplicar uma transformação num modelo não linear para deixá-lo linear. Destaca-se que o sistema obtido é realmente linear, diferentemente da linearização via matrizes Jacobianas, que apenas encontra um modelo linear que se comporta de modo similar ao não-linear em torno de um ponto de operação. Destaca-se que alguns efeitos, como saturação de atuadores, não podem ser compensados pela inversão não-linear, sendo que nesses casos a transformação linear deixa de funcionar perfeitamente.

Transformando o modelo em linear, qualquer técnica de controle pode ser utilizada, mesmo lineares, sendo que o desempenho e estabilidade não são degradados ao se afastar do ponto de operação. Alguns exemplos de uso de inversão não-linear seguida de outro controlador: alocação de polos [11], controle linear distribuído [26] controle projetado via teoria de jogos [38] e controlador robusto $\mathcal{H}_\infty$ para formação síncrona [15].

A inversão não linear, entretanto, não provê um modelo perfeitamente não linear se existirem incertezas de modelo. O controle por modos deslizantes é capaz de prover o desempenho projetado mesmo se a linearização é inexata, sendo robusto tanto a incertezas de modelo quanto perturbações limitadas. Exemplos de uso dos modos deslizantes (ou variantes) em formação incluem trabalhos com aeronaves de asa fixa [13, 14], asa rotativa [25], e uma abordagem que não especifica veículo [22]. Para sinstemas de único agente, pode-se citar [40, 41].

Uma desvantagem da versão original do controle por modos deslizantes é que o sinal de controle varia bruscamente, chaveando entre dois valores bastante diferentes. Esse efeito é chamado de *chattering*, e aparece matematicamente na lei de controle como uma função *sinal* (*sign*) [13, 22].

Existem diversas técnicas de atenuar esse efeito de *chattering*, mas todas envolvem algum compromisso. Por exemplo, pode-se abrir mão da precisão e trocar a função *sinal* pela função *saturação* [42, 25, 40] ou tangente hiperbólica [27]. Outra técnica é chamada de modolos deslizantes de segunda ordem, em que se adiciona um integrador entre o controlador e a planta. O integrador atenua o *chattering*, de forma que a saída do integradfor é um sinal contínuo. Mas, por outro lado, deve-se assumir que a derivada da perturbação tem magnitude limitada, suposição que não era necessária na abordagem original. Uma generalização dos modos deslizantes de segunda ordem é a abordagem filtro passa baixas, que utiliza um filtro de primeira ordem ao invés de um integrador [43, 24, 41, 44, 14, 45].

B.4 DEFINIÇÃO DO PROBLEMA

Existem ao menos dois VANTs voando juntos, e eles precisam realizar uma formação prescrita, que pode ter qualquer formato, sendo que o formato pode variar no tempo. Durante a formação, a frota pode ter que manobrar, por exemplo, fazendo uma curva ou aumentar sua velocidade. Um único avião é definido como líder ou, alternativamente, uma entidade virtual chamada líder virtual é computada em cada VANT. Todos os aviões que não são o líder são definidos como seguidores. O líder (virtual ou não) é não cooperativo, no sentido que ele voa livremente sem tentar atingir a formação, nem é influenciado pelos seguidores. O formato da formação é descrito a partir da posição e orientação do líder. Todas as aeronaves possuem acesso às informações do líder. Os seguidores trocam dados apenas com seguidores próximos (vizinhos) ou não trocam dados entre si. O grafo de comunicação bidirecional entre seguidores, quando estes conversam entre si, é fixo e conhecido previamente. O controlador desenvolvido é robusto a incertezas e perturbações que afetem o sistema de modo limitado. Faz-se apenas um desenvolvimento teórico, mas almeja-se o teste em hardware real como trabalho futuro.

B.5 OBJETIVOS

O objetivo principal é desenvolver teoricamente um controlador capaz de efetuar um voo em formação autônomo com formato variável para aeronaves de asa fixa. O objetivo foi cumprido ao realizar os objetivos abaixo:

- Implementar um simulador de aeronave asa fixa realista, que recebe comando de atuadores e provê posição, velocidade e orientação como saída.
- Implementar em simulação um sistema de controle de malha interna.
- Revisar a literatura e avaliar alguns controladores de voo em formação autônomos já existentes.
- Desenvolver um controlador de voo em formação autônomo que obtém uma resposta transiente rápida e erro em regime permanente muito baixo. O controlador deve ser robusto a incertezas de modelos e perturbações.

- Demonstrar matematicamente a estabilidade do controlador desenvolvido.
- Avaliar os algoritmos desenvolvidos via simulações realistas.

B.6 CONTRIBUIÇÕES

- No Capítulo 3 é proposto um controlador por inversão não linear para a abordagem do sistema de referência do líder. Ele apresenta desempenho superior a um equivalente da literatura [11] quando ocorrem manobras de maior intensidade. O controlador desenvolvido foi apresentado em congresso [3].
- No capítulo 4 demonstra-se que um controlador robusto, síncrono, baseado em modos deslizantes com filtro passa baixas, inversão não linear, e abordagem do sistema de referência inercial, publicado recentemente em periódico científico [14] está conceitualmente errado.
- Apresenta-se uma correção para o controlador robusto da literatura. Dessa forma, remove-se o erro conceitual, e obtém-se um controlador mais simples. O controlador é demonstrado estável via função candidata de Lyapunov. Simulações demonstram que o controlador apresenta o desempenho esperado.
- Apresenta-se uma segunda arquitetura para o controlador proposto. Ela redireciona uma componente do sinal de controle para atuar diretamente na planta, ao invés de ser previamente filtrada. O resultado obtido é um controlador mais simples e que responde mais rápido, pois a componente que não é filtrada não tem sua banda limitada pelo filtro. Destaca-se que a componente não filtrada é naturalmente contínua e, por isso, o controlador continua sem apresentar *chattering*, mesmo aplicando parte do sinal de controle diretamente na planta. O novo controlador é demonstrado estável via função candidata de Lyapunov. Simulações demonstram que o controlador apresenta o desempenho esperado.