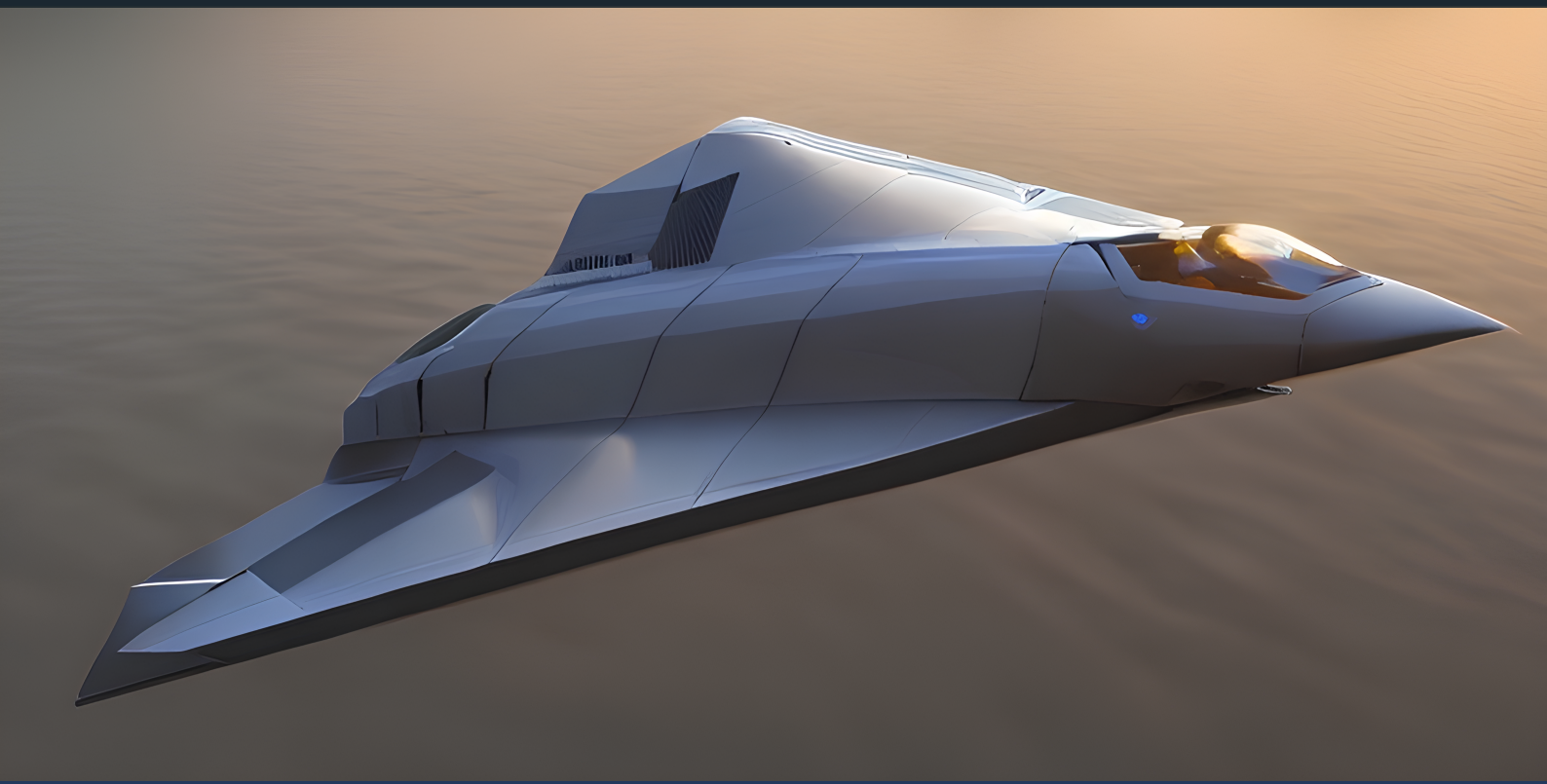
Adaptive dynamic incremental nonlinear control allocation

An actuator fault-tolerant control solution for high-performance aircraft

Noah Stam



TU Delft

Adaptive dynamic incremental nonlinear control allocation

An actuator fault-tolerant control solution for high-performance aircraft

Thesis report

by

Noah Stam

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on May 16, 2024 at 14:00

Thesis committee:

Chair:	Dr. ir. E.J.J. Smeur
Supervisor:	Dr. ir. C.C. de Visser
Examiner:	Dr. ir. E. Mooij
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	May, 2023 - May, 2024
Student number:	5660513

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Faculty of Aerospace Engineering · Delft University of Technology

Copyright © Noah Stam, 2024
All rights reserved.

Acknowledgements

This report represents the work I did for the past year. It also marks the final step towards my graduation from TU Delft and the end of my formal education. As for any student, the process of this thesis can be described by its ups and downs. Handing in this document I consider an up. It is a milestone I can be rightfully proud of. Nevertheless, it could not have occurred without the enthusiastic guidance from my supervisor Coen. Our bi-weekly meetings helped me to prioritize and to see what now lies before you, the tiny dot on the horizon. In addition, I want to thank Nikki for being by my side throughout my studies and work all the same. Combining the two has not always been straightforward. I could not have done this without her support.

Noah Stam,
April 2024,
Rotterdam

Contents

Nomenclature

List of Abbreviations

AD-INCA	Adaptive dynamic INCA
CA	Control allocation
CEJ	Control effectiveness Jacobian
D-INCA	Dynamic INCA
DQP	Dynamic quadratic programming
FFAC	Feed-forward actuator compensation
INCA	Incremental nonlinear control allocation
LCA	Lyapunov control allocation
MPCA	Model predictive control allocation
PCH	Pseudo-control hedge
QP	Quadratic programming
RLS	Recursive least squares

List of Symbols

$\mathbf{A}$	Actuator dynamics model state matrix	-
$\mathbf{B}$	Actuator dynamics model input matrix	-
$\mathbf{u}$	Command input vector	rad
$\mathbf{x}$	Aircraft state vector	-
$\mathbf{z}$	Second order actuator model state vector	-
J	Aircraft inertia matrix	$\text{lbf} \cdot \text{ft}^2$
p	Roll rate	$\text{rad} \cdot \text{s}^{-1}$
q	Pitch rate	$\text{rad} \cdot \text{s}^{-1}$
r	Yaw rate	$\text{rad} \cdot \text{s}^{-1}$
t	Continuous time	s
u	Longitudinal velocity in the body frame	$\text{ft} \cdot \text{s}^{-1}$
v	Lateral velocity in the body frame	$\text{ft} \cdot \text{s}^{-1}$
w	Normal velocity in the body frame	$\text{ft} \cdot \text{s}^{-1}$
α	Angle of attack	rad

β	Sideslip angle	rad
$\boldsymbol{\delta}$	Physical control input vector	rad
$\boldsymbol{\nu}$	Virtual control input vector	$\text{rad} \cdot \text{s}^{-2}$
$\boldsymbol{\omega}$	Body angular velocity vector	$\text{rad} \cdot \text{s}^{-1}$
$\boldsymbol{\tau}$	Torque vector	$\text{lbf} \cdot \text{ft}$
Δ	Increment	-
$\nabla_\delta \Phi$	Control effectiveness function Jacobian	$\text{lbf} \cdot \text{ft} \cdot \text{rad}^{-1}$
Φ	Control effectiveness function	$\text{lbf} \cdot \text{ft}$

Subscripts

c	Command
h	Hedge
k	Discrete time step
ref	Reference
rm	Reference model

List of Figures

List of Tables

1 Introduction

Stealth is one of the key design drivers for modern military aircraft. Minimizing signatures makes it harder for adversaries to detect, identify and engage an airborne asset. Therefore, a logical design step is to eliminate unnecessary corners, making a tailless configuration an interesting candidate. However, to maintain lateral control, high-manoeuvrability and add redundancy in control, tailless aircraft require unconventional, redundant control effectors. With this in mind, the innovative control effectors (ICE) program was conceived in 1993 to investigate the potential for new and innovative methods for stabilization and control of high performance, low all-aspect signature fighters [1]. Specifically, two tailless aircraft with novel control surfaces were designed; a carrier-based and a land-based configuration.

Over-actuation of dynamic systems, as seen in the ICE aircraft designs, potentially offers increased performance, redundancy, and agility in the control solution. However, to exploit these advantages requires high complexity control systems, introducing various engineering problems. Firstly, one must navigate through a continuous solution space to select a control solution that adheres to the physical actuator constraints. Secondly, complex systems may exhibit highly nonlinear cross-coupled dynamics. Matamoros [2] addressed these problems for the ICE land-based configuration through incremental nonlinear control allocation (INCA). INCA has proven to increase performance in real-time applications compared to linear control allocation (CA) techniques by exploiting nonlinear cross-coupling in the control effectiveness function.

Another challenge includes the actuator dynamics, often overlooked in CA techniques. Nevertheless, neglecting actuator dynamics may result in the optimal control solution not being achieved. Assuming perfect actuator dynamics in CA design can be argued for systems with relatively fast actuators compared to the system dynamics. However, for faster, high performance systems this assumption reduces performance of the CA strategy [3]. Ways to explicitly account for the actuator dynamics in CA have been proposed with model predictive control allocation [3, 4] and Lyapunov based control allocation [5, 6, 7]. Unfortunately, these methods suffer from high computational load and bring forth non-optimal control solutions, respectively. Moreover, more recent research [8] proposed reinforcement learning control allocation (RLCA) as a solution to the CA problem. However, the study concluded that performance remains less than that of model-based CA methods.

On the other hand, incremental methods [9, 10, 11] heavily rely on output derivative measurements rather than system dynamics modeling. INCA has proved effective in dealing with nonlinear non-control affine systems in a robust manner [2, 9]. Nevertheless, it was found that INCA is rather conservative in accounting for actuator saturation limits [3]. Also, system delays introduced by the actuator dynamics are not regarded in the control solution, hereby underutilizing a system's potential performance. Control schemes have been designed that account for actuator dynamics in incremental control allocation in additional control loops [3, 10], however, no previous study has combined the dynamic incremental control allocation problem in a single method. Consequently, the integration of so-called dynamic CA techniques with incremental nonlinear control remains under-addressed in existing research.

Besides performance gain, over-actuation may offer increased actuator fault tolerance through redundant effectors. To harness this advantage, previous research has elaborated on adaptive control for several CA methods. For example, parameter update laws have been defined for both the actuator dynamics

model [7] as well as the control effectiveness model [4, 7, 12]. While INCA provides robustness in control effectiveness model mismatch, adaptive control may leverage actuator redundancy more effectively.

The current thesis' contribution includes the design of a CA scheme for non-control affine systems that accounts for actuator dynamics in the optimal CA solution, and adapts its strategy for in-flight actuator fault modes. INCA presents significant advantages in terms of robust CA for non-control affine systems. In addition, the incremental approach allows for linear solvers and presents flexibility in defining the CA problem. Therefore, INCA will be used as the baseline method in this thesis. The research objective can be defined as follows.

Research Objective

The primary objective of this research project is to design an incremental nonlinear control allocation method that compensates for actuator dynamics and adapts for online actuator failure modes for fast non-control affine, nonlinear over-actuated systems.

Subsequently, the main research question that can be posed is:

Research Question

How can actuator dynamics and online actuator failure modes in non-control affine nonlinear over-actuated systems be effectively accounted for within the incremental nonlinear control allocation framework?

Several sub-questions will need to be answered in light of the main research question. These include:

1. How can actuator dynamics be taken into account within the INCA control loop under output derivative measurement constraints and possible onboard model limitations? And
2. to what extent can the compensation of actuator dynamics in the incremental nonlinear control allocation framework improve the tracking performance?
3. What update laws are required to account for actuator failures under the incremental nonlinear control allocation framework? And
4. to what extent can these update laws increase tracking performance under actuator failure modes?

In order to answer these questions, this thesis will start with a scientific article in Part I. This article includes both the adopted methodology and the derived results and conclusions. Subsequently, the preliminary literature review is included in Part II. Any results additional to those in the scientific article are depicted and discussed in Part III. Lastly, the research questions are answered in Part IV. This final chapter also includes the limitations of the current study and recommendations for future work.

Part I

Scientific Article

Adaptive dynamic incremental nonlinear control allocation and its applications for innovative control effectors

N. Stam∗
Delft University of Technology, 2629HS Delft, The Netherlands

Neglecting actuator dynamics in nonlinear control and control allocation can lead to performance degradation, especially when considering fast dynamic systems. This paper provides a novel method to account for actuator dynamics in the control allocation solution, dynamic incremental nonlinear control allocation, or D-INCA. The incremental approach allows for the implementation of a first order discrete-time actuator dynamics model in the quadratic programming (QP) solver. This model is used to find the optimal command inputs in addition to the desired physical actuator deflections, hereby compensating for actuator dynamics delays. Whereas, the baseline incremental nonlinear control allocation (INCA) approach requires pseudo-control hedging of the outer loop reference to increase closed loop stability margins under actuator dynamics delays. To its advantage, D-INCA does not require feedback of higher order output derivatives than INCA and can be used with nonlinear non-control affine systems. Furthermore, with adaptive D-INCA, or AD-INCA, an actuator dynamics parameter estimator is introduced to adapt the actuator model online, minimizing actuator tracking errors after actuator failures. The proposed methods are applied to a fighter aircraft model with an over-actuated innovative control effectors suite and results are compared to the baseline INCA controller.

Nomenclature

$\mathbf{A}$	=	actuator system state matrix
$\mathbf{B}$	=	actuator system input matrix
b	=	wing span [ft]
C	=	aerodynamic coefficient
$\bar{c}$	=	mean aerodynamic chord [ft]
$\mathbb{D}$	=	set of feasible physical control inputs [rad]
$\mathbf{e}$	=	output error

∗MSc student, Control and Simulation Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629HS Delft, The Netherlands

$\mathbf{f}(\mathbf{x})$	=	system state dynamics equation
$\mathbf{F}(\mathbf{x}, \boldsymbol{\tau})$	=	system output derivative function
$\mathbf{G}(\mathbf{x})$	=	system input dynamics equation
$H(s)$	=	transfer function
$\mathbf{h}(\mathbf{x})$	=	system output dynamics equation
I	=	identity matrix
J	=	aircraft inertia matrix [lbf $\cdot$ ft^2]
$\mathcal{J}$	=	cost function
K	=	recursive least squares gain
k	=	discrete time step
$L_f h(\mathbf{x})$	=	Lie derivative of scalar function $h(\mathbf{x})$ along the vector $\mathbf{f}(\mathbf{x})$
L	=	roll moment [lbf $\cdot$ ft]
M	=	pitch moment [lbf $\cdot$ ft]
N	=	yaw moment [lbf $\cdot$ ft]
P	=	recursive least squares parameter estimate covariance
p	=	roll rate [rad $\cdot$ s^{-1}]
q	=	pitch rate [rad $\cdot$ s^{-1}]
$\mathbb{R}$	=	set of real numbers
r	=	yaw rate [rad $\cdot$ s^{-1}]
t	=	continuous time [s]
$\mathbf{U}$	=	matrix with upper and lower bound command inputs [rad]
$\mathbf{u}$	=	command input vector [rad]
V	=	true airspeed [ft $\cdot$ s^{-1}]
$\mathcal{W}$	=	active set-based programming working set
$\mathbf{x}$	=	system state vector
$\mathbf{y}$	=	system output vector
α	=	angle of attack [rad]
β	=	sideslip angle [rad]
Δ	=	increment
$\boldsymbol{\delta}$	=	physical control deflection [rad]
$\boldsymbol{\kappa}$	=	perturbation for the active set-based optimizer [rad]
$\boldsymbol{\epsilon}$	=	recursive least squares predication error [rad]

η = maximum distance to feasibility in the active set-based optimizer

$\hat{\boldsymbol{\theta}}$ = actuator dynamics parameter estimates

Λ = recursive least squares adaptive forgetting factor

λ = Lagrange multiplier for the active set-based optimizer

$\boldsymbol{\nu}$ = virtual-control input vector [rad $\cdot$ s^{-2}]

$\boldsymbol{\xi}$ = recursive least squares regressor [rad]

Σ_0 = recursive least squares information filter

$\boldsymbol{\tau}$ = pseudo-control input vector [lbf $\cdot$ ft]

$\boldsymbol{\Phi}$ = control effectiveness function [lbf $\cdot$ ft]

ϕ = roll angle [rad]

$\nabla_\delta \boldsymbol{\Phi}$ = control effectiveness Jacobian [lbf $\cdot$ ft $\cdot$ rad^{-1}]

ω = body angular rates [rad $\cdot$ s^{-1}]

I. Introduction

Over-actuation of dynamic systems potentially offers increased performance, redundancy, and agility in the control solution. However, to exploit these advantages requires high complexity control systems, introducing various engineering problems. Firstly, one must navigate through a continuous solution space to select a control solution that adheres to the physical actuator constraints. Secondly, complex systems may exhibit highly nonlinear cross-coupled dynamics. These problems have been studied thoroughly, however, still remain subject to ongoing research. Another challenge in control allocation (CA) includes actuator dynamics compensation in the CA solution, also referred to as dynamic CA.

The choice to neglect actuator dynamics in CA techniques can be argued for systems with relatively fast actuators compared to the system dynamics. However, for faster, high performance systems it is found that neglecting the actuator dynamics reduces performance of the CA strategy [1]. Ways to explicitly account for the actuator dynamics have been proposed with model predictive control allocation [1, 2] and Lyapunov based control allocation [3–5]. Unfortunately, these methods suffer from high computational load and bring forth non-optimal control solutions, respectively. More recent research [6] showed reinforcement learning control allocation (RLCA) can learn to effectively distribute the control moments and forces over the available actuators. However, the study concluded that performance remains less than that of model-based CA methods.

On the other hand, incremental methods [7–9] heavily rely on output derivative measurements rather than system dynamics modeling. Moreover, incremental nonlinear control allocation (INCA) has proven effective in dealing with nonlinear non-control affine systems in a robust manner [7, 10]. Control schemes have been designed that account for actuator dynamics in incremental CA in additional control loops [1, 8], however, no previous research has combined

the dynamic incremental CA problem in a single control loop. Consequently, the integration of dynamic CA techniques with incremental nonlinear control remains under-addressed in previous research.

Although the INCA framework provides significant robustness, over-actuated systems may offer far more potential in terms of actuator fault-tolerant control. In order to harness this advantage, previous research has elaborated on adaptive CA for several other CA methods. For example, parameter update laws have been defined for both the actuator dynamics model [5] as well as the control effectiveness model [2, 5, 11].

The current research paper's contribution is the design of an adaptive dynamic incremental nonlinear control allocation scheme, or AD-INCA, for non-control affine systems that compensates for actuator dynamics in the optimal CA solution, and adapts its strategy for in-flight actuator fault modes. Furthermore, INCA presents significant advantages in terms of robust CA for non-control affine systems. In addition, the incremental approach allows for linear solvers and flexibility in defining the CA problem. Therefore, INCA is considered the baseline controller for this study.

This paper is organized as follows. Firstly, a brief review on feedback linearization will be given in Section II. In Section III the step will be made from incremental feedback linearization towards INCA. The theory behind the dynamic CA problem and the proposed control solution are discussed in Section III.A, along with a novel adaptive dynamic CA scheme in Section III.B. The innovative control effectors suite and its flight control system are introduced in Section IV and Section V. Furthermore, simulations and results are portrayed in Section VI. Lastly, conclusions on the proposed CA methods are derived in Section VII, including limitations and recommendations for future research.

II. Nonlinear control

The dynamics of a nonlinear time-invariant input-affine system are governed by both state and input equations. Assume a multiple-input multiple-output (MIMO) system described by the differential equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\boldsymbol{\tau} \tag{1}$$

where $\mathbf{f}$ is a smooth vector field on $\mathbb{R}^n$. $\mathbf{G} \in \mathbb{R}^{n\times m}$ is a matrix whose columns are smooth vector fields. Moreover, $\mathbf{x} \in \mathbb{R}^n$ is the state vector and $\boldsymbol{\tau} \in \mathbb{R}^m$ is the vector of pseudo-control inputs representing a number forces and moments to be controlled m. The output equation, assuming no feedthrough, becomes

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{2}$$

with $\mathbf{h}$ a smooth vector field on $\mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ the output vector. The relation between the physical control deflection $\boldsymbol{\delta}$ and the pseudo-control input $\boldsymbol{\tau}$ is described by a state dependent control effectiveness function

$$\boldsymbol{\tau} = \boldsymbol{\Phi}(\mathbf{x}, \boldsymbol{\delta}) \tag{3}$$

where $\mathbf{\Phi}$ is a smooth vector field on $\mathbb{R}^{n+m}$ and the physical control inputs $\boldsymbol{\delta} \in \mathbb{D} \subset \mathbb{R}^p$. With p being the number of independent control actuators and $\mathbb{D}$ the set of feasible physical control inputs. Substituting Eq. (3) into Eq. (1) results in a non-input affine nonlinear time-invariant system. In addition, the actuator dynamics can be distinguished. Assuming these to be linear as in

$$\dot{\boldsymbol{\delta}} = \mathbf{A}\boldsymbol{\delta} + \mathbf{B}\mathbf{u} \tag{4}$$

with $\mathbf{A}$ and $\mathbf{B}$ the state and input matrices, respectively. The goal of any control system is to select a command input $\mathbf{u}$ such that the system output vector follows a reference signal $\mathbf{y}_{\text{ref}}$. To achieve this goal, a control input $\boldsymbol{\delta}_{\text{ref}}$ and pseudo-control input $\boldsymbol{\tau}_{\text{ref}}$ need to be tracked. A graphical representation of Eqs. (1) to (4) is depicted in Fig. 1.

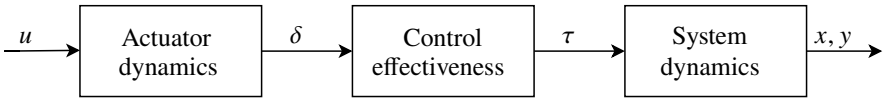


Fig. 1 Open loop dynamics schematic.

A. Nonlinear dynamic inversion

One well-established approach to nonlinear control is nonlinear dynamic inversion (NDI). This method employs a dual-layered control structure, featuring a linear outer loop controller coupled with an inner linearization loop. The fundamental concept behind this so-called feedback linearization is to iteratively differentiate the output equation r times in order to reveal the input explicitly, with r being the relative degree of the system. Next, the obtained nonlinear dynamics for r differentiations are inverted in order to linearize the input-output relationship, which can then be effectively represented by a serie of integrators. The pseudo-control input $\boldsymbol{\tau}$ becomes a function of a virtual control input $\boldsymbol{\nu}$ which can be controlled by a linear outer loop controller [12]. Consider a MIMO system with $\mathbf{h}(\mathbf{x})$ on $\mathbb{R}^n$, output vector $\mathbf{y} \in \mathbb{R}^m$, input vector $\boldsymbol{\tau} \in \mathbb{R}^m$. Let the first time derivative of the i-th component y_i of the output vector $\mathbf{y}$ be

$$\dot{y}_i = \frac{\mathrm{d}y_i}{\mathrm{d}t} = \frac{\partial h_i}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \nabla h_i(\mathbf{x})\dot{\mathbf{x}} = \nabla h_i(\mathbf{x})\left[\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\boldsymbol{\tau}\right] \tag{5}$$

with $\nabla h_i(\mathbf{x})$ the gradient of the scalar function $h_i \in \mathbf{h}(\mathbf{x})$ with respect to the system states $\mathbf{x}$. Next, it is convenient to introduce the Lie derivative as

$$L_f h_i(\mathbf{x}) = \nabla h_i(\mathbf{x})\, \mathbf{f}(\mathbf{x}) = \sum_{j=1}^{n} \frac{\partial h_i(\mathbf{x})}{\partial x_j} f_j(\mathbf{x}). \tag{6}$$

Furthermore, similar to any ordinary derivative, a Lie derivative can be computed multiple times. For example, the k-th derivative with respect to $\mathbf{x}$ along the vector $\mathbf{f}(\mathbf{x})$ is

$$L_f^k h_i(\mathbf{x}) = L_f\left(L_f^{k-1} h_i(\mathbf{x})\right) = \nabla\left(L_f^{k-1} h_i(\mathbf{x})\right)\mathbf{f}(\mathbf{x}) \tag{7}$$

with $L_f^0 h_i(\mathbf{x}) = h_i(\mathbf{x})$. Adopting this definition, Eq. (5) becomes

$$\dot{y}_i = \nabla h_i \Big[\mathbf{f}(\mathbf{x}) + \mathbf{g}_1(\mathbf{x})\tau_1 + \cdots + \mathbf{g}_m(\mathbf{x})\tau_m \Big] = L_f h_i(\mathbf{x}) + \sum_{j=1}^{m} L_{g_j} h_i(\mathbf{x})\tau_j \tag{8}$$

with $\mathbf{g}_j$ the smooth column vector functions in $\mathbf{G}$ and τ_j the pseudo-control components in $\boldsymbol{\tau}$. Moreover, by differentiating the output equation k times, we obtain the more general solution

$$y_i^{(k)} = L_f^k h_i(\mathbf{x}) + \sum_{j=1}^{m} L_{g_j} L_f^{k-1} h_i(\mathbf{x})\tau_j. \tag{9}$$

The pseudo-control input does not show in the expressions for $y_i^{(k)}$ until $k = r_i$. Hence, for $k < r_i$ the Lie derivatives $L_{g_j} L_f^{k-1} h_i(\mathbf{x}) = 0 \quad \forall j$. The solutions for $k = r_i$ become

$$y_i^{(r_i)} = L_f^{r_i} h_i(\mathbf{x}) + \sum_{j=1}^{m} L_{g_j} L_f^{r_i - 1} h_i(\mathbf{x})\tau_j \tag{10}$$

with $\sum_{j=1}^{m} L_{g_j} L_f^{r_i-1} h_i(\mathbf{x})\tau_j \neq 0$ for at least one j. The remaining unobservable input-output relationships $r_i < k \leq n$ are referred to as the internal dynamics of the system. These dynamics need to be asymptotically stable for the NDI controller to function properly. The resulting equations in Eq. (10) for all the outputs y_i can be put into matrix form

$$\underbrace{\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix}}_{\mathbf{y}^{(r)}} = \underbrace{\begin{bmatrix} L_f^{r_1} h_1(\mathbf{x}) \\ L_f^{r_2} h_2(\mathbf{x}) \\ \vdots \\ L_f^{r_m} h_m(\mathbf{x}) \end{bmatrix}}_{\mathbf{c}(\mathbf{x})} + \underbrace{\begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(\mathbf{x}) & L_{g_2} L_f^{r_1-1} h_1(\mathbf{x}) & \cdots & L_{g_m} L_f^{r_1-1} h_1(\mathbf{x}) \\ L_{g_1} L_f^{r_2-1} h_2(\mathbf{x}) & L_{g_2} L_f^{r_2-1} h_2(\mathbf{x}) & \cdots & L_{g_m} L_f^{r_2-1} h_2(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(\mathbf{x}) & L_{g_2} L_f^{r_m-1} h_m(\mathbf{x}) & \cdots & L_{g_m} L_f^{r_m-1} h_m(\mathbf{x}) \end{bmatrix}}_{\mathbf{D}(\mathbf{x})} \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_m \end{bmatrix}}_{\tau} \tag{11}$$

or for simplicity reasons in short

$$\mathbf{y}^{(r)} = \mathbf{F}(\mathbf{x}, \boldsymbol{\tau}) \tag{12}$$

with $\mathbf{F} \in \mathbb{R}^{n \times m; m}$ a smooth vector function. Assuming $\mathbf{D}(\mathbf{x})$ is full rank, a virtual control input $\boldsymbol{\nu}$ can be defined. By choosing the virtual control law as

$$\boldsymbol{\tau} = \mathbf{D}^{-1}(\mathbf{x})(\boldsymbol{\nu} - \mathbf{c}(\mathbf{x})), \tag{13}$$

a linear relationship between $\boldsymbol{\nu}$ and the output

$$\boldsymbol{\nu} = \mathbf{y}^{(r)} \tag{14}$$

is obtained. Hence, the inner loop dynamics are reduced to a cascade of integrators and all nonlinearities are canceled. The outer loop controller can now be an ordinary PID controller that feeds the inner loop with the virtual control input. A candidate for the linear control law is

$$\boldsymbol{\nu} = \mathbf{y}_{ref}^{(r)} + K(\mathbf{e}, \dot{\mathbf{e}}, \ldots, \mathbf{e}^{(r-1)}) \tag{15}$$

with $\mathbf{e} = \mathbf{y} - \mathbf{y}_{\text{ref}}$ and $K(\mathbf{e}, \dot{\mathbf{e}}, \ldots, \mathbf{e}^{(r-1)})$ a feedback control law that can be tuned for the desired error dynamics.

B. Incremental nonlinear dynamic inversion

A downside of NDI control is that complete knowledge about the controlled system and its states is required. To reduce this dependency, an incremental approach or incremental nonlinear dynamic inversion (INDI) can be adopted. This method computes the control signal in individual increments $\Delta\boldsymbol{\tau}$. In discrete time, $\mathbf{y}^{(r)}$ can be linearized for each time step. Taking the Taylor series of Eq. (12) around a local point $(\mathbf{x}_k, \boldsymbol{\tau}_k)$ yields

$$\mathbf{y}^{(r)} = \mathbf{F}(\mathbf{x}_k, \boldsymbol{\tau}_k) + \frac{\partial \mathbf{F}(\mathbf{x}_k, \boldsymbol{\tau}_k)}{\partial \mathbf{x}} (\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{\partial \mathbf{F}(\mathbf{x}_k, \boldsymbol{\tau}_k)}{\partial \boldsymbol{\tau}} (\boldsymbol{\tau}_{k+1} - \boldsymbol{\tau}_k) + O \tag{16}$$

or

$$\mathbf{y}^{(r)} = \mathbf{F}(\mathbf{x}_k, \boldsymbol{\tau}_k) + \mathbf{F}_{\mathbf{x}}(\mathbf{x}_k, \boldsymbol{\tau}_k) (\mathbf{x}_{k+1} - \mathbf{x}_k) + \mathbf{F}_{\boldsymbol{\tau}}(\mathbf{x}_k, \boldsymbol{\tau}_k) (\boldsymbol{\tau}_{k+1} - \boldsymbol{\tau}_k) + O \tag{17}$$

with O being the higher order terms. These terms will be discarded for the control design. Next, it is assumed that the change in pseudo-control input is orders of magnitude faster than the change in the state vector. Under this assumption, for a sufficiently small sampling time, Eq. (17) is reduced to

$$\mathbf{y}^{(r)} \approx \mathbf{F}(\mathbf{x}_k, \boldsymbol{\tau}_k) + \mathbf{F}_{\boldsymbol{\tau}}(\mathbf{x}_k, \boldsymbol{\tau}_k)\Delta\boldsymbol{\tau} \tag{18}$$

with $\Delta\boldsymbol{\tau} = \boldsymbol{\tau}_{k+1} - \boldsymbol{\tau}_k$. Adopting the definition in Eq. (14), the following virtual control to pseudo-control increment mapping is obtained

$$\Delta\boldsymbol{\tau} = \mathbf{F}_{\boldsymbol{\tau}}^{-1}(\mathbf{x}_k, \boldsymbol{\tau}_k)(\boldsymbol{\nu} - \mathbf{y}_k^{(r)}). \tag{19}$$

Next, Eq. (19) can be rewritten in terms of the former obtained $\mathbf{D}(\mathbf{x})$ matrix of Lie derivatives in Eq. (11)

$$\Delta\boldsymbol{\tau} = \mathbf{D}^{-1}(\mathbf{x}_k)(\boldsymbol{\nu} - \mathbf{y}_k^{(r)}). \tag{20}$$

Hence, the dependency on plant dynamics knowledge is reduced. Instead, this incremental method relies on output derivative measurements $\mathbf{y}_k^{(r)}$ to account for the system dynamics. Knowledge of the input dynamics is still required.

Equation (20) only considers increments in pseudo-control input. The physical deflection $\boldsymbol{\delta}$ to achieve the pseudo-control increments is not yet included. However, this control effectiveness can easily be accounted for. Linearizing the control effectiveness function Eq. (3) around the local operating point yields [10]

$$\Delta\boldsymbol{\tau} = \frac{\partial\boldsymbol{\Phi}(\mathbf{x}_k, \boldsymbol{\delta}_k)}{\partial\mathbf{x}}(\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{\partial\boldsymbol{\Phi}(\mathbf{x}_k, \boldsymbol{\delta}_k)}{\partial\boldsymbol{\delta}}(\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k) + \mathcal{O}. \tag{21}$$

Applying the same time-scale separation principle, the state dependent term can be discarded, yielding

$$\Delta\boldsymbol{\tau} \approx \frac{\partial\boldsymbol{\Phi}(\mathbf{x}_k, \boldsymbol{\delta}_k)}{\partial\boldsymbol{\delta}}\Delta\boldsymbol{\delta} = \nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi}\Delta\boldsymbol{\delta}. \tag{22}$$

$\nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi}$ indicates the gradient of $\boldsymbol{\Phi}$ in the directions $\delta_i \in \boldsymbol{\delta}$. This function is referred to as the control effectiveness Jacobian (CEJ) throughout this work. Moreover, assuming perfect zero-order hold actuator dynamics $\Delta\boldsymbol{\delta} = \boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k$. However, in reality the control increment may be governed by more complex dynamics. Rewriting Eq. (22) leads to

$$\Delta\boldsymbol{\delta} = (\nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi})^{-1}\Delta\boldsymbol{\tau}. \tag{23}$$

Substituting Eq. (20) in Eq. (23) results in the following expression for the control increment

$$\Delta\boldsymbol{\delta} = (\nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi})^{-1}\mathbf{D}^{-1}(\mathbf{x}_k)(\boldsymbol{\nu} - \mathbf{y}_k^{(r)}). \tag{24}$$

Which relies on the same assumption as for the NDI case that $\mathbf{D}(\mathbf{x}_k)$ is invertible. For the case $p = m$ the Jacobian matrix $\nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi}$ can also be inverted. However, in case of an over-actuated system, the Jacobian matrix cannot be inverted and Eq. (24) cannot be solved.

III. Incremental nonlinear control allocation

To solve the control increment for over-actuated systems, Tol [7] proposed a control method that combines INDI with a spline based CA technique. This work adopted a multivariate B-spline model to capture the control effectiveness function. A more general form of this method was described and implemented by Matamoros [10]. The so-called incremental nonlinear control allocation (INCA) technique aims to find a control increment $\Delta\boldsymbol{\delta}$ such that the pseudo-control reference is achieved. In mathematical terms: solve Eq. (22) subject to the constraint

$$\underline{\Delta\boldsymbol{\delta}} \leq \Delta\boldsymbol{\delta} \leq \overline{\Delta\boldsymbol{\delta}} \tag{25}$$

with $\underline{\Delta\boldsymbol{\delta}}$ and $\overline{\Delta\boldsymbol{\delta}}$ the lower and upper limits of the control increments, based on the most stringent limitation between the physical actuator position and rate constraints

$$\begin{aligned} \underline{\Delta\boldsymbol{\delta}} &= \max\left(\boldsymbol{\delta}_{\min} - \boldsymbol{\delta}_k, -\dot{\boldsymbol{\delta}}_{\max}\Delta t\right) \\ \overline{\Delta\boldsymbol{\delta}} &= \min\left(\boldsymbol{\delta}_{\max} - \boldsymbol{\delta}_k, \dot{\boldsymbol{\delta}}_{\max}\Delta t\right) \end{aligned} \tag{26}$$

with $\boldsymbol{\delta}_{\min}$ and $\boldsymbol{\delta}_{\max}$ the vectors with the minimum and maximum actuator positions and $\dot{\boldsymbol{\delta}}_{\max}$ the vector containing the maximum rates of control deflection. The sampling time is denoted as Δt.

Moreover, since the current problem definition is linear in the control increments, it can be solved with linear optimizers. Previous research [10] adopted the quadratic programming (QP) active-set based solver for INCA and it was found that this method outperformed other linear methods. In QP an ℓ_2-norm cost function is defined that should be minimized to find an optimal solution. Furthermore, since the current problem is under-determined, an additional equation is required besides minimizing the error in pseudo-control, or CA error. One example of a secondary objective is to drive the output to a preference value, usually zero. This approach prevents the use of greater control inputs than necessary. For the incremental case a QP cost function and constraints can be defined as

$$\begin{aligned} \min_{\Delta\boldsymbol{\delta}} \mathcal{J} = &\left\| \mathbf{W}_1 \left(\nabla_{\boldsymbol{\delta}} \boldsymbol{\Phi} \Delta\boldsymbol{\delta} - \Delta\boldsymbol{\tau}_c \right) \right\|_2 + \left\| \mathbf{W}_2 \left(\Delta\boldsymbol{\delta} - \Delta\boldsymbol{\delta}_p \right) \right\|_2 \\ &\text{subject to } \underline{\Delta\boldsymbol{\delta}} \leq \Delta\boldsymbol{\delta} \leq \overline{\Delta\boldsymbol{\delta}}, \end{aligned} \tag{27}$$

with $\Delta\boldsymbol{\tau}_c$ the commanded increment in pseudo-control input. Moreover, $\Delta\boldsymbol{\delta}_p$ indicates the preferred increment in control deflection. In Eq. (27) the first term minimizes the CA error in a weighted least-squares sense in accordance with Eq. (22). The second objective attracts the control element to a preferred value. $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices that can give preference to either one of these two objectives or prioritize between the pseudo-control components $\tau_j \in \boldsymbol{\tau}$ or actuators $\delta_i \in \boldsymbol{\delta}$. Before applying the QP optimizer to an incremental controller, however, the preferred control deflection needs to be evaluated for every time step. Otherwise, if the preferred control increment would be set to zero, the cost function would not drive the control actuators to zero, but rather try to keep them in their current position. To solve this problem, Matamoros [10] adopted the following definition

$$\Delta\boldsymbol{\delta}_p = \min\left(\left| \boldsymbol{\delta}_p - \boldsymbol{\delta}_k \right|, |\overline{\Delta\boldsymbol{\delta}}|, |\underline{\Delta\boldsymbol{\delta}}| \right) \cdot \operatorname{sign}\left(\boldsymbol{\delta}_p - \boldsymbol{\delta}_k \right) \tag{28}$$

with $\boldsymbol{\delta}_p$ the (zero) vector of preferred actuator deflections and $\boldsymbol{\delta}_k$ the current actuator deflections at time step k. The advantage of QP when applied in increments, is that the CEJ model and the weights do not need to be constant. Instead they can be adapted at each increment which can be beneficial during actuator failure modes [13]. Moreover, the QP problem can be solved through various optimization methods, such as the active set-based method described by

Härkegård [14].

A drawback of INCA is that it only computes the optimal control inputs and then adopts these values as command inputs, implicitly assuming perfect actuator dynamics. De Heer [1] found that with assuming perfect actuator dynamics, the optimal control solutions are not actually achieved. Also, INCA is rather conservative in defining the QP output constraints as the physical actuator constraints are directly imposed on the command inputs and not on the actuator deflections.

A. Dynamic incremental nonlinear control allocation

To overcome these drawbacks, this article introduces a novel dynamic incremental nonlinear control allocation (D-INCA) method to compute the required command input $\mathbf{u}$ in addition to the desired physical deflection $\boldsymbol{\delta}$. In other words, the actuator dynamics as depicted on the left-hand side of Fig. 1 are taken into account. Assume linear first order actuator dynamics in discrete state space form

$$\boldsymbol{\delta}_{k+1} = \mathbf{A}\boldsymbol{\delta}_k + \mathbf{B}\mathbf{u}_k \tag{29}$$

with diagonal discrete time state and input matrices $\mathbf{A}$ and $\mathbf{B}$. The increment in control deflection can be written as

$$\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k = \Delta\boldsymbol{\delta}_k = \mathbf{A}\boldsymbol{\delta}_k + \mathbf{B}\mathbf{u}_k - \boldsymbol{\delta}_k. \tag{30}$$

Substituting the expression for $\Delta\boldsymbol{\delta}_k$ in Eq. (22) results in the following equality

$$\Delta\boldsymbol{\tau} = \nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi}(\mathbf{A}\boldsymbol{\delta}_k + \mathbf{B}\mathbf{u}_k - \boldsymbol{\delta}_k), \tag{31}$$

which can be solved directly for $\mathbf{u}_k$ for non over-actuated systems ($p = m$). For over-actuated systems, a cost function analogous to Eq. (27) can be found by substituting Eq. (30) into Eq. (27) and rewriting the inequality constraint for $\mathbf{u}_k$

$$\begin{aligned} \min_{\mathbf{u}_k} \mathcal{J} = &\left\| \mathbf{W}_1 \left(\nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi}(\mathbf{A}\boldsymbol{\delta}_k + \mathbf{B}\mathbf{u}_k - \boldsymbol{\delta}_k) - \Delta\boldsymbol{\tau}_c \right) \right\|_2 + \left\| \mathbf{W}_2 \left((\mathbf{A}\boldsymbol{\delta}_k + \mathbf{B}\mathbf{u}_k - \boldsymbol{\delta}_k) - \Delta\boldsymbol{\delta}_p \right) \right\|_2, \\ &\text{subject to } \mathbf{B}^{-1}\left(\underline{\Delta\boldsymbol{\delta}} - \mathbf{A}\boldsymbol{\delta}_k + \boldsymbol{\delta}_k \right) \leq \mathbf{u}_k \leq \mathbf{B}^{-1}\left(\overline{\Delta\boldsymbol{\delta}} - \mathbf{A}\boldsymbol{\delta}_k + \boldsymbol{\delta}_k \right), \end{aligned} \tag{32}$$

with $\underline{\Delta\boldsymbol{\delta}}$ and $\overline{\Delta\boldsymbol{\delta}}$ being the most pressing lower and upper boundaries from the actuator rate and position constraints as described by Eq. (26). The optimal solution for $\mathbf{u}_k$ now depends on the first order actuator dynamics described by $(\mathbf{A}, \mathbf{B})$ as well as the CEJ $\nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi}$. Equation (27) was solved by Matamoros [10] through an active set-based optimization

method. This method can also be applied to the dynamic QP problem by reformulating Eq. (32) as

$$\min_{\mathbf{u}_k} \left\| \begin{bmatrix} \mathbf{W}_1 \nabla_\delta \mathbf{\Phi} \mathbf{B} \\ \mathbf{W}_2 \mathbf{B} \end{bmatrix} \mathbf{u}_k + \begin{bmatrix} \mathbf{W}_1 \nabla_\delta \mathbf{\Phi} (\mathbf{A}\boldsymbol{\delta}_k - \boldsymbol{\delta}_k) \\ \mathbf{W}_2 (\mathbf{A}\boldsymbol{\delta}_k - \boldsymbol{\delta}_k) \end{bmatrix} - \begin{bmatrix} \mathbf{W}_1 \Delta\boldsymbol{\tau}_c \\ \mathbf{W}_2 \Delta\boldsymbol{\delta}_p \end{bmatrix} \right\|, \tag{33}$$
$$\text{subject to } \mathbf{C}\mathbf{u}_k \geq \mathbf{U},$$

with $\mathbf{C} = [I \quad -I]^T$ and $\mathbf{U} = \left[\mathbf{B}^{-1}\left(\underline{\Delta\boldsymbol{\delta}} - \mathbf{A}\boldsymbol{\delta}_k + \boldsymbol{\delta}_k\right) \quad \mathbf{B}^{-1}\left(\overline{\Delta\boldsymbol{\delta}} - \mathbf{A}\boldsymbol{\delta}_k + \boldsymbol{\delta}_k\right) \right]$ indicating the lower and upper bound command inputs that respect the physical actuator constraints. The active set algorithm solves a sequence of equality constrained problems, where in every iteration some inequality constraints are regarded as equality constraints and form a working set $\mathcal{W}$, whereas the remaining inequality constraints are disregarded. The working set at the optimum is known as the active set of the solution. The active set algorithm taken from [14], used to solve the QP problem in Eq. (33), is given in pseudo-code in Algorithm 1.

Algorithm 1 Active set-based quadratic programming control allocation with first order actuator dynamics.

Let $\mathbf{u}^0$ be a feasible starting point. A point is feasible if it satisfies $\mathbf{C}\mathbf{u} \geq \mathbf{U}$. Let the working set $\mathcal{W}$ contain the active inequality constraints at $\mathbf{u}^0$. Let N be the maximum number of iterations.

for i=0,1,…,$N-1$ **do**
  Given $\mathbf{u}^i$, find the optimal perturbation $\boldsymbol{\kappa}$, considering the constraints in the working set as equality constraints and disregarding the remaining inequality constraints, by solving

$$\min_{\boldsymbol{\kappa}} \left\| \begin{bmatrix} \mathbf{W}_1 \nabla_\delta \mathbf{\Phi} \mathbf{B} \\ \mathbf{W}_2 \mathbf{B} \end{bmatrix} \left(\mathbf{u}^i + \boldsymbol{\kappa}\right) + \begin{bmatrix} \mathbf{W}_1 \nabla_\delta \mathbf{\Phi} (\mathbf{A}\boldsymbol{\delta}_k - \boldsymbol{\delta}_k) \\ \mathbf{W}_2 (\mathbf{A}\boldsymbol{\delta}_k - \boldsymbol{\delta}_k) \end{bmatrix} - \begin{bmatrix} \mathbf{W}_1 \Delta\boldsymbol{\tau}_c \\ \mathbf{W}_2 \Delta\boldsymbol{\delta}_p \end{bmatrix} \right\| \tag{34}$$
$$\text{subject to } \boldsymbol{\kappa}_i = 0, \quad i \in \mathcal{W}.$$

  if $\mathbf{u}^i + \boldsymbol{\kappa}$ is feasible **then**
    Set $\mathbf{u}^{i+1} = \mathbf{u}^i + \boldsymbol{\kappa}$ and compute the Lagrange multiplier λ associated with the active constraints in $\mathbf{C}\mathbf{u}^{i+1} \geq \mathbf{U}$.
    if All $\lambda \geq 0$ **then**
      $\mathbf{u}_k = \mathbf{u}^{i+1}$ is the optimal command input. $\boldsymbol{\delta}_{c,k+1} = \mathbf{A}\boldsymbol{\delta}_k + \mathbf{B}\mathbf{u}_k$ is the optimal control input. Terminate operation.
    else
      Remove the constraint associated with the most negative λ from $\mathcal{W}$.
    end if
  else
    Determine the maximum step length η such that $\mathbf{u}^{i+1} = \mathbf{u}^i + \eta\boldsymbol{\kappa}$ is feasible. Add the dominant constraint to the working set.
  end if
end for

The Lagrange multipliers are computed in accordance with [14]

$$\begin{bmatrix} \mathbf{W}_1\nabla_\delta\mathbf{\Phi}\mathbf{B} \\ \mathbf{W}_2\mathbf{B} \end{bmatrix}^T \left(\left(\begin{bmatrix} \mathbf{W}_1\nabla_\delta\mathbf{\Phi}\mathbf{B} \\ \mathbf{W}_2\mathbf{B} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{W}_1\nabla_\delta\mathbf{\Phi}\left(\mathbf{A}\boldsymbol{\delta}_k - \boldsymbol{\delta}_k\right) \\ \mathbf{W}_2\left(\mathbf{A}\boldsymbol{\delta}_k - \boldsymbol{\delta}_k\right) \end{bmatrix} - \begin{bmatrix} \mathbf{W}_1\Delta\boldsymbol{\tau}_c \\ \mathbf{W}_2\Delta\boldsymbol{\delta}_p \end{bmatrix} \right) = \mathbf{C}_0^T\lambda \tag{35}$$

where $\mathbf{C}_0$ contains the rows of $\mathbf{C}$ that correspond to constraints in the working set.

The main advantage of the proposed dynamic CA method is that it does not require the feedback of higher order output signals than for INCA or INDI itself. The control scheme allows for an incremental implementation of a discrete time onboard actuator dynamics model, which is limited to first order dynamics. Nevertheless, depending on the application the method might also be useful for systems that portray second order actuator dynamics.

B. Adaptive dynamic incremental nonlinear control allocation

As Algorithm 1 accounts for first order actuator dynamics, it also allows us to adapt the actuator dynamics model with each control increment. This work proposes a new adaptive dynamic incremental nonlinear control allocation (AD-INCA) method that includes online actuator dynamics model updates to increase actuator fault tolerance. For this, an actuator dynamics model is to be identified and, if needed, reduced to a first order model for the active set-based solver. The current paper assumes a recursive least squares (RLS) method as proposed by Fortescue et al. [15] to estimate the actuator dynamics model parameters. The main advantage of this estimator is that it does not require persistence of excitation. Instead, the parameters only update for sufficient measure of excitation, preventing covariance windup.

Algorithm 2 Recursive least squares parameter estimation with adaptive forgetting factor.

Given an initial covariance P_0, minimum forgetting factor $\Lambda_{\min}$ and information filter Σ_0.

$\hat{\boldsymbol{\delta}}_{k+1} = \min\left(\max\left(\boldsymbol{\xi}_k\hat{\boldsymbol{\theta}}_k, \underline{\boldsymbol{\delta}}\right), \overline{\boldsymbol{\delta}}\right)$ ▷ Output prediction

$\boldsymbol{\epsilon} = \boldsymbol{\delta}_{k+1} - \hat{\boldsymbol{\delta}}_{k+1}$ ▷ Output prediction error

$K_k = P_k\boldsymbol{\xi}_k^T/[1 + \boldsymbol{\phi}_k P_k\boldsymbol{\xi}_k^T]$ ▷ Least squares gain

$\Lambda_k = 1 - (1 - \boldsymbol{\xi}_k K_k)\boldsymbol{\epsilon}^2/\Sigma_0$ ▷ Forgetting factor

if $\Lambda_k \leq \Lambda_{\min}$ **then**

    Set $\Lambda_k = \Lambda_{\min}$

end if

$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + K_k\boldsymbol{\epsilon}$ ▷ Estimate update

$P_{k+1} = (1 - K_k\boldsymbol{\xi}_k)P_k/\Lambda_k$ ▷ Covariance update

The RLS algorithm taken from [15] is included in pseudo-code in Algorithm 2. In this algorithm, the parameter estimates are indicated by $\hat{\boldsymbol{\theta}}_k$ and the regressor vector is $\boldsymbol{\xi}_k$. The tuning parameters include a minimum forgetting factor $\Lambda_{\min}$, an information filter Σ_0 and initial covariance P_0. Even though the QP optimizer should prevent command

inputs beyond the saturation limits, this may not be the case for underdamped actuators that have not yet been identified properly. Therefore, it is necessary to include actuator constraints explicitly in the RLS estimator output prediction as captured in the first line of Algorithm 2. Note that in Algorithm 2, $\boldsymbol{\delta}_{k+1}$ is the measured quantity and that therefore the parameter estimator is lacking one time step behind the actual actuator dynamics. The resulting parameter estimates $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ are fed to the QP active-set based optimizer to compute the desired command input $\mathbf{u}_k$ based on Algorithm 1.

IV. Innovative control effectors (ICE) model

Fighter aircraft have been part of the battle space since the early 20-th century. Moreover, during the two world wars air superiority developed into a critical tool for establishing the upper hand on the ground. In order to counter the new threat from above, the Royal Air Force used early radio detection and ranging (radar) systems to provide themselves with early warning against incoming German bombing parties during World War II. Radar technology has advanced rapidly since, which makes aircraft nowadays prone to enemy detection, identification and engagement. Furthermore, reducing radar cross section (RCS) remains one of the key drivers in military aircraft design. A logical step herein is to eliminate unnecessary corners, making a tailless configuration an interesting candidate.

However, to maintain lateral control, high-manoeuvrability and add redundancy in control, tailless aircraft require unconventional, redundant control effectors. With this in mind, the innovative control effectors (ICE) program was conceived in 1993 to investigate the potential for new and innovative methods for stabilization and control of high performance, low all-aspect signature fighters [16]. Specifically, two tailless designs with novel control surfaces were investigated, a carrier-based and a land-based configuration. A schematic representation of the land-based version, hereafter referred to as 'ICE', is depicted in Fig. 2. Its actuator suite consists of the following components:

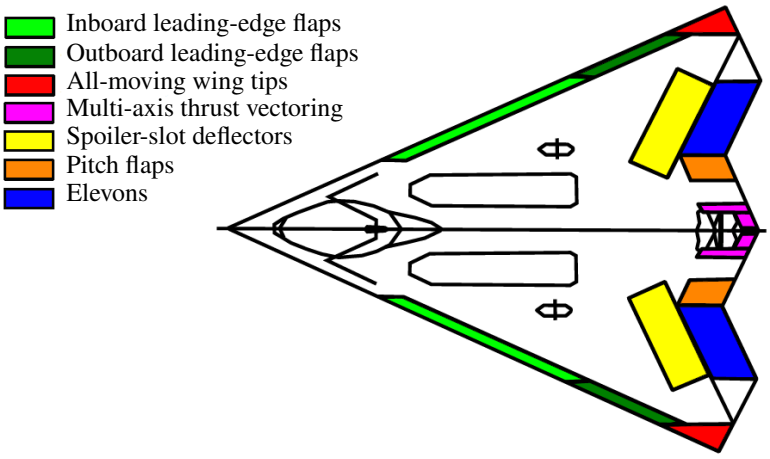


Fig. 2 A 2D schematic of the ICE aircraft [10].

- Leading-edge flaps: The ICE suite consists of two inboard and two outboard leading edge flaps (LEF). Differential use of the flaps provides directional-lateral control. The choice for four control surfaces adds redundancy and flexibility in control strategy.
- All-moving wingtips: Next, two all-moving wingtips (AMT) are included in the design. The control surfaces are constrained to trailing edge down (TED) movements for RCS reduction purposes. Yaw control magnitudes at large deflections are on the order of that produced by an F-16 rudder [16].
- Multi-axis thrust vectoring: In order to relieve the need for a tail even further, multi-axis thrust vectoring (MATV) was integrated in the ICE model. MATV provides both pitch and yaw control power. Unfortunately, yaw thrust vectoring is not very effective at high speed conditions due to the dependence of thrust vector control power on dynamic pressure, and structural limitations on nozzle deflections at high speeds.
- Spoiler-slot deflectors: Spoiler-slot deflectors (SSD) let airflow through the wings. The actuators allow for both yaw and roll control through differential lift and drag. As a drawback, control effectiveness of downstream surfaces is highly coupled with SSD deflection.
- Pitch flaps: These surfaces provide good pitch control effectiveness throughout the angle of attack (AoA) range. These are however, restricted to moving symmetrically in the same direction.
- Elevons: The elevons are located on the trailing edge outboard of the pitch flaps and can move independently from one another, allowing for both pitch and roll authority. The elevons can also produce a secondary-axis yawing moment through differential lift-induced drag.

Goals for these control concepts included 1) improved low signature characteristics; 2) improved high AoA effectiveness; 3) applicability to tailless fighters; 4) potential weight and drag reduction; 5) reduced hinge moments and 6) reduced susceptibility to aeroelastic effects [16]. As part of the project the ICE model was tested in a wind tunnel environment at Lockheed Martin Tactical Aircraft Systems, Aerodynamic Development Facility in Fort Worth, Texas [17]. The obtained aerodynamic model has been used for numerous studies into flight control.

Although the innovative actuator concepts could provide the required control forces and moments, the resulting over-actuated ICE configuration requires high complexity CA systems to exploit the aircraft's potential agility and redundancy in control. ICE exhibits highly cross-coupled nonlinear control effectiveness. Nevertheless, Matamoros [10] applied INCA to effectively utilize inherent cross-coupling.

A. ICE control effectiveness modeling

Van der Peijl [18] captured the ICE aerodynamics with a simplex-spline model. For each degree of freedom $i = l, m, n, X, Y, Z$ the dimensionless moments and forces coefficients were represented by a set of B-splines $C^s_{i_n}$ defined over a simplex triangulation with 0-th-order continuity. The sum of the B-splines represents the ICE aerodynamics

$$
\begin{aligned}
C_i =& C^s_{i_1}(\alpha, M) + C^s_{i_2}(\alpha, \beta, M) + C^s_{i_3}(\alpha, \delta_{\text{ls}}, \delta_{\text{le}}, M) + C^s_{i_4}(\alpha, \delta_{\text{rs}}, \delta_{\text{re}}, M) + C^s_{i_5}(\alpha, \beta, \delta_{\text{lfi}}) + C^s_{i_6}(\alpha, \beta, \delta_{\text{rfi}}) \\
&+ C^s_{i_7}(\alpha, \beta, \delta_{\text{lf}}, \delta_{\text{lfo}}, M) + C^s_{i_8}(\alpha, \beta, \delta_{\text{rf}}, \delta_{\text{rfo}}, M) + C^s_{i_9}(\alpha, \delta_{\text{lfo}}, \delta_{\text{la}}) + C^s_{i_{10}}(\alpha, \delta_{\text{rfo}}, \delta_{\text{ra}}) + C^s_{i_{11}}(\alpha, \delta_{\text{la}}, \delta_{\text{le}}) \\
&+ C^s_{i_{12}}(\alpha, \delta_{\text{ra}}, \delta_{\text{re}}) + C^s_{i_{13}}(\alpha, \delta_{\text{rs}}, \delta_{\text{ls}}, \delta_{\text{pf}}, M) + C^s_{i_{14}}(\alpha, \beta, \delta_{\text{la}}) + C^s_{i_{15}}(\alpha, \beta, \delta_{\text{ra}}) + C^s_{i_{16}}(\alpha, \beta, \delta_{\text{ls}}) \\
&+ C^s_{i_{17}}(\alpha, \beta, \delta_{\text{rs}}) + \frac{pb}{2V} C^s_{i_{18}}(\alpha, M) + \frac{q\bar{c}}{2V} C^s_{i_{19}}(\alpha, M) + \frac{rb}{2V} C^s_{i_{20}}(\alpha, M)
\end{aligned}
\tag{36}
$$

with α the angle of attack, β the sideslip angle, M the Mach number, $\bar{c}$ the mean aerodynamic chord, b the wing span and V the true airspeed. The body angular rates are given by p, q and r. Lastly, the actuators δ_i are included in Table 1. For use in INCA, [10] derived a CEJ from the aerodynamic model by taking the partial derivative of each term with respect to $\boldsymbol{\delta}$

$$
\nabla_{\boldsymbol{\delta}} \Phi(\alpha, \beta, M, \boldsymbol{\delta}) = \begin{bmatrix}
\sum_{j=1}^{20} \frac{\partial C^s_{l_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_1} & \sum_{j=1}^{20} \frac{\partial C^s_{l_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_2} & \cdots & \sum_{j=1}^{20} \frac{\partial C^s_{l_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_{13}} \\
\sum_{j=1}^{20} \frac{\partial C^s_{m_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_1} & \sum_{j=1}^{20} \frac{\partial C^s_{m_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_2} & \cdots & \sum_{j=1}^{20} \frac{\partial C^s_{m_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_{13}} \\
\sum_{j=1}^{20} \frac{\partial C^s_{n_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_1} & \sum_{j=1}^{20} \frac{\partial C^s_{n_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_2} & \cdots & \sum_{j=1}^{20} \frac{\partial C^s_{n_j}(\mathrm{x}, \boldsymbol{\delta})}{\partial \delta_{13}}
\end{bmatrix}. \tag{37}
$$

For the MATV a CEJ can be derived analytically [10, p. 55].

B. ICE actuator dynamics and constraints

The actuator dynamics can be modeled as second order transfer functions. Both a low and a high-bandwidth transfer function were chosen, based on the type of actuator.

$$
H_l(s) = \frac{1800}{s^2 + 118s + 1800} \tag{38}
$$

$$
H_h(s) = \frac{4000}{s^2 + 140s + 4000} \tag{39}
$$

The position and rate constraints for each actuator type are included in Table 1.

Table 1 Dynamic properties of the ICE actuators.

Control effector	Notation	Position limits [deg]	Rate limit [deg/s]	Dynamics
Inboard LEF	$\delta_{\text{lfi}}, \delta_{\text{rfi}}$	[0, 40]	40	$H_l(s)$
Outboard LEF	$\delta_{\text{lfo}}, \delta_{\text{rfo}}$	[-40, 40]	40	$H_l(s)$
AMT	$\delta_{\text{la}}, \delta_{\text{ra}}$	[0, 60]	150	$H_h(s)$
Elevons	$\delta_{\text{le}}, \delta_{\text{re}}$	[-30, 30]	150	$H_h(s)$
SSD	$\delta_{\text{ls}}, \delta_{\text{rs}}$	[0, 60]	150	$H_h(s)$
PF	δ_{pf}	[-30, 30]	150	$H_h(s)$
MTV	$\delta_{\text{ptv}}, \delta_{\text{ytv}}$	[-15, 15]	150	$H_h(s)$

V. Flight control system design

A. INCA-based flight control system

The current research builds on the INCA-based flight control system (FCS), depicted in Fig. 3. The inner control loop consists of an angular acceleration inversion loop, together with a CA optimization function, being the active set-based QP solver. Perfect knowledge is assumed of the angular accelerations $\dot{\boldsymbol{\omega}}$. The angular acceleration inversion loop is described in detail in [10], and will not be further elaborated upon in this work.

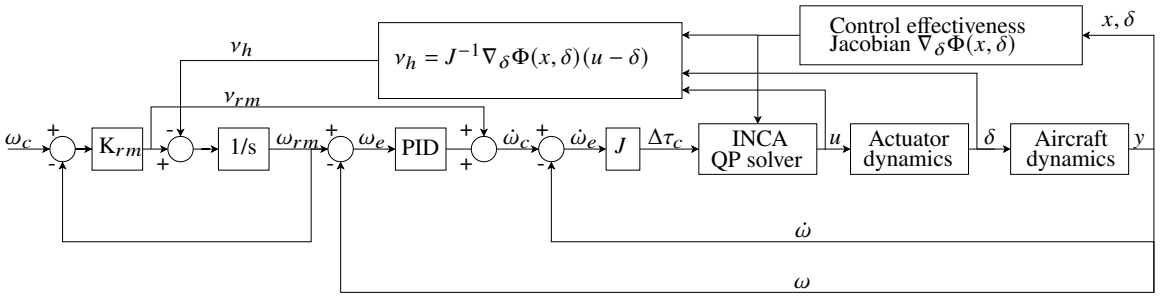


Fig. 3 INCA ICE FCS block diagram [10].

1. INCA and PCH

To omit command saturation problems, Matamoros [10] applied a pseudo-control hedge (PCH) within the INCA-based FCS. The PCH is defined by the difference in virtual control input $\boldsymbol{\nu}$ or $\dot{\boldsymbol{\omega}}_{c,k-1}$ and the estimated obtained virtual control $\hat{\boldsymbol{\nu}}$ or $\hat{\dot{\boldsymbol{\omega}}}_k$. Assuming that no unoptimalities originate from the QP solver, the PCH only depends on actuator dynamics and saturation limits. For INCA the PCH $\boldsymbol{\nu}_h$ can be computed as

$$\boldsymbol{\nu}_h = \dot{\boldsymbol{\omega}}_{c,k-1} - \hat{\dot{\boldsymbol{\omega}}}_k = J^{-1}\nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi}(\mathbf{x}_{k-1}, \boldsymbol{\delta}_{k-1})(\mathbf{u}_{k-1} - \boldsymbol{\delta}_k). \tag{40}$$

Next, the hedge is fed back into a reference model as a compensation signal. The reference model $\boldsymbol{\nu}_{\mathrm{rm}}$ is given by

$$\boldsymbol{\nu}_{\mathrm{rm}} = K_{\mathrm{rm}}(\boldsymbol{\omega_c} - \boldsymbol{\omega}_{\mathrm{rm}}) \tag{41}$$

with K_{rm} a diagonal matrix. The hedged reference signal fed as a control input into the attitude control system is the state vector of the reference model

$$\boldsymbol{\omega}_{\mathrm{rm}} = \frac{1}{\mathrm{s}}\left(\boldsymbol{\nu}_{\mathrm{rm}} - \boldsymbol{\nu}_h\right). \tag{42}$$

As indicated, the PCH was introduced to act in the case of actuator saturation by avoiding unfeasible virtual command signals. However, the underlying problem with INCA is not primarily actuator saturation limits. These are hard-coded in the QP solver and cannot be violated by the constrained optimization problem. Instead, the solver cannot

catch up with the virtual control input references as the optimal command inputs are not achieved due to actuator dynamics delays. With the PCH, the outer loop signal is hedged or scaled to preserve closed loop stability with these system delays present. The current study provides an alternative approach to cope with inner loop time delays originating from the actuator dynamics.

B. (A)D-INCA-based flight control system

The novel FCS designs with D-INCA and AD-INCA are depicted in Fig. 4 with in red the AD-INCA specific system identification loop. With (A)D-INCA, the need for a PCH is omitted. Instead, (A)D-INCA produces more aggressive command inputs to mitigate actuator dynamics delays through an onboard actuator dynamics model. In addition, (A)D-INCA imposes the physical saturation limits of the actuators as QP output constraints on the expected physical deflections $\boldsymbol{\delta}_c$, rather than on the command inputs $\mathbf{u}_k$ as is the case with INCA. It is expected that this results in less conservative CA solutions and higher system performance.

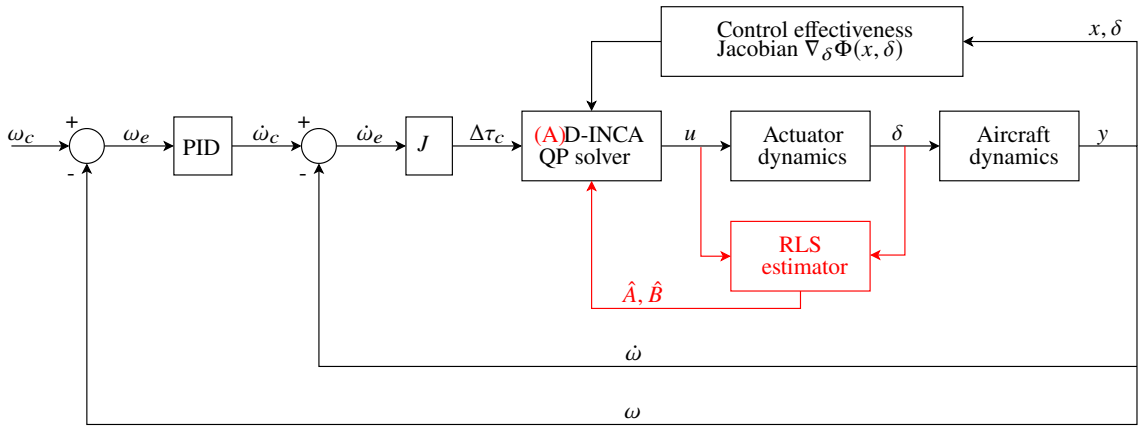


Fig. 4 (A)D-INCA ICE FCS block diagram.

1. PID tuning

The adopted PID gains for each FCS are the same and taken from [10]. For the INCA-based FCS a reference model gain K_{rm} was tuned. The chosen values are stated in Table 2.

Table 2 PID and reference model gains for ω control.

	p	q	r
K_P	6.5	6.5	5.8
K_I	0.0	0.0	0.0
K_D	0.5	0.5	0.5
K_{rm}	12	12	12

2. (A)D-INCA QP solver

The CA optimizer is based on the active-set based QP solver described by Algorithm 1. The maximum number of iterations that was chosen $N = 100$ and the actuator preference vector $\boldsymbol{\delta}_p = \mathbf{0}$. All the weights were taken from the static method in [19, p. 16]. The weights on the CA error are $\mathbf{W}_1 = I$. For the penalty on $\boldsymbol{\delta}$, $\mathbf{W}_2 = \text{diag}([\overset{\text{lfi}}{1}\ \overset{\text{lfo}}{1}\ \overset{\text{la}}{0.5}\ \overset{\text{le}}{1}\ \overset{\text{ls}}{10}\ \overset{\text{pf}}{0.5}\ \overset{\text{rfi}}{1}\ \overset{\text{rfo}}{1}\ \overset{\text{ra}}{0.5}\ \overset{\text{re}}{1}\ \overset{\text{rs}}{10}\ \overset{\text{ptv}}{1}\ \overset{\text{ytv}}{1}])$. To prevent convergence to local optima of the spoiler-slot deflectors due to non-convex SSD CEJ terms, the weights on the SSDs are augmented as follows

$$W_{1_{\text{SSD}}}(\alpha) = [-0.25 + 0.25\exp(1.6\alpha)]^{-1}. \tag{43}$$

3. RLS estimator

The actuator dynamics parameter estimator is described by Algorithm 2. For ICE control, the adopted parameter estimates include $\hat{\boldsymbol{\theta}}_k = [\hat{a}_{1,k}, \hat{a}_{2,k}, \hat{b}_{1,k}, \hat{b}_{2,k}]$ and the regressors include $\boldsymbol{\xi}_k = [\boldsymbol{\delta}_k, \boldsymbol{\delta}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k-1}]$. As the actuator behavior is governed by second order dynamics, a second order model needs to be estimated as first order model parameter estimates would diverge after observing second order phenomena. The discrete time second order model to be estimated is the following difference equation

$$\boldsymbol{\delta}_{k+1} = a_{1,k}\boldsymbol{\delta}_k + a_{2,k}\boldsymbol{\delta}_{k-1} + b_{1,k}\mathbf{u}_k + b_{2,k}\mathbf{u}_{k-1}. \tag{44}$$

The tuning parameters include a minimum forgetting factor $\Lambda_{\text{min}} = 0.25$, an information filter $\Sigma_0 = 10$ and initial covariance $P_0 = 0.01 \cdot I$. Next, the second order model is reduced to a first order lag through a Matlab built-in balanced truncation routine. The resulting parameter estimates $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ are fed to the AD-INCA QP solver. Moreover, to prevent extreme command inputs for input scalars $\hat{B}_i \in \hat{\mathbf{B}}$ approaching zero, $u_k = 10$ deg is forced on actuators with large value (>1000 deg) command input constraints (i.e., with low command input effectiveness or being stuck).

VI. Simulations and results

Simulations were conducted for the INCA, D-INCA and AD-INCA control schemes. Specifically, for these simulations a Matlab Simulink R2023b environment was used. Matamoros [10] created four different reference trajectories with three different outer loop inversion controllers to obtain $\boldsymbol{\omega_c}$. These outer loop controllers are described in detail in [10] and will not be further elaborated upon in the this work. Three of the four reference trajectories were adopted in the current study and are briefly discussed in Section VI.A.

A. Challenging flight trajectories

1. Maneuver A

The barrel roll maneuver uses a sideslip compensation outer loop with direct roll and pitch body rate control inputs. It consists of constant pitch and roll rate inputs, synchronized to complete two barrel rolls and exit the maneuver with the initial heading and zero flight path angle. The simulation time is 50 s.

2. Maneuver B

Maneuver B is the most demanding test performed in this work, consisting of a sequence of aggressive aerobatic maneuvers: aileron roll, asymmetric looping with sideslip, half 8-point hesitation roll and Immelmann turn. These maneuvers are done with a sideslip compensation outer loop with direct roll and pitch body rate control inputs, same as with maneuver A. The aileron roll consists of a roll input of 180 deg/s resulting in two full turns in 4 s. The asymmetric looping consists of a pitch rate input of 19 deg/s, yielding a full looping in roughly 19 s. During the first 10 s of the looping a 5 deg sideslip input is given, which returns to 0 deg for the last 9 s of the looping. This results in lateral displacement during the looping. The half 8-point hesitation roll consists of four quick roll rate step inputs of 90 deg/s with a duration of 0.5 s each and with a separation of 1 s, resulting in four quick steps in roll of 45 deg each, yielding a 180 deg roll angle. Lastly, a downward half-looping follows ending the maneuver in straight and level fight, flying in the opposite direction at a lower altitude than initially. The simulation time is 50 s.

3. Maneuver C

This maneuver is intended to test the behavior of the FCS at high AoA and high sideslip, because most nonlinearities and interactions occur within these regions of the flight envelope. The simulations were performed using the aerodynamic inversion outer loop with the aerodynamic angles α and β and the roll angle ϕ as control inputs. The reference consists of a doublet α input of $\approx$ 40 deg followed by a doublet β input of $\approx$ 25 deg, while keeping the roll angle reference at 0 deg. The simulation time is 30 s.

B. Measures of performance

For each reference maneuver, performance metrics are evaluated for D-INCA and AD-INCA. The results are compared to those obtained with the baseline INCA. The measures of performance that are regarded include the following errors and their root mean square (RMS) values:

- CA error: evaluates the ability to attain the control-induced moments required by the high-level FCS

$$\mathrm{RMS}(\boldsymbol{\tau} - \boldsymbol{\tau}_c). \tag{45}$$

- Tracking error: evaluates the overall ability of the CA method to track the reference command inputs∗

$$\begin{aligned} &\mathrm{RMS}(\boldsymbol{\delta} - \boldsymbol{\delta}_c) \quad ((\mathrm{A})\mathrm{D} - \mathrm{INCA}), \\ &\mathrm{RMS}(\boldsymbol{\delta} - \mathbf{u}) \quad (\mathrm{INCA}), \end{aligned} \tag{46}$$

  with $\boldsymbol{\delta}_c$ the commanded physical inputs. These are the expected physical inputs based on the command inputs, the current actuator positions and the onboard actuator dynamics model.

- Control effort: evaluates the amount of deflection of the control effectors required

$$\mathrm{RMS}(\boldsymbol{\delta} - \boldsymbol{\delta}_p). \tag{47}$$

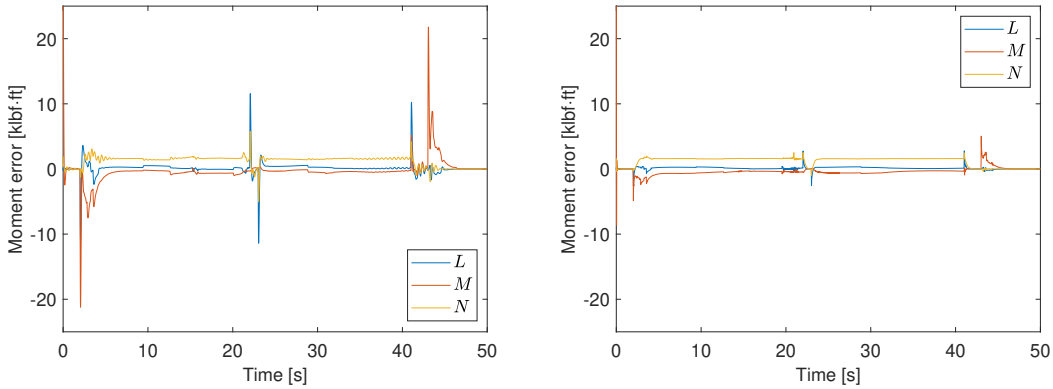
Additionally, the outer loop tracking performance is investigated by comparing the references to the attained trajectories.

C. D-INCA

This section includes the results obtained with the D-INCA ICE FCS for trajectories A, B and C. The results are compared to those found for INCA.

1. Control allocation performance with D-INCA

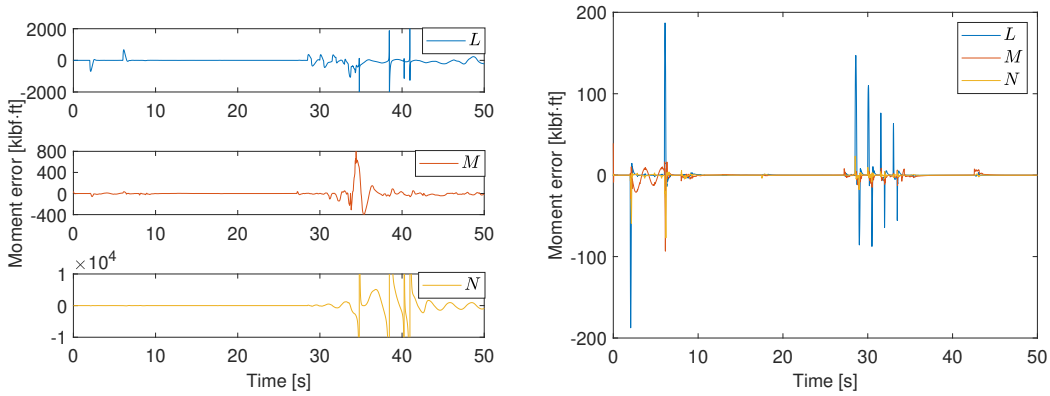
Firstly, CA performance will be determined in terms of tracking the pseudo-control (torque) input. Note that the CA errors for INCA are determined after virtual control input $\boldsymbol{\nu} = \dot{\boldsymbol{\omega}}_c$ augmentation by the PCH. This PCH scales the virtual control input signal based on the difference between input command and actuator deflection. Whereas, for D-INCA (and AD-INCA) the commanded torque aims to achieve the outer loop rate commands. To mitigate this problem, the unbiased outer loop tracking performance metrics are included in Section VI.C.3.



(a) INCA **(b) D-INCA**

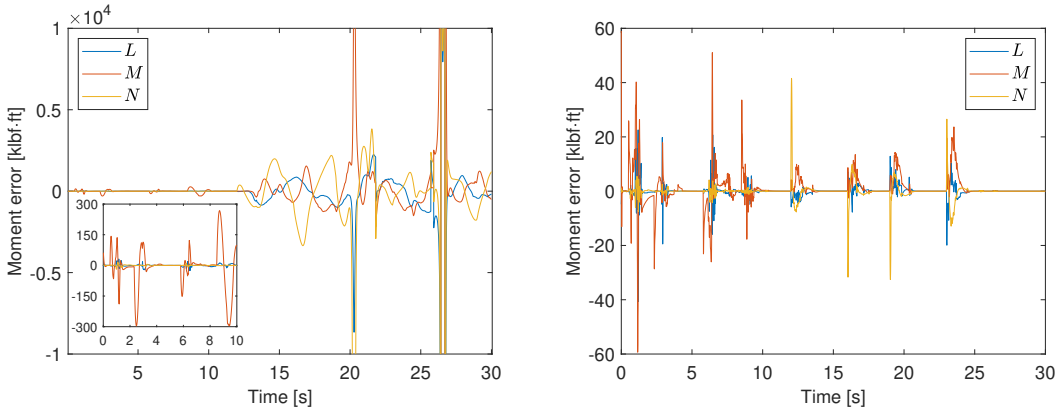
Fig. 5 Error in attained control torque for maneuver A.

∗These definitions differ between INCA and (A)D-INCA, because for the latter, $\mathbf{u}$ is merely an auxiliary signal to reach $\boldsymbol{\delta} = \boldsymbol{\delta}_c$.



(a) INCA **(b) D-INCA**

Fig. 6 Error in attained control torque for maneuver B.



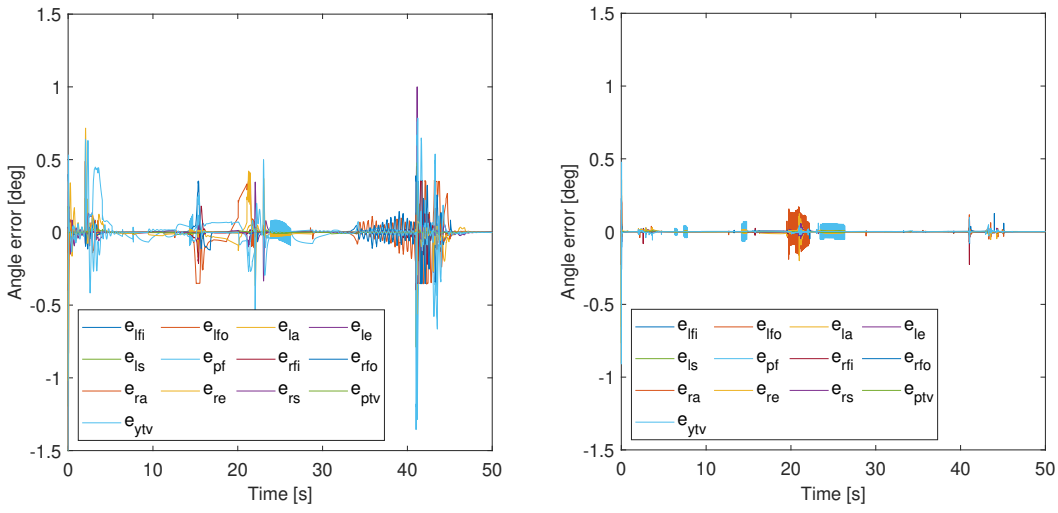
(a) INCA **(b) D-INCA**

Fig. 7 Error in attained control torque for maneuver C.

Maneuver A can be tracked by both INCA and D-INCA. However, errors in control moment $\boldsymbol{\tau} - \boldsymbol{\tau}_c$ significantly decrease with D-INCA. In particular, temporary peaks in CA error are reduced by compensating for the actuator dynamics. Maneuver B is the most challenging trajectory that was flown. INCA cannot follow the reference maneuver which results in an unstable system response. Especially, during the sequence of aggressive roll maneuvers around $t = 30$ s. Whereas, D-INCA can still keep the CA error within certain boundaries. Maneuver C gives similar information as B. In particular, INCA cannot track the yaw and pitch torque commands at the challenging sideslip doublet reference. The RMS CA errors are included in Table 4 in the Appendix. It is found that the average CA error decreases with D-INCA. Not only for the case in which INCA cannot track the reference, but also for maneuver A that can be tracked by both INCA and D-INCA. Especially, during maneuver A pitch and roll torque errors reduce with D-INCA, as the reference trajectory consists of aggressive pitch and roll maneuvers. On the other hand, yaw tracking performance is fairly similar between the two adopted methods.

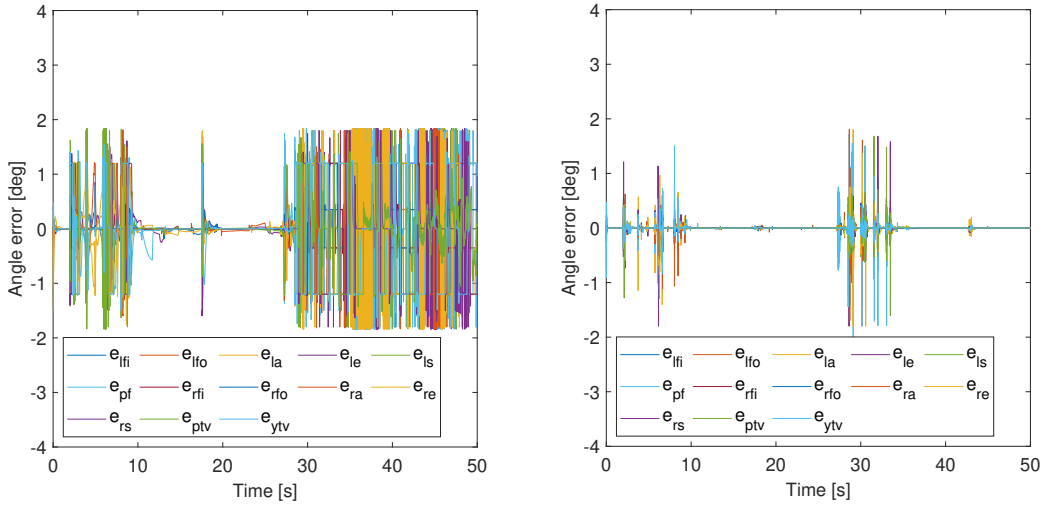
2. Actuator deflections in D-INCA

This section will dive into actuator deflections attained with D-INCA and the errors in actuator deflection. The tracking errors and control effort metrics are also depicted. The tracking errors obtained for maneuver A are in line with the CA errors seen in Fig. 5. The actuator position errors with D-INCA decrease compared to those with the baseline INCA implementation. With maneuver B, it can be seen that the magnitude of the errors stays the same. This is because the saturation limits together with the sample time bound the tracking errors. However, the density of errors is much lower for D-INCA than for INCA. With D-INCA, the onboard actuator dynamics model largely follows the actual actuator dynamics. However, it cannot predict the actuator deflections perfectly throughout the entire simulation. This shows the limitations of the discrete first order onboard actuator dynamics model. Lastly, for maneuver C, INCA and D-INCA show a similar relationship as for maneuvers A and B. The RMS tracking errors for each maneuver are included in Table 5 in the Appendix. From the entries in this table it becomes clear that INCA cannot follow the reference actuator deflection in maneuver B and C. The tracking errors significantly reduce with the D-INCA QP solver. Table 6 in the Appendix states the control effort required for each maneuver. Since INCA cannot track trajectories B and C, the control effort for these maneuvers is less interesting as performance metric. The control effort for the maneuver A that can be tracked by both methods, is not significantly different. Hence, in terms of control effort, there is no reason to prefer one method over the other.



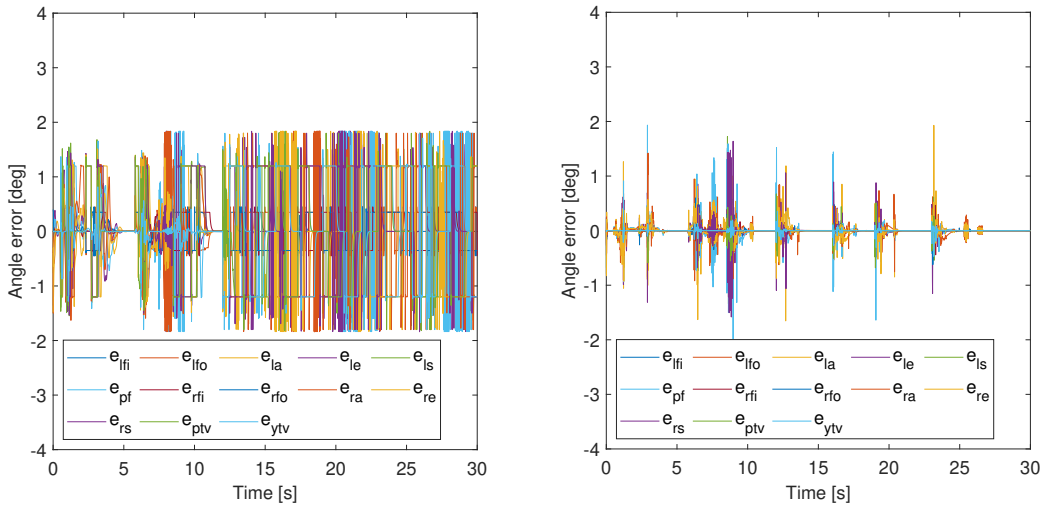
(a) INCA **(b) D-INCA**

Fig. 8 Error in attained actuator deflection for maneuver A.



(a) INCA **(b) D-INCA**

Fig. 9 Error in attained actuator deflection for maneuver B.

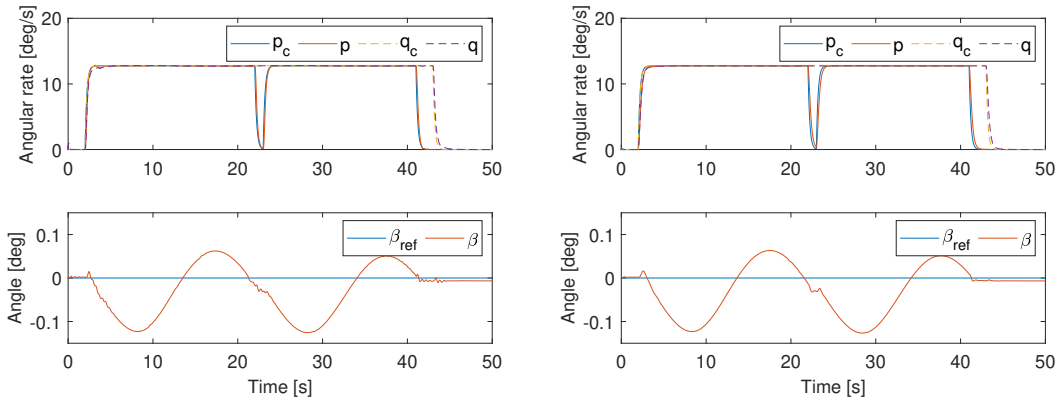


(a) INCA **(b) D-INCA**

Fig. 10 Error in attained actuator deflection for maneuver C.

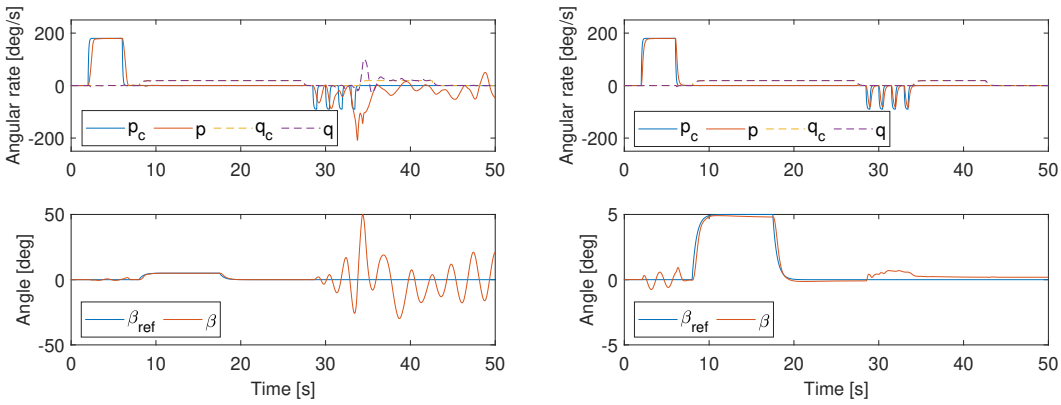
3. Outer loop tracking performance

Figures 11 to 13 show the tracking of the outer loop references. Both INCA and D-INCA can track maneuver A. The challenging maneuver B cannot be tracked by INCA. Whereas, D-INCA follows the signal with great precision. Similarly, only D-INCA can follow the high sideslip reference in maneuver C. This is in line with the CA and tracking errors found. The simulated translational trajectories are included in Fig. 32 in the Appendix.



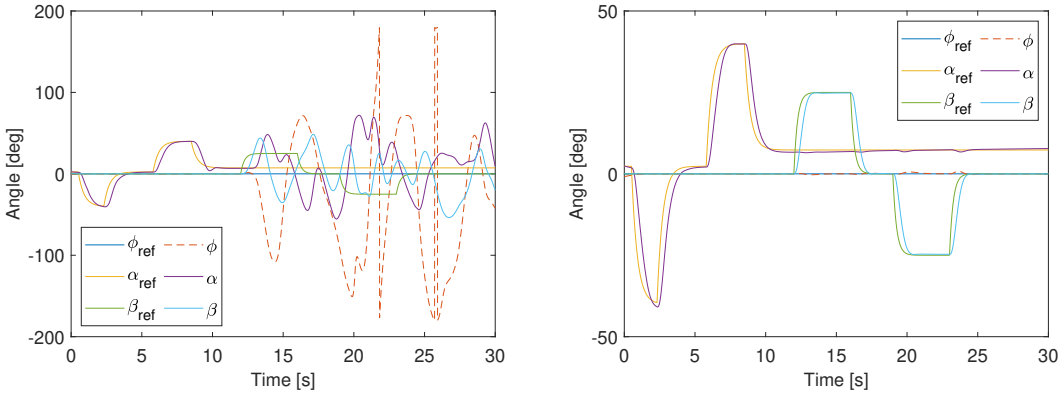
(a) INCA **(b) D-INCA**

Fig. 11 Outer loop reference tracking maneuver A.



(a) INCA **(b) D-INCA**

Fig. 12 Outer loop reference tracking maneuver B.

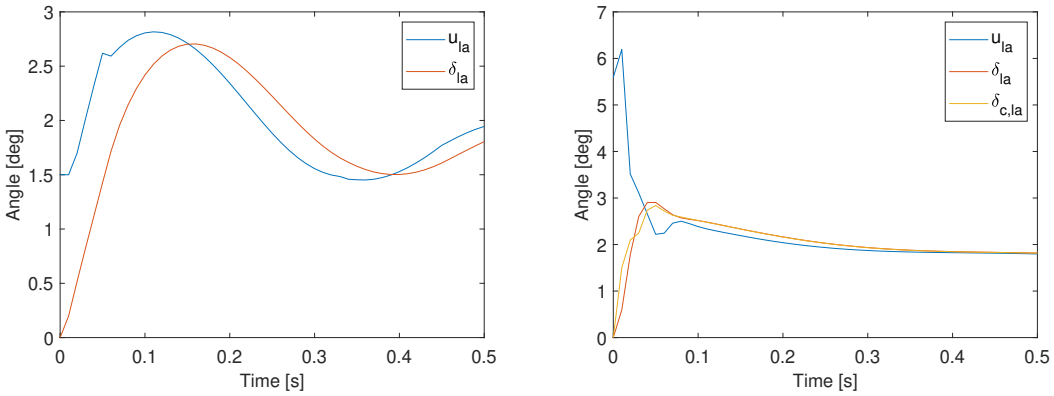


(a) INCA **(b) D-INCA**

Fig. 13 Outer loop reference tracking maneuver C.

4. Command inputs and model limitations

This section will briefly demonstrate the difference between INCA and D-INCA from a command input perspective. Consider the initial control of the left all-moving wingtip for maneuver A in Fig. 14. The time evolution of the command input u_{la}, physical control input δ_{la} and the commanded physical control input $\delta_{c,\text{la}}$ are included. At $t = 0$ s, the command input with INCA is much smaller than it is with D-INCA. This is because the rate limit is 150 deg/s, which does not allow for a command input greater than 1.5 deg from the zero start position at the given sample time $\Delta t = 0.01$ s. Limited by this QP constraint, the actuator by far does not reach its physical constraint. Instead, a deflection rate of 20 deg/s is observed. After initialization, the actuator dynamics prevent the AMT to converge to a steady state.



(a) INCA **(b) D-INCA**

Fig. 14 Command input and position of the left all-moving wing tip for maneuver A.

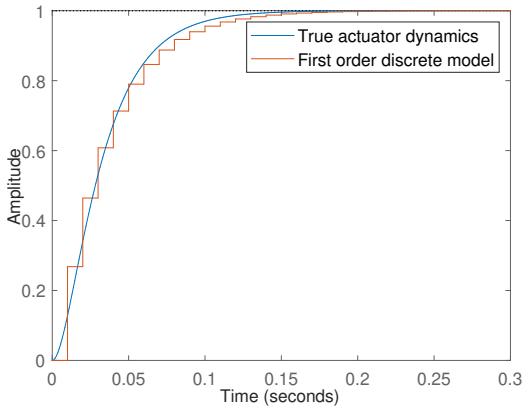


Fig. 15 Step response for actuator dynamics and onboard first order discrete model.

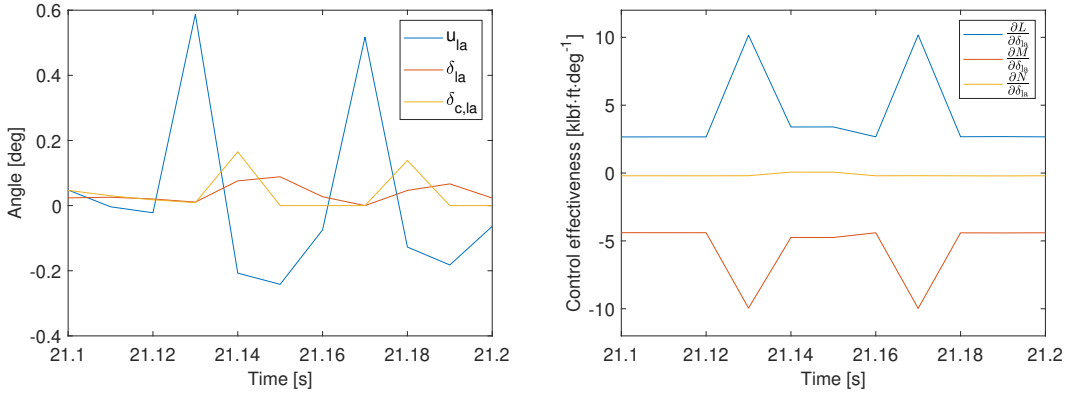
D-INCA on the other hand, produces a command input for which the expected physical deflection matches the 150 deg/s rate constraint. Also for D-INCA, the real physical constraint is not obtained after the first sample time, because of a mismatch between the first order discrete time model and the true actuator dynamics. More specifically, this can

be explained by the differences in step responses as depicted in Fig. 15. The zero-order-hold reduced-order-model does not align with the real second order actuator dynamics at its own sample time. After one sample time $t = 0.01$ s, the model overestimates the actuator response. Hence, the QP solver will produce a smaller command input than needed for the optimal deflection. The expected and obtained actuator positions converge during the subsequent sample times.

To conclude, the D-INCA QP solver is much less conservative as it imposes output constraints on the actuator deflections rather than the command inputs. A higher order actuator dynamics model will likely yield better results. Nevertheless, the current implementation already outperforms the conservative INCA control solution.

5. Command input chatter

A great portion of the tracking errors in Figs. 8 to 10 can be pinpointed to command input chatter. It was found that both INCA and D-INCA suffer from this problem. An example of the phenomenon is depicted in Fig. 16 for D-INCA, where command input chatter at the left all-moving wingtip occurs during maneuver A after $t = 21.13$ s. This chatter is induced by discontinuity in the adopted CEJ model, as the command input follows a jump in the CEJ.



(a) Command input and physical input **(b) CEJ**

Fig. 16 Chatter in D-INCA LAMT control solution for maneuver A.

Particularly, for extremely challenging reference trajectories the chatter problem becomes more prominent as the QP solution space is reduced. This chatter affects the control solution as the first order actuator dynamics model cannot accurately predict the increased second order actuator behavior. This can also be observed in Fig. 16a as a mismatch arises between δ_{la} and $\delta_{c,\text{la}}$. Additionally, discontinuity in the CEJ model increases the chance of finding local optima. This makes it hard to predict the total potential of D-INCA without a continuous CEJ model. Last but not least, input chatter poses a problem for practical implementation of INCA due to structural limitations of the actuators. It would be worthwhile to investigate whether a continuous CEJ model would prevent the chatter and increase closed loop performance.

6. Control effectiveness Jacobian model mismatch robustness

Matamoros [10] showed INCA's significant robustness against CEJ mismatch. In order to test D-INCA's CEJ mismatch robustness, a grid of maneuver A simulations was conducted for different CEJ offsets and scaling errors. The scale factor was varied between 0.6 and 1.4 with increments of 0.1. The model offset is determined as a percentage of the absolute maximum CEJ values obtained for the nominal simulation with INCA. The offset was varied between -30% and 30% with increments of 5%. The performance metrics for different CEJ mismatches are shown in Figs. 17 to 19. In terms of CA errors and tracking errors, D-INCA's robustness against Jacobian mismatch is higher than INCA's.

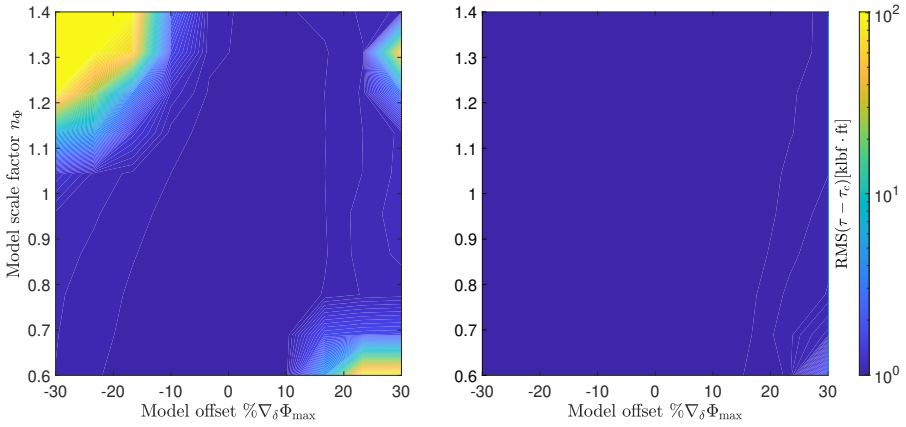


Fig. 17 CA error robustness for INCA (left) and D-INCA (right).

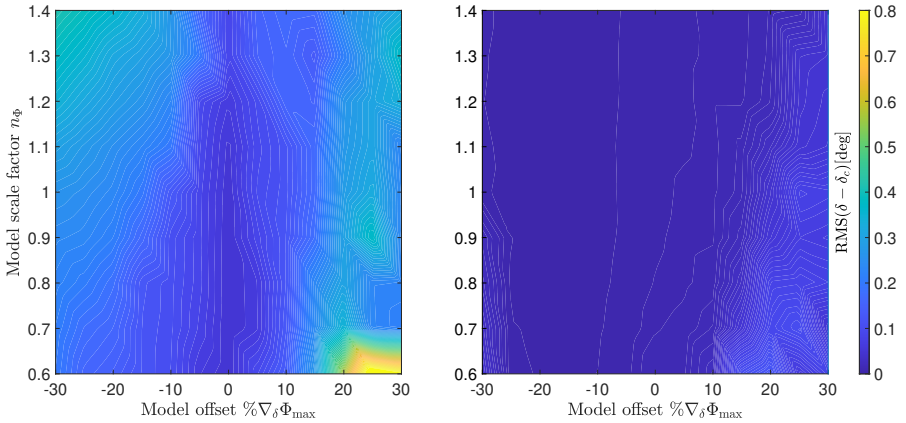


Fig. 18 Tracking error robustness for INCA (left) and D-INCA (right).

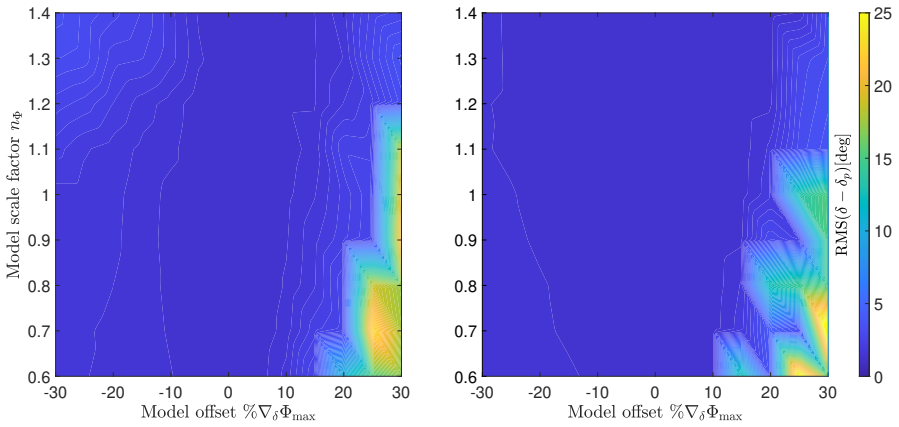


Fig. 19 Control effort robustness for INCA (left) and D-INCA (right).

D. AD-INCA

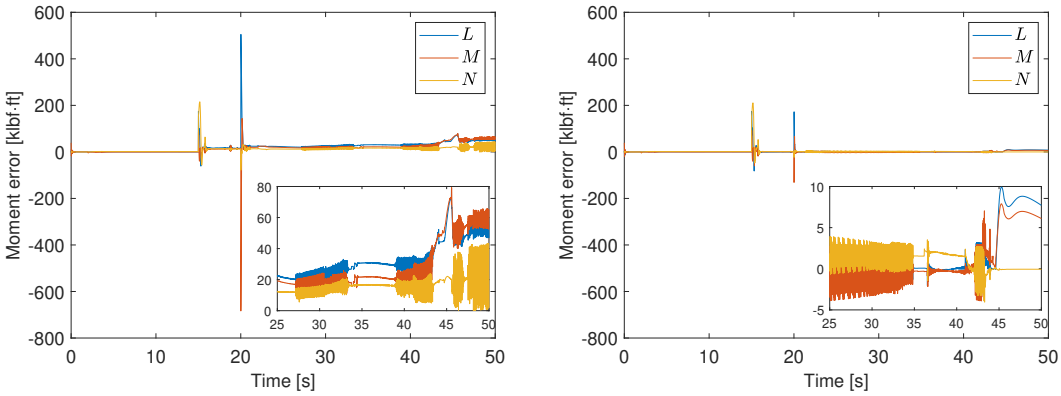
In addition to the simulations with nominal actuators, the ICE FCS was tested under actuator failure events, for example due to loss of hydraulic pressure. The sequence of failure events adopted is included in Table 3. Failures were forced on the elevons and all-moving wingtips. In addition, the right spoiler-slot deflector is stuck. This is an interesting failure as it will have large impact on downstream actuator effectiveness. The multi-axis thrust vector control, pitch flaps and leading edge flaps are left intact. This should provide the aircraft with both pitch and yaw control authority. Roll references however, are likely more difficult to attain. The results for D-INCA and AD-INCA are compared. Results for INCA are not depicted in the applicable figures as earlier simulations showed D-INCA to be superior during nominal actuator behavior. Nevertheless, performance metrics for INCA are included in Tables 7 to 9 in the Appendix.

Table 3 Tested actuator failure sequence.

Actuator	Event	Time [s]
δ_{la}	underdamped: $H(s) = \dfrac{4000}{s^2 + 40s + 4000}$	0
δ_{le}	stuck in 30 deg position	20
δ_{ra}	stuck in 0 deg position	20
δ_{re}	underdamped: $H(s) = \dfrac{4000}{s^2 + 100s + 4000}$	10
δ_{rs}	stuck in 30 deg position	15

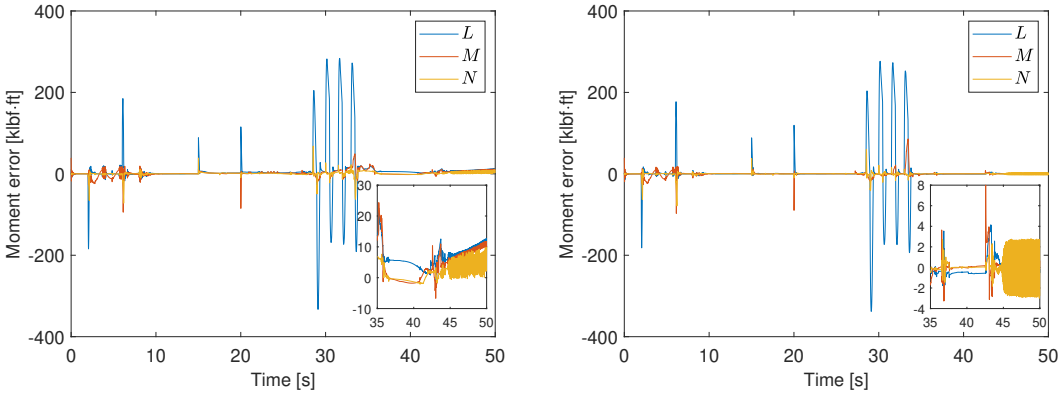
1. Control allocation performance with AD-INCA

Firstly, CA performance will be determined in terms of tracking the pseudo-control input. The time evolutions of the CA errors are depicted in Figs. 20 to 22. AD-INCA and D-INCA perform very similar under the given actuator failure events. However, peaks in CA error for maneuver A are higher for D-INCA than for AD-INCA. Also, higher CA errors at the end of the simulation time are observed for each maneuver.



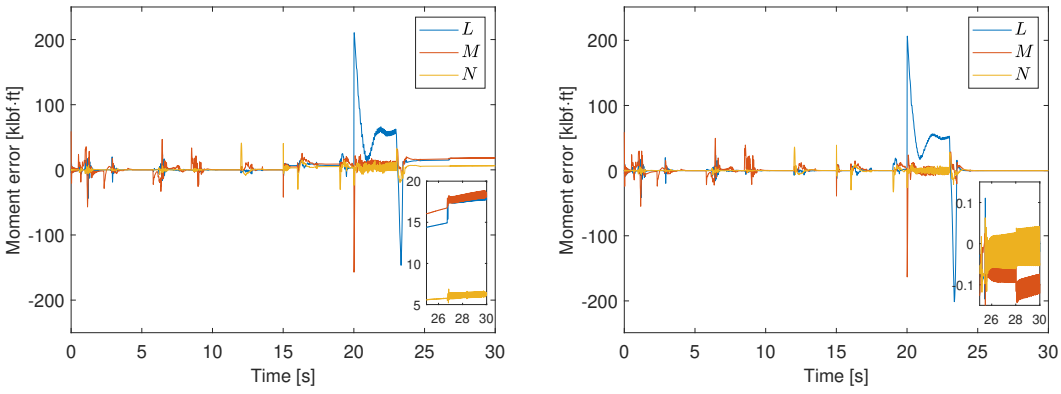
(a) D-INCA (b) AD-INCA

Fig. 20 Error in attained control torque for maneuver A.



(a) D-INCA (b) AD-INCA

Fig. 21 Error in attained control torque for maneuver B.



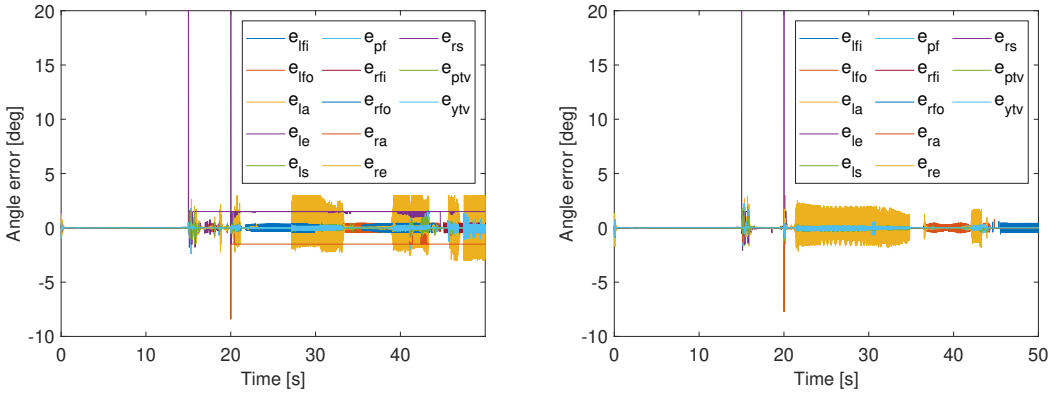
(a) D-INCA (b) AD-INCA

Fig. 22 Error in attained control torque for maneuver C.

The RMS CA errors for the actuator failure sequence for the three maneuvers are included in Table 7 in the Appendix. On average, the RMS CA error is reduced by 34% with the AD-INCA implementation. Especially, in maneuver A the adaptive actuator dynamics model proves its advantage over D-INCA. This is logical as the elevons and AMTs, providing significant roll and pitch authority, are stuck during high pitch and roll rate references. During maneuver B and C AD-INCA also performs slightly better. Nevertheless, D-INCA already exhibits robustness for actuator failures. Since the actuator positions are fed back, the actuator errors are bounded by the saturation limits and the sample time.

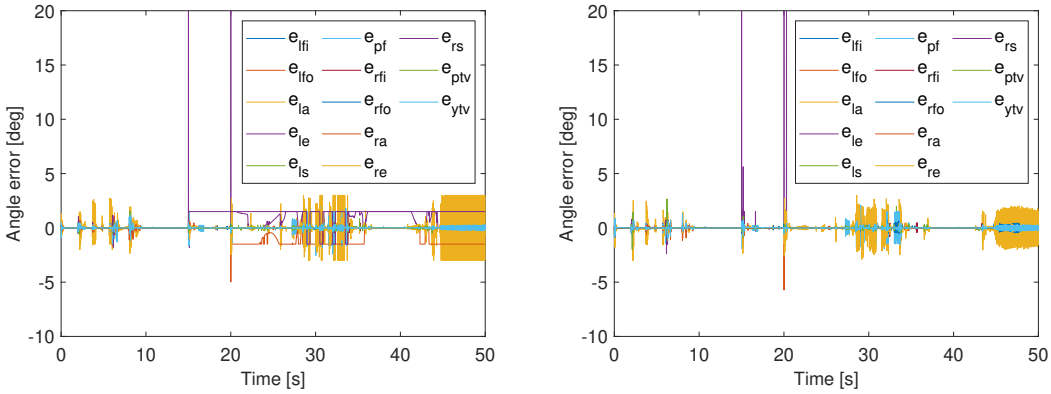
2. Actuator deflections in AD-INCA

Next, the actuator errors are depicted in Figs. 23 to 25. Although actuator errors seem similar between the two CA schemes, some additional steady state errors are observed with D-INCA for each maneuver. These start after the actuator failures and are mitigated by AD-INCA through actuator dynamics parameter estimation.



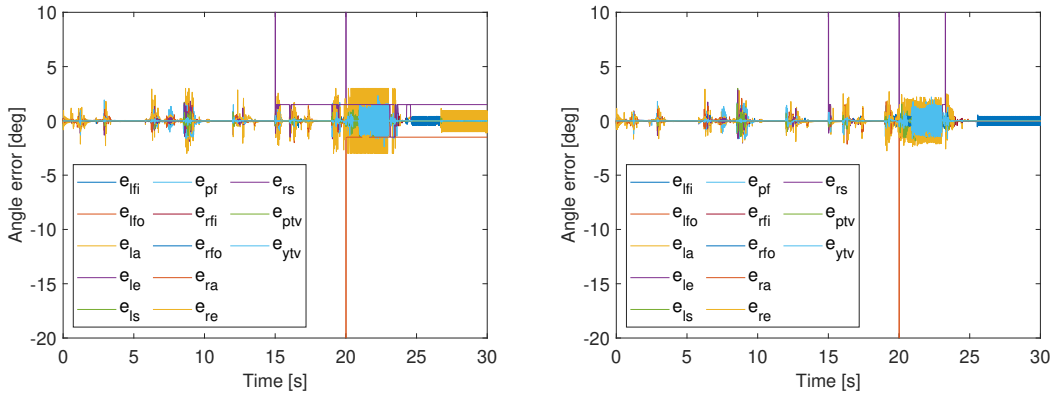
(a) D-INCA **(b) AD-INCA**

Fig. 23 Error in attained actuator deflection for maneuver A.



(a) D-INCA **(b) AD-INCA**

Fig. 24 Error in attained actuator deflection for maneuver B.



(a) D-INCA **(b) AD-INCA**

Fig. 25 Error in attained actuator deflection for maneuver C.

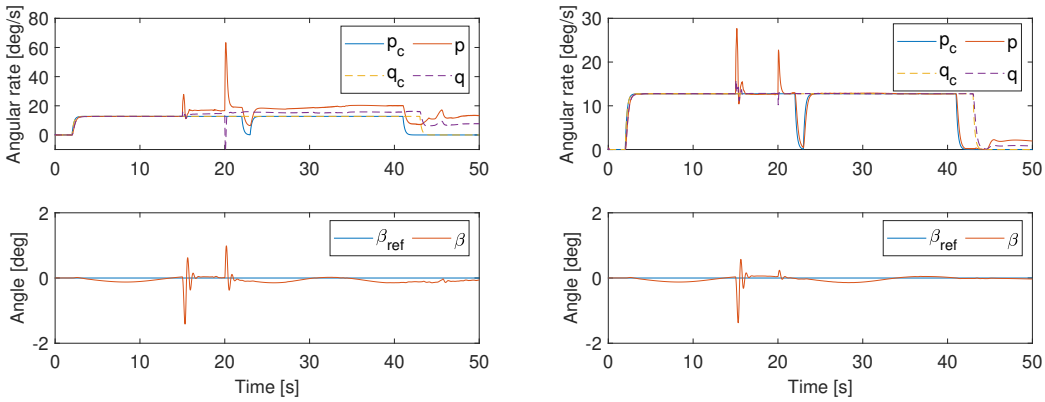
Table 8 in the Appendix gives the RMS tracking errors for the actuator failure sequence. What jumps out is that the INCA tracking errors are not necessarily worse than D-INCA's tracking errors regarding the broken actuators. Neither D-INCA nor INCA 'know' how to cope with the stuck actuators. Therefore, no significant difference between the two methods is expected. As for the heavily underdamped actuator, the assumed perfect actuator dynamics by INCA may actually be a reasonable first order representation of the underdamped dynamics. Whereas, with slightly underdamped dynamics, the D-INCA actuator dynamics model performs better than INCA's 'perfect actuator dynamics model'.

Table 8 also shows that for D-INCA the tracking errors are higher for the faulty than for the nominal actuators. The RLS estimator can reduce the tracking errors of the erroneous effectors slightly. However, the faulty actuators' tracking errors remain higher. Hence, the adaptive actuator dynamics model cannot describe the faulty actuator dynamics as good as it can describe the nominal actuator dynamics. This has two main reasons: 1) the RLS estimator needs time to update the parameters and 2) the first order discrete-time model is limited in representing the dominantly second order dynamics. Moreover, the error in δ_{le} tracking during maneuver C actually increases slightly with AD-INCA compared to D-INCA. From the time evolution of the errors in Fig. 25 it can be concluded that AD-INCA needs two large command inputs. Hence, it is likely that the applicable estimates did not converge to the true values after the failure event at $t = 20$ s. Despite its limitations, AD-INCA still presents reduced average tracking errors.

The control effort, included in Table 9 in the Appendix, increases significantly during the actuator failure events. Due to the fact that actuators are stuck in a non-zero position, other actuators need to compensate for these actuators with relatively large deflections as well. This also indicates the reduced solution space for the QP optimizer.

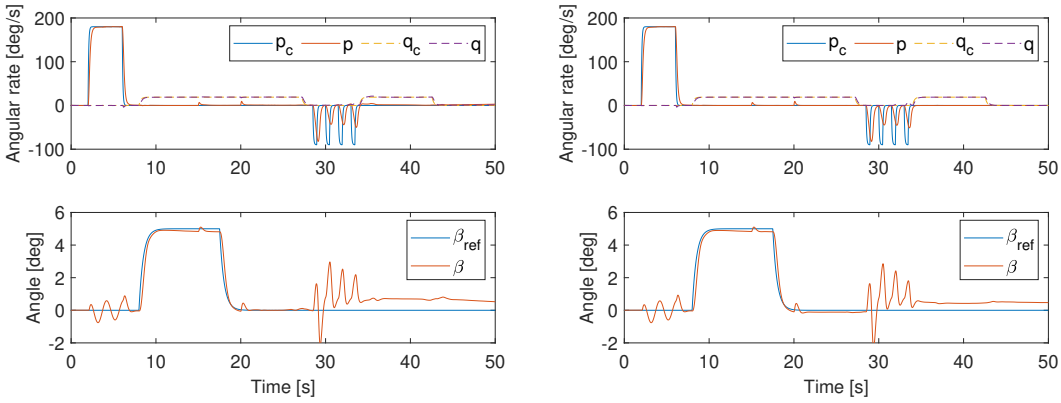
3. Outer loop tracking performance

Next, the results on tracking performance of the outer loop reference signals are shown in Figs. 26 to 28 and compared. In addition, the simulated translational trajectories are included in Fig. 32 in the Appendix.



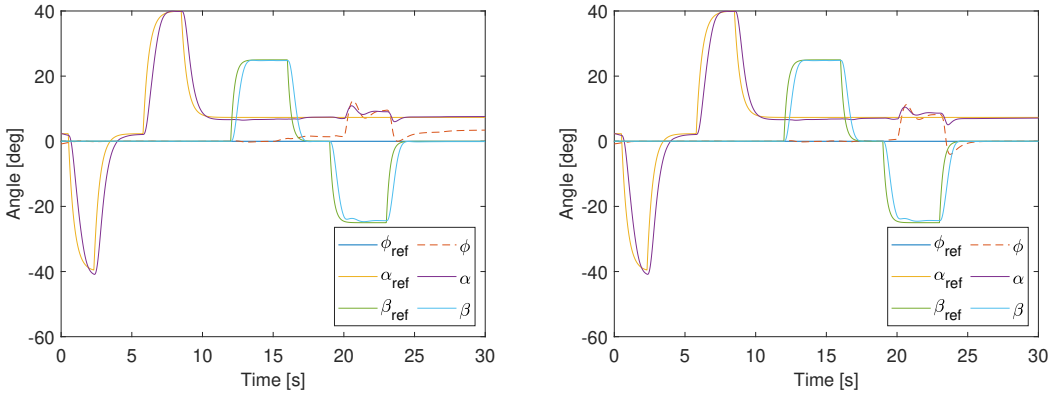
(a) D-INCA (b) AD-INCA

Fig. 26 Outer loop reference tracking maneuver A.



(a) D-INCA (b) AD-INCA

Fig. 27 Outer loop reference tracking maneuver B.



(a) D-INCA (b) AD-INCA

Fig. 28 Outer loop reference tracking maneuver C.

During maneuver A, the moment of actuator error initialization can be clearly distinguished in Fig. 26. Especially, at $t = 15$ s and $t = 20$ s when three actuators become stuck, errors peak. AD-INCA is able to recover itself after these errors. However, D-INCA cannot follow the roll and pitch rate references accurately afterwards. Figure 27 shows fairly similar tracking performance for both INCA and D-INCA, despite the steady state tracking errors found in Fig. 24. For maneuver C, the roll tracking performance deteriorates after actuator failures. AD-INCA is able to force the roll angle ϕ back to zero, whereas a steady state error in ϕ is introduced with D-INCA.

4. Parameter estimator performance

This section includes a short discussion on the parameter estimates evolution obtained for the adopted failure sequence. The parameters of the faulty actuators for maneuvers A, B and C are included in Figs. 29 to 31. The state scalar estimates $\hat{A}_i$ of the faulty actuators are pointed out, the input scalars $\hat{B}_i$ values are (almost) symmetrical to the $\hat{A}_i$ values around $\hat{\theta}_i = 0.5$. The three actuators that become stuck at 15 s and 20 s are clearly identified by the parameter estimates. During sufficient excitation, these parameters instantly converge to $\hat{A}_i = 1, \hat{B}_i = 0$. The drawback of the adopted least squares estimator is that for a single time step, two possible solutions exist. For example, during maneuver B and C the parameter estimate of the state scalar function dips to zero briefly after updating the erroneous actuator dynamics. However, after one sample time the estimator recovers to the correct parameter estimate.

The underdamped actuators are visualized too. In terms of a reduced first order lag model, the underdamped actuators become faster, and hence their input scalar B_i increases. Since, the left all-moving wingtip is quite heavily underdamped, the first order parameter estimates may eventually indicate perfect actuator dynamics. This can be seen in Fig. 30. However, convergence of these parameters is not instantly achieved, but only after sufficient excitation of the applicable actuators. Changing the adopted tuning parameters may exhibit different estimator behavior.

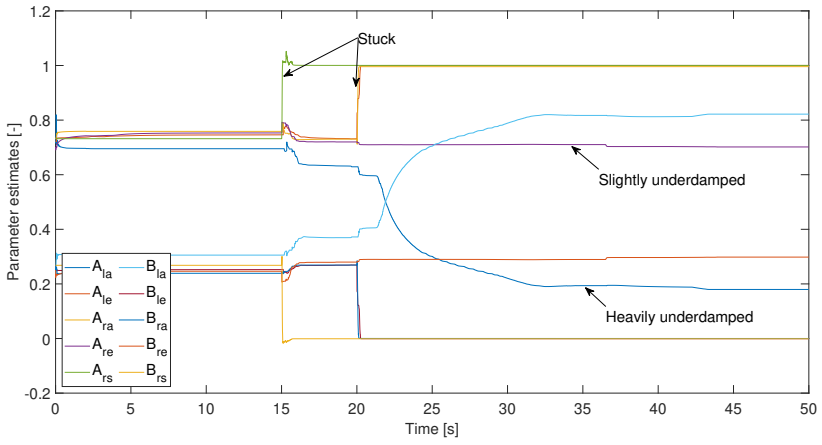


Fig. 29 Parameter estimates of faulty actuators for maneuver A.

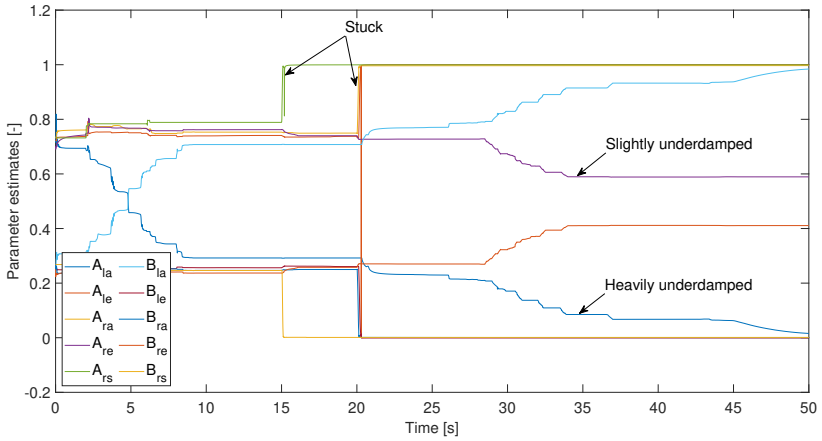


Fig. 30 Parameter estimates of faulty actuators for maneuver B.

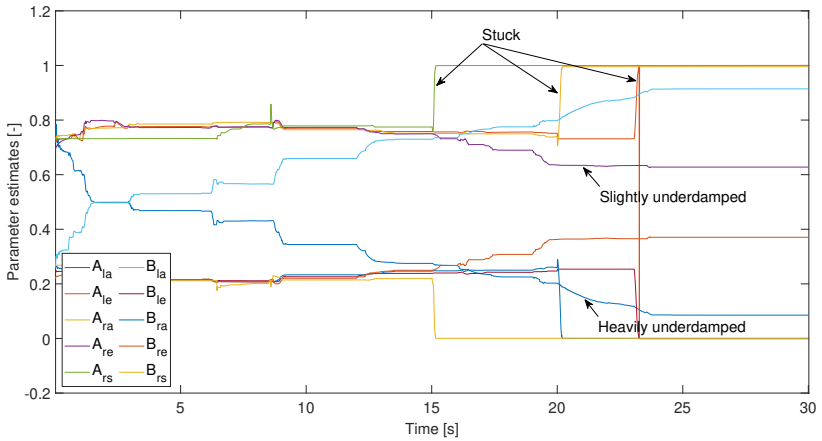


Fig. 31 Parameter estimates of faulty actuators for maneuver C.

VII. Conclusion and recommendations

The D-INCA framework compensates for actuator dynamics delays using an onboard model. It computes a desired command input to achieve the required physical deflections and pseudo-control inputs. Furthermore, D-INCA is much less conservative than INCA as it imposes physical actuator limits as output constraints on the expected actuator deflections rather than on the command inputs. Altogether, D-INCA increases closed loop stability under actuator dynamics delays. This relieves the closed loop controller from the need for a PCH as implemented for INCA. The PCH scales the outer loop command virtual control signal while these are actually feasible references with D-INCA.

For all the observed maneuvers, D-INCA is superior in terms of CA errors. Especially, for challenging reference trajectories. This also becomes clear through the tracking errors, which are significantly decreased compared to the

INCA results. However, the control actuator deflections are still not optimal at all times. Discontinuity in the CEJ model may cause actuators to be stuck in local optima between two discontinuous splines. Furthermore, second order actuator behavior becomes dominant during command input chatter introduced by discontinuities in the CEJ model. The first order onboard actuator dynamics model cannot accurately predict these second order dynamic phenomena, resulting in performance degradation. This makes it hard to predict the total potential of D-INCA without a continuous CEJ model. Besides, input chatter forms a barrier for practical implementation of the CA method. Command input chatter was already a problem for INCA, although this was never observed or discussed in detail in earlier work. These observations yield the need for a continuous CEJ. The implementation of such a model is recommended future research. Moreover, a first order actuator dynamics model was adopted to describe the second order actuator dynamics. This has proven to be effective for application within the ICE FCS. However, a higher order actuator dynamics model will likely increase performance. Lastly, it was shown that D-INCA embeds the same robustness against CEJ mismatch as the baseline INCA scheme.

The potential of D-INCA was further exploited through AD-INCA. Results show improved tracking and CA performance after various challenging actuator failure events. Pushing the ICE aircraft even further towards its potential in fault tolerance. Especially, steady state actuator errors are reduced which enhances outer loop reference tracking. However, the step in performance is minor compared to that from INCA to D-INCA. This is likely due to the fact that INCA and D-INCA already exhibit significant robustness. Due to the incremental approach, the mismatch between the optimal control deflection and the actual control deflection after actuator failure remains bounded by the actuator saturation limits and the sample time. The solution space does decrease, but this is also the case for AD-INCA. At the same time, this reduced solution space increases command input chatter.

The observed failure modes include stuck actuators as well as underdamped effectors. The case for zero torque was not investigated as it cannot be modelled through the actuator dynamics. Instead, such a case should be observed by the FCS and fed back to the CEJ rather than the actuator dynamics model. Hence, implementation of such fault tolerance requires a different approach. Robustness of the parameter estimator was achieved through an adaptive forgetting factor, preventing diverging parameter estimates even with little to no excitation. Unfortunately, a reduced-order-model needs to be determined in order for the second order parameter estimates to be useful within the D-INCA frame. The computational load from the Matlab balanced truncation model-order-reduction routine is high, because it needs to be called as an extrinsic function in Simulink. However, this is a drawback introduced by Matlab and not necessarily a practical problem. To overcome this issue, the algorithm may be modeled as a native S-function within Simulink as part of future research. Alternatively, a way to implement second order actuator dynamics directly in the QP optimizer could be further investigated.

Appendix

Table 4 RMS CA errors [lbf· ft] for nominal actuator behavior.

CA method	Maneuver	Control moment axis			Average	Total average
		L	*M*	*N*		
INCA	A	1.066E+03	2.146E+03	1.450E+03	1.554E+03	1.984E+07
	B	2.047E+05	8.503E+04	4.709E+07	1.579E+07	
	C	2.756E+07	7.582E+07	2.779E+07	4.372E+07	
D-INCA	A	2.439E+02	9.098E+02	1.378E+03	8.438E+02	4.231E+03
	B	1.571E+04	4.584E+03	3.639E+03	7.978E+03	
	C	2.370E+03	5.924E+03	3.317E+03	3.871E+03	
AD-INCA	A	2.536E+02	8.715E+02	1.371E+03	8.320E+02	4.018E+03
	B	1.509E+04	4.373E+03	3.338E+03	7.600E+03	
	C	2.279E+03	5.696E+03	2.889E+03	3.621E+03	

Table 5 RMS tracking errors [deg] for nominal actuator behavior.

CA method	Man.	Actuator												Avg.
		δ_{lfi}	δ_{lfo}	δ_{la}	δ_{le}	δ_{ls}	δ_{pf}	δ_{rfi}	δ_{rfo}	δ_{ra}	δ_{re}	δ_{rs}	δ_{ptv}	δ_{ytv}	
INCA	A	0.02	0.11	0.07	0.04	0.03	0.13	0.04	0.06	0.03	0.05	0.00	0.01	0.10	0.05
	B	0.24	0.25	0.78	0.80	0.78	0.88	0.24	0.25	0.82	0.87	0.63	0.56	0.77	0.61
	C	0.25	0.32	1.02	0.93	0.88	1.03	0.31	0.31	1.00	0.96	0.88	0.86	0.78	0.73
D-INCA	A	0.00	0.02	0.02	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.02	0.01
	B	0.02	0.02	0.06	0.06	0.06	0.12	0.02	0.03	0.07	0.06	0.05	0.06	0.04	0.05
	C	0.03	0.04	0.12	0.10	0.10	0.16	0.04	0.04	0.10	0.10	0.11	0.05	0.03	0.08
AD-INCA	A	0.00	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.02	0.01
	B	0.02	0.03	0.06	0.08	0.06	0.13	0.02	0.03	0.08	0.08	0.05	0.07	0.05	0.06
	C	0.08	0.06	0.14	0.14	0.13	0.20	0.05	0.05	0.11	0.13	0.10	0.08	0.06	0.10

Table 6 RMS control effort [deg] for nominal actuator behavior.

CA method	Man.	Actuator												Avg.
		δ_{lfi}	δ_{lfo}	δ_{la}	δ_{le}	δ_{ls}	δ_{pf}	δ_{rfi}	δ_{rfo}	δ_{ra}	δ_{re}	δ_{rs}	δ_{ptv}	δ_{ytv}	
INCA	A	0.03	2.08	0.95	0.91	0.05	10.5	0.18	0.34	0.32	1.45	0.01	0.36	1.10	1.40
	B	5.05	1.63	29.5	12.6	13.1	12.2	11.3	2.15	27.4	12.6	5.8	5.08	6.59	11.1
	C	2.37	13.5	30.8	18.2	20.9	15.2	4.78	8.8	32.8	15.7	18.2	8.63	8.34	15.3
D-INCA	A	0.01	2.05	1.03	0.87	0.01	10.5	0.09	0.15	0.32	1.45	0.00	0.36	1.15	1.38
	B	0.67	1.31	3.52	3.21	2.62	11.0	0.31	1.44	4.00	5.36	1.22	2.16	1.78	2.97
	C	0.86	2.80	14.0	7.92	1.56	9.02	1.79	2.95	14.6	8.65	1.50	0.82	1.79	5.25
AD-INCA	A	0.01	2.10	1.02	0.88	0.01	10.5	0.09	0.15	0.32	1.46	0.00	0.36	1.16	1.39
	B	0.71	1.30	3.53	3.31	2.65	10.9	0.35	1.44	4.02	5.42	1.16	2.12	1.72	2.97
	C	0.88	2.88	14.0	8.03	1.45	8.95	1.68	3.06	14.6	8.83	1.34	0.81	1.78	5.25

Table 7 RMS CA errors [lbf· ft] for actuator failure sequence.

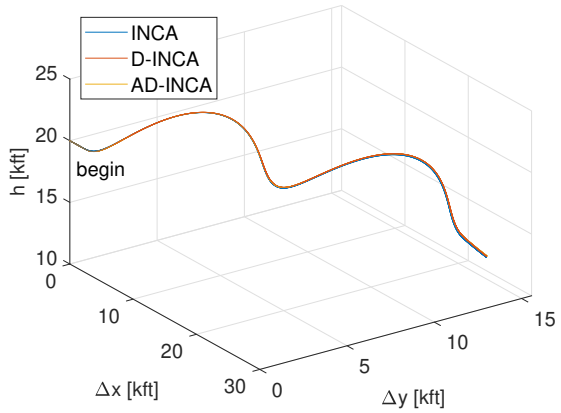
CA method	Maneuver	Control moment axis			Average	Total average
		L	*M*	*N*		
INCA	A	1.577E+09	2.165E+09	8.562E+09	4.101E+09	1.404E+09
	B	1.013E+08	1.623E+08	6.843E+07	1.107E+08	
	C	4.603E+05	7.558E+05	1.141E+06	7.856E+05	
D-INCA	A	3.402E+04	3.255E+04	1.980E+04	2.879E+04	2.173E+04
	B	5.094E+04	7.834E+03	6.035E+03	2.160E+04	
	C	2.709E+04	1.203E+04	5.270E+03	1.480E+04	
AD-INCA	A	8.595E+03	5.478E+03	1.410E+04	9.391E+03	1.434E+04
	B	5.015E+04	7.681E+03	4.765E+03	2.086E+04	
	C	2.689E+04	7.895E+03	3.554E+03	1.278E+04	

Table 8 RMS tracking errors [deg] for actuator failure sequence with stuck (red), heavily underdamped (orange) and slightly underdamped (yellow) actuators.

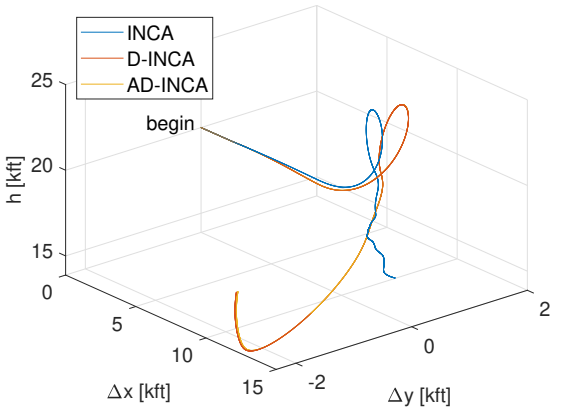
CA method	Man.	Actuator													Avg.
		δ_{lfi}	δ_{lfo}	δ_{la}	δ_{le}	δ_{ls}	δ_{pf}	δ_{rfi}	δ_{rfo}	δ_{ra}	δ_{re}	δ_{rs}	δ_{ptv}	δ_{ytv}	
INCA	A	0.19	0.21	0.52	1.04	0.71	0.67	0.21	0.21	0.68	0.70	1.10	0.68	0.56	0.58
	B	0.21	0.27	0.53	0.97	0.72	0.80	0.18	0.23	0.80	0.79	1.30	0.64	0.66	0.62
	C	0.27	0.32	0.80	1.16	0.91	1.08	0.30	0.32	1.12	0.88	1.25	0.84	0.78	0.77
D-INCA	A	0.08	0.14	0.80	1.19	0.05	0.11	0.10	0.15	1.16	0.16	1.32	0.12	0.11	0.42
	B	0.03	0.05	0.71	0.93	0.05	0.26	0.04	0.05	0.94	0.27	1.32	0.05	0.07	0.37
	C	0.06	0.10	0.70	1.13	0.29	0.33	0.05	0.14	0.97	0.38	1.19	0.14	0.24	0.44
AD-INCA	A	0.01	0.13	0.62	0.57	0.07	0.10	0.02	0.09	0.14	0.09	0.46	0.09	0.07	0.19
	B	0.04	0.05	0.47	0.64	0.08	0.26	0.03	0.09	0.13	0.27	0.44	0.07	0.08	0.20
	C	0.14	0.10	0.49	1.24	0.28	0.34	0.05	0.11	0.46	0.33	0.56	0.16	0.24	0.35

Table 9 RMS control effort [deg] for actuator failure sequence with stuck (red), heavily underdamped (orange) and slightly underdamped (yellow) actuators.

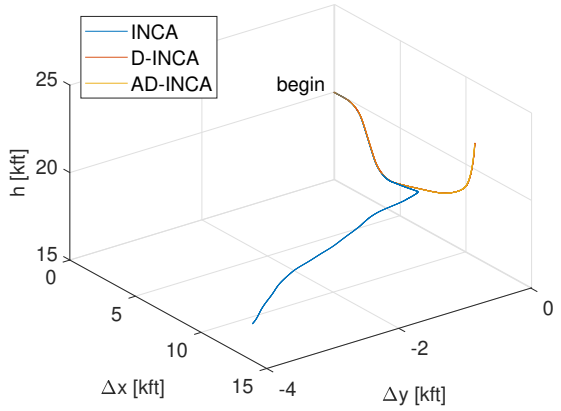
CA method	Man.	Actuator													Avg.
		δ_{lfi}	δ_{lfo}	δ_{la}	δ_{le}	δ_{ls}	δ_{pf}	δ_{rfi}	δ_{rfo}	δ_{ra}	δ_{re}	δ_{rs}	δ_{ptv}	δ_{ytv}	
INCA	A	1.81	5.38	7.11	11.4	12.7	15.5	4.67	3.87	2.85	9.81	17.6	4.38	7.22	8.02
	B	1.98	24.2	32.2	20.8	15.3	8.98	3.59	4.51	8.04	12.5	23.3	5.10	7.16	12.9
	C	2.99	7.13	25.9	21.0	27.7	12.4	7.93	9.33	14.3	16.9	22.8	7.86	8.44	14.2
D-INCA	A	28.9	12.0	10.4	23.7	13.7	18.7	2.86	14.5	2.54	20.5	25.1	8.68	11.5	14.9
	B	21.3	25.9	36.5	23.4	15.3	4.17	2.02	10.5	4.24	9.75	25.1	6.66	5.48	14.6
	C	22.3	20.5	26.5	19.9	19.6	20.5	3.53	23.1	13.7	19.4	21.2	5.32	4.63	16.9
AD-INCA	A	30.6	14.8	3.93	24.0	14.1	18.6	12.4	29.1	2.35	21.7	25.1	7.28	10.5	16.5
	B	21.8	25.3	36.2	23.4	15.5	4.36	1.87	10.4	4.30	13.8	25.1	7.38	5.60	15.0
	C	15.0	23.2	25.4	19.6	19.6	20.6	3.41	20.9	13.4	19.9	21.2	5.58	4.64	16.3



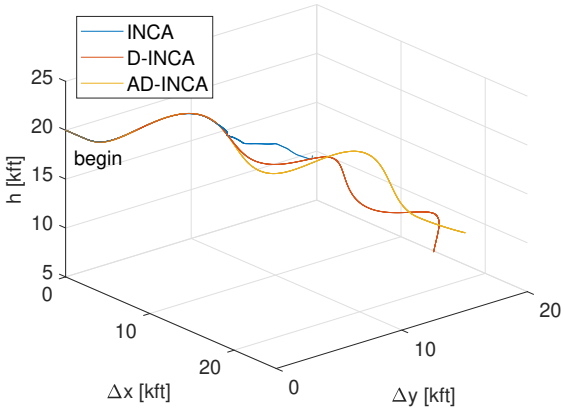
(a) Maneuver A with nominal actuators



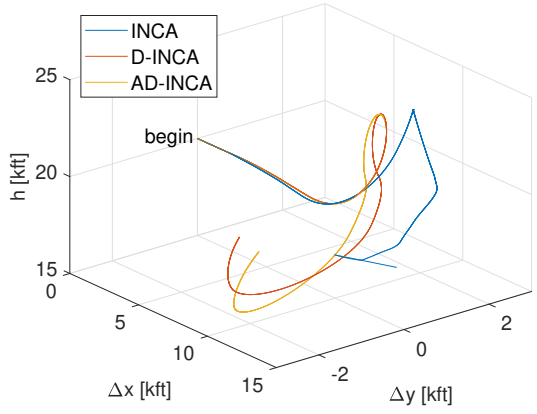
(b) Maneuver B with nominal actuators



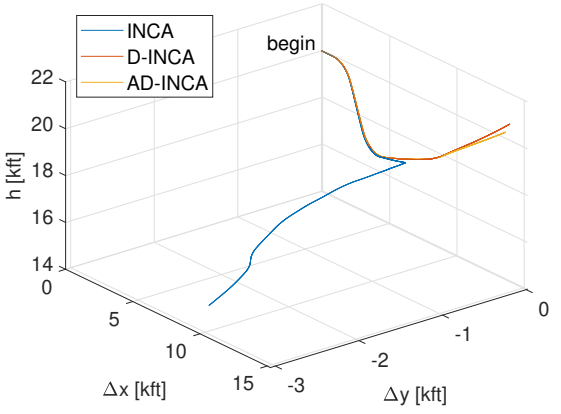
(c) Maneuver C with nominal actuators



(d) Maneuver A with actuator failures



(e) Maneuver B with actuator failures



(f) Maneuver C with actuator failures

Fig. 32 Translational trajectories.

References

[1] De Heer, P., "Incremental nonlinear control allocation for an aircraft with distributed electric propulsion," , 2021.

[2] Chatrath, K., Zheng, Y., and Shyrokau, B., "Vehicle Dynamics Control using model predictive control allocation combined with an adaptive parameter estimator," *SAE International Journal of Connected and Automated Vehicles*, Vol. 3, No. 2, 2020. https://doi.org/10.4271/12-03-02-0009.

[3] Johansen, T., "Optimizing Nonlinear Control Allocation," *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, 2004. https://doi.org/10.1109/cdc.2004.1429240.

[4] Tjønnås, J., and Johansen, T. A., "Optimizing Nonlinear Adaptive Control Allocation," *IFAC Proceedings Volumes*, Vol. 38, No. 1, 2005, p. 1160–1165. https://doi.org/10.3182/20050703-6-cz-1902.00850.

[5] Tjønnås, J., and Johansen, T. A., "Optimizing Adaptive Control Allocation with actuator dynamics," *Modeling, Identification and Control: A Norwegian Research Bulletin*, Vol. 29, No. 2, 2008, p. 69–76. https://doi.org/10.4173/mic.2008.2.4.

[6] De Vries, P. S., and Van Kampen, E., "Reinforcement learning-based control allocation for the innovative control effectors aircraft," *AIAA Scitech 2019 Forum*, 2019. https://doi.org/10.2514/6.2019-0144.

[7] Tol, H. J., de Visser, C. C., Van Kampen, E., and Chu, Q. P., "Nonlinear multivariate spline-based control allocation for high-performance aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, p. 1840–1862. https://doi.org/10.2514/1.g000065.

[8] Li, X., Jiang, Y., Zhang, J., Shi, S., and Zhao, L., "A method to compensate interaction between actuator dynamics and control allocator under Incremental Nonlinear Dynamic Inversion Controller," *IOP Conference Series: Materials Science and Engineering*, Vol. 428 012048, 2018. https://doi.org/10.1088/1757-899x/428/1/012048.

[9] Steffensen, R., Steinert, A., and Smeur, E. J., "Nonlinear dynamic inversion with actuator dynamics: An incremental control perspective," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 4, 2023, p. 709–717. https://doi.org/10.2514/1.g007079.

[10] Matamoros, I., "Nonlinear Control Allocation for a High-Performance Tailless Aircraft with Innovative Control Effectors An Incremental Robust Approach," , 2017.

[11] Baggi, R., Serrani, A., and Franco, E., "Hierarchical Dynamic Control Allocation for over-actuated aircraft: Methodology and flight tests on a scaled-down model," *2022 IEEE Conference on Control Technology and Applications (CCTA)*, 2022. https://doi.org/10.1109/ccta49430.2022.9966141.

[12] Acquatella, P., Falkena, W., Van Kampen, E., and Chu, Q. P., "Robust nonlinear spacecraft attitude control using incremental nonlinear dynamic inversion." *AIAA Guidance, Navigation, and Control Conference*, 2012. https://doi.org/10.2514/6.2012-4623.

[13] Härkegård, O., "Dynamic control allocation using constrained quadratic programming," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, p. 1028–1034. https://doi.org/10.2514/1.11607.

[14] Härkegård, O., "Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation," *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, 2002, p. 1295–1300. https://doi.org/10.1109/cdc.2002.1184694.

[15] Fortescue, T., Kershenbaum, L., and Ydstie, B., "Implementation of self-tuning regulators with variable forgetting factors," *Automatica*, Vol. 17, No. 6, 1981, p. 831–835. https://doi.org/10.1016/0005-1098(81)90070-4.

[16] Dorsett, K. M., and Mehl, D. R., "Innovative Control Effectors (ICE)," Tech. rep., 1996. URL https://apps.dtic.mil/sti/pdfs/ADB212813.pdf.

[17] Dorsett, K. M., Houlden, H. P., and Fears, S. P., "Innovative Control Effectors (ICE) Phase II," Tech. rep., 1997. URL https://apps.dtic.mil/sti/pdfs/ADB232172.pdf.

[18] Van der Peijl, I., "Physical Splines for Aerodynamic Modelling of Innovative Control Effectors," , 2017.

[19] Matamoros, I., and De Visser, C. C., "Incremental nonlinear control allocation for a tailless aircraft with innovative control effectors," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018. https://doi.org/10.2514/6.2018-1116.

Part II

Preliminary Analysis

*This part has been assessed for the course AE4020 Literature Study.

3

Literature Review

3.1. Introduction

Dynamical system designs rely on actuators that allow them to be effectively controlled to a desired state. Redundancy in control or over-actuation refers to the case that more independent actuators are available than that there are degrees of freedom to be controlled. Over-actuation of control systems provides several advantages, such as improved fault tolerance as well as increased tracking performance as each actuator is subject to its own constraints and control effectiveness. However, a redundant actuator suite also imposes high complexity on the control laws required to exploit these advantages to the full. For over-actuated systems, infinite control strategies can be adopted to follow a given reference signal[1]. To distribute control forces and moments to the available effectors, a control allocation (CA) method must be chosen. Moreover, over-actuation of systems often presents nonlinear and coupled control effectiveness. Accounting for these properties in CA has shown to increase the tracking performance of systems in previous research [2].

Various research papers have dived into the topic of solving the CA problem. The current literature study will investigate the CA solutions provided by previous work, building on earlier surveys [13, 14]. This paper considers the literature survey on CA methods by Matamoros [2, ch. 2] as a baseline for existing methods. To build upon this work, Figure 3.1 depicts a slightly updated breakdown of the earlier identified methods. The upper branch covers the static CA algorithms, including the CA methods that assume constant control effectiveness and therefore perfect actuator dynamics. This assumption is reasonable for many systems as actuator dynamics are often significantly faster than the system dynamics they are involved with. However, in the distribution of control moments for fast dynamic systems, this assumption might degrade the optimality in control allocation and ultimately closed loop stability.

Therefore, this study focuses on the incorporation of actuator dynamics in the search for optimal control distribution, in literature referred to as dynamic CA. This subset of CA methods is described in the lower branch of Figure 3.1. The current work will both include a brief summary on the dynamic CA methods as identified by [2] and include other methods based on more recent literature. Subsequently, a step will be made in the direction of fault tolerant control allocation, exploring parameter update laws for control effectiveness and actuator dynamics models. This topic is important to consider, because fault tolerance is one of the main advantages of over-actuated systems if properly exploited through the control laws.

[1] Assuming no actuator constraints are present.

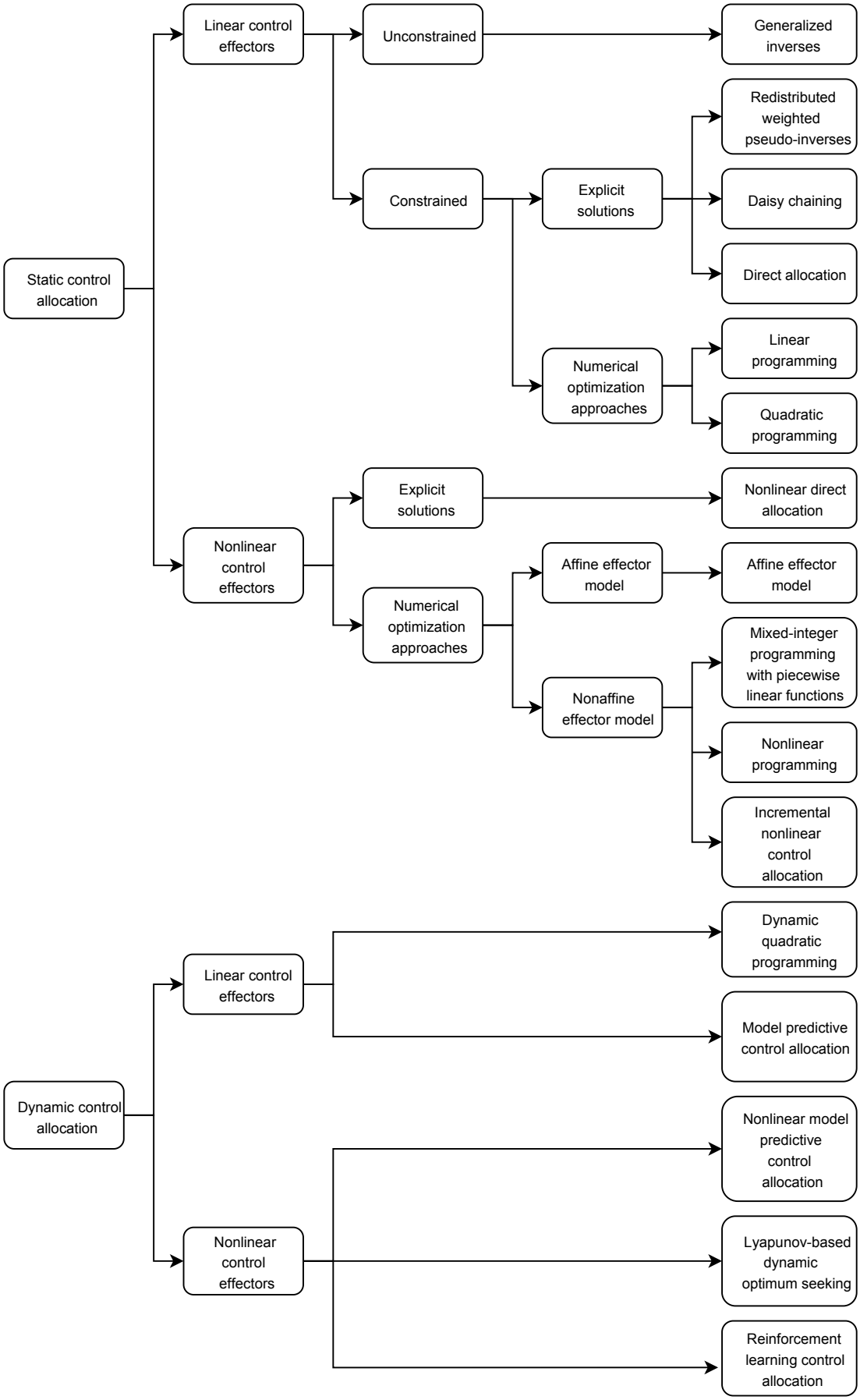


Figure 3.1: Updated breakdown of control allocation methods identified by Matamoros [2].

3.2. Dynamic control allocation methods

This chapter will focus on dynamic CA methods found in existing literature. The literature study aims to evaluate these methods in terms of their advantages, disadvantages and possible future research. A comprehensive view of the identified methods is included in Table 3.1.

3.2.1. Dynamic quadratic programming

In the quadratic programming (QP) problem, a weighted ℓ_2-norm cost function is minimized. This framework can be used in CA to drive the control deflection to a position that produces the required control moments and forces [13]. Additionally, a secondary objective is chosen such that the ill-posed problem can be solved numerically through various solvers. A popular choice for a secondary objective is to minimize the actuator deflection, keeping the actuators as far away from position limits as possible. This is beneficial for both performance and stability margins. Also, operating with all actuators at their limits is usually not very efficient in terms of power consumption. A drawback of QP is that it requires a linear control effectiveness function. Nevertheless, QP CA with an active set solver[2] has been effectively adopted for aerospace systems by [2, 3]. Härkegård [15] introduced dynamic QP (DQP), an extension of QP. Besides DQP taking into account the preferred actuator positions, it also includes previous actuator positions in order to penalize actuator rates. This is especially interesting when different actuators operate in different bandwidths. The slower components in the reference signal can be assigned to slow (but perhaps more effective) actuators, whereas fast components in the reference signal can be solved for with the faster effectors. A disadvantage of this method is that the relative importance between preferred actuator position and actuator rate should be tuned through weight matrices. Hence, the actuator bandwidth is only implicitly accounted for through these matrices. Moreover, the approach does not provide a reference for the actuator input that results in the optimal physical control deflection after actuator dynamics. Also, the example provided by Härkegård did not account for higher order actuator dynamics. This could be done through additional actuator position information, however causality may become issue. In a similar fashion, Yang and Hu [16] used dynamic weights to prevent reaction wheel (i.e., rotational rate or torque) saturation in a weighted pseudo-inverse framework. The method proved useful for energy saving compared to a static pseudo-inverse approach. The concept of actuator dynamics beyond saturation (i.e., rotational acceleration and higher order rotational time derivatives), however, was not considered in this work.

3.2.2. Incremental nonlinear control allocation

Matamoros [2] proposed incremental nonlinear control allocation (INCA), which is an extension of the incremental nonlinear dynamic inversion (INDI) method, for example found in [17, 18, 19]. Moreover, the INCA approach is a more generic version of the earlier multivariate simplex spline based CA method discussed in a paper by Tol et al. [9]. For the INCA approach, a model of both input dynamics and control effectiveness is required, however, no model of the system dynamics is needed. These dynamics are instead captured through output derivative feedback. This makes the method more robust and reconfigurable than classical feedback linearization. An additional advantage of posing the CA problem in control increments is that actuator rate constraints can be explicitly included in discretized position constraints. Also, the control effectiveness function can be linearized in the control increments. This makes it possible to solve the CA problem with linear solvers, such as general inverse, linear programming or the QP method discussed in Subsection 3.2.1, while still taking into account the nonlinear control effectiveness and coupling between actuators. The work of Matamoros [2] found that the incorporation of these nonlinearities and cross-coupling improved the performance compared to ordinary linear programming control allocation. Also, the control method was robust to control effectiveness model mismatch. The control approach even performed better for underestimating the control effectiveness, effectively increasing the control gains. Later work from De Heer [3] found that the implementation of actuator constraints in the INCA framework was rather conservative. Matamoros [2] adopted pseudo-control hedging (PCH) to avoid command windup problems, in combination with input constraints in the QP formulation. The resulting feasible input command however, is still subject to the actuator dynamics (i.e., the physical control deflections remain unequal to the optimal input command found by the control allocation algorithm). Therefore, the INCA framework does not exploit the system its full potential. Lastly, Matamoros [2] recommended fault tolerance through online system identification and control effectiveness update laws as future work since the current INCA framework does not consider actuator fault modes. This topic will be elaborated upon further in Section 3.3.

[2] Active set methods are iterative methods, where at each iteration they improve their guess of the optimal active set [13].

3.2.3. Model predictive control allocation

In order to account for the conservative implementation of actuator constraints in INCA, De Heer [3] adopted a separate model predictive controller (MPC), or receding horizon approach, to drive the actuator deflections to the control allocation input command. Hereby, effectively canceling the actuator dynamics. The results showed that the actuators follow the input commands and therefore that the performance of the INCA control strategy increased. The advantage of using MPC for this goal is that this approach will yield the optimal control strategy, as opposed to for example linear feedback control. Also, MPC reduces the order of lead compensation required compared to simply inverting the actuator dynamics in a pre-filter (See Subsection 3.2.4). Moreover, Chatrath et al. [4] adopted model predictive control allocation (MPCA) for the CA problem as a whole, from pseudo-control input (i.e., control force/moment) to input command, omitting the quadratic programming approach used in [2] and [3]. It was shown that the incorporation of the actuator dynamics in the MPCA framework ensures stable motion for a vehicle operating at its friction limits, whereas a QP approach neglecting actuator dynamics is not able to do so. Thus, increasing the performance of the system. Another example of MPCA is described by Naderi et al. [20]. The method entails a combination of pseudo-inverse control allocation with MPCA to effectively determine and enforce the feasible region of the pseudo-control input as integral part of the CA algorithm. In order to reduce computational load of the discussed approach, an approximate feasible region was used by the MPC rather than the exact feasible region. Computational load remains the main downside for using MPCA, making it hard to implement on fast complex real-world systems with limited processing power. In addition, the MPC controller requires information on the actuator rates and filtering the applicable signals introduces delays. MPC might not be able to mitigate such delays and/or increased computation times. This way it could break down to the point of non-optimal CA or eventually closed loop instability. Besides, these constraints make it hard to certify for, for example, implementation in flight control systems. Also, the proposed work [3, 4] assumed the actuator dynamics to be known and linear. Hence, this approach fails to capture nonlinearities and uncertainties in the actuator dynamics. An analogous nonlinear adaptive MPCA would be a logical next step, however, this step will increase the computational load of the controller even more.

3.2.4. Feedforward actuator compensation

Similar methods to MPC [3] have been found to drive the physical actuator deflection to the control allocation reference, for example [21, 10]. These works show that the actuator dynamics can be effectively canceled for some cases through feedforward compensation. Although this approach is limited in the lead it can produce, higher order actuator dynamics can be partly accounted for by canceling the steady state gain and neglecting the higher frequency lag dynamics in the compensator [10]. This method could be sufficient for certain applications. However, by adopting a mere gain correction the system might break down for higher frequency components. The CA approach will not exploit the system's full potential for the entire bandwidth. Also, the approach relies on the assumption that the actuator dynamics are fully known. Note that even with canceling the actuator dynamics, the actuators are still physically constrained in position and rate and this needs to be accounted for in the control allocation algorithm.

3.2.5. Nonlinear dynamic inversion with actuator dynamics

Steffensen et al. [11] found a nonlinear control method that incorporates actuator dynamics in the feedback linearization loop, actuator nonlinear dynamic inversion (ANDI). By inverting the actuator dynamics a linear relationship is obtained between output and input command rather than between output and physical input. The result is an incremental nonlinear control approach that relies on knowledge of the output time derivatives. This knowledge might not always be obtained easily and filtering of noisy derivative signals would introduce delays reducing the system performance. Also, the adopted method only takes into account first order actuator dynamics, or bandwidth, which should be known for the control law design. In case of higher order actuator dynamics, an equivalent bandwidth could be considered that approaches the higher order dynamics. However, this will affect the performance of the control strategy. Otherwise, higher order output derivatives may help in the incorporation of higher order actuator dynamics, imposing significant requirements on the sensor and filtering capabilities. It was shown that the proposed control method does not necessarily outperform the existing INDI control approach [11]. Instead, its performance needs to be evaluated on a case to case basis. However, for systems with slower actuator dynamics, fast system dynamics, or systems that require control allocation to distribute the control forces and moments, accounting for actuator bandwidth could make the control strategy more optimal. Furthermore, in their

control design Steffensen et al. [11] did not consider any control allocation problems, but rather assumed a system for which the degrees of freedom equals the number of independent actuators[3]. Hence, future research would be required to apply the proposed method to over-actuated systems.

3.2.6. Lyapunov-based control allocation

Another approach to tackle the CA problem, Lyapunov-based control allocation (LBCA), pioneered by Johansen [5], consists of employing control Lyapunov-functions (CLF) to derive dynamic actuator reference update-laws for the CA problem. The CLF was augmented with a barrier function to enforce input constraints. The method uses Lagrangian multipliers for optimum seeking. Furthermore, the CLF approach guarantees closed loop stability under the assumption that a higher level controller Lyapunov function exists. Since no iterative solver is required in the proposed control-Lyapunov framework, its numerical efficiency is superior to earlier discussed numerical optimization based approaches, such as QP and MPCA. However, this also means that optimality of the control strategy is only obtained asymptotically rather than for each individual input command. This can lead to some level of performance degradation as shown through a case study by Tavasoli and Naraghi [22]. Another disadvantage of the method includes possible convergence problems in the case of non-convex cost function and constraints [5]. Later on, De Castro and Brembeck [23] aimed to combine the best of both worlds to tackle the problem of infeasible reference signals and its implications on closed-loop stability. The approach uses Lyapunov-based constraints, control barrier functions and numerical optimization. Preserving the numerical optimization formulation, the Lyapunov-based constraints allowed to improve the response to infeasible virtual inputs, while the control barrier functions were used to force the system to a safe state space. A numerical optimizer was used to decrease the effect of infeasible pseudo-control inputs on the overall closed loop stability through evaluating the Lyapunov function. Also, the barrier functions allow for effective use of the null-space to return to a preferred actuator space. However, the problem of actuator dynamics was neglected by De Castro and Brembeck [23].

3.2.7. Reinforcement learning control allocation

Recent progress in the field of reinforcement learning has led to an alternative view on the control allocation problem [8, 24]. More specifically, De Vries and Van Kampen [8] adopted a Q-learning method to solve the CA problem. This approach relieves the need for a system model. Although reinforcement learning control allocation (RLCA) methods are promising, their certification and implementation for safety critical systems remains problematic. To reduce complexity of the training process, a reduced model (longitudinal aircraft motion) was used with four control actuators. This makes the method less useful for high dimensional nonlinear and coupled systems. In addition, the Q-learning approach with action and state discretization resulted in aggressive flight behavior far less practical than that of model-based methods. It was argued that the problem of 'similar cost' and therefore the cycling between control strategies would be a problem for other types of RLCA as well. A possible solution for these drawbacks includes the use of continuous function approximators. Another topic for future research is the use of hybrid CA, combining a neural network representation of the control effectiveness with model independent outer loops. Also, penalizing actuator positions and rates is something to consider within the proposed framework.

[3] I.e., the control effectiveness matrix is square and full rank.

Table 3.1: Identified methods for implementing actuator dynamics in control allocation.

Method	References	Advantages	Downsides	Possible future research topics
DQP	[15]	✓Allows to allocate the control tasks based on actuator bandwidth.	✗Implicit modeling of actuator bandwidth through weights. ✗Assumes linear control effectiveness.	- Implement in CA increments.
RLCA	[8, 24]	✓No model required.	✗Poor tracking performance compared to model-based CA. ✗Problematic for high dimensional nonlinear coupled systems. ✗Hard to certify for safety critical systems.	- Explore hybrid solutions. - Include continuous function approximator.
INCA	[2, 3, 9]	✓Allows a linear solver for non-control affine problems. ✓Allows for discretized rate constraints. ✓Robust to control effectiveness mismatch.	✗No actuator dynamics included. ✗No adaptive control allocation.	- Include actuator dynamics. - Combination with online parameter estimation.
MPCA	[3, 4]	✓Optimal solution. ✓Explicit modeling of actuator dynamics. ✓Reduces the required derivative order of output feedback. ✓Parameter update law is implemented.	✗Computationally expensive. ✗Prone to delays in feedback signals. ✗Current work assumes linear control effectiveness. ✗Hard to certify for safety critical systems.	- Nonlinear MPCA (even higher computational load).
FFAC	[21, 10]	✓Simple implementation.	✗Higher order actuator dynamics require non-causal filter, or ✗Feedforward of steady state gain reduces performance.	- Use to increase actuator bandwidth in combination with other methods.
ANDI	[11]	✓Explicit implementation in the existing NDI controller.	✗Only first order dynamics are taken into account. ✗No control allocation included.	- Feasibility for higher order actuator dynamics. - Extent to CA problems.
LBCA	[5, 23, 6, 7]	✓Computationally efficient. ✓Allows for parameter update law. ✓Can be used in combination with numerical optimizers for infeasible pseudo-control inputs.	✗Only asymptotic optimality. ✗Convergence issues for non-convex (often occurs with nonlinearity) problems.	

3.3. Fault tolerant control allocation

This chapter elaborates on fault tolerant control allocation (FTCA) methods found in existing literature. This topic often touches upon the methods described in the previous chapter. Moreover, FTCA in this work is with regard to actuator failures and does not consider FTC in terms of unpredicted system configurations (e.g., wing surface reduction on aircraft). This topic is usually accounted for through a higher level controller and has been thoroughly studied (e.g., Smit et al. [25] and Smeur et al. [26]).

3.3.1. Lyapunov-based methods

Tjønnås and Johansen build on the initial work [5] to include fault tolerance [6] in the Lyapunov CA scheme. Unknown parameters were defined as part of the control effectiveness function and in later work also for the actuator dynamics model [7]. Subsequently, update laws were adopted for the input command reference and the unknown parameters. Asymptotic stability and convergence of the parameters was obtained with the proposed method. However, the example used in the paper considered a slow system (ship at low speed), focusing on disturbance rejection. This raises the question as to what extent the proposed control allocation method, which is only asymptotically optimal, is applicable for faster dynamical systems.

3.3.2. Fault detection and isolation

Cristofaro et al. [27] used fault detection, isolation and control reconfiguration to tackle the FTCA problem. The key point of the method is the use of a family of unknown input observers that are designed to decouple faults affecting selected actuators or clusters of actuators. These observers use linear algebraic rules and nonlinearities are thus treated as disturbances. This may limit the applicability for highly nonlinear systems. Once the faults are detected and isolated, the control reconfiguration module is activated. The actuators that are identified as faulty are not used and the desired control effect is produced by the joint action of the healthy devices only. Moreover, no parameter update laws are required, as the entire control strategy is changed. However, this also leaves less room for adapting the control law in other ways that are less drastic. Another downside of the method is that it does not consider actuator dynamics as for example [7] does.

3.3.3. Adaptive MPCA

To address the assumption of perfectly known actuator dynamics in the MPCA framework, Chatrath et al. [4] adopted online system identification to update the control effectiveness function. Furthermore, the proposed method uses auxiliary model-based recursive least-squares to estimate the actuator dynamics parameters. The designed update law takes into account unknown dynamics, as well as actuator fault modes. Through simulating the controller with errors in the initial parameter estimates and a steering action after 10 seconds, it was shown that the parameter estimates converge quickly. A drawback of the proposed method is that it assumes the actuator dynamics to be linear and the order and number of zeros to be known. Implementation of nonlinear actuator dynamics requires to reconsider the adopted parameter update laws. Additionally, the method requires the system to be constantly excited by the input for the adaptive update laws to converge.

3.3.4. Other methods

Another example of FTCA was investigated by Baggi et al. [12]. The aim of this work was to take advantage of the secondary control surfaces of airliners (i.e., flaps, slats and spoilers) to add redundancy in aircraft attitude control. The fault tolerance was obtained through a control effectiveness parameter update law rather than defining the uncertainty in terms of both the actuator dynamics and control effectiveness together. By putting preference on the primary control surfaces, the secondary effectors were only used if necessary to obtain the desired control moments. For airliners this is a sensible approach. However, by enforcing such preference on 'primary' actuators, the actuator suite in its entirety might be under-exploited. This means the proposed method is less suitable for implementation in high performance over-actuated systems, for example fighter aircraft. Also, when looking at the physical reality of faulty actuators, it makes more sense to model this concept in terms of the actuator dynamics rather than the control effectiveness function. Lastly, Tohidi et al. [28] used a control method wherein a virtual dynamics model with virtual reference compensates for unknowns in the control effectiveness. This relieves the need for uncertainty estimation and online system identification in the control allocation layer. Furthermore, the control design

relies on measurements of the pseudo-control input to realize the desired virtual dynamics. An advantage of this approach is that the controller does not discard actuator surfaces with reduced control effectiveness like [27] does. The optimal solution, however, is obtained only asymptotically as is the case for the control-Lyapunov approach.

3.4. Conclusion

The choice to neglect actuator dynamics in control allocation techniques can be argued for systems with relatively fast actuators compared to the system dynamics. However, for faster, high performance systems it is found that neglecting the actuator dynamics reduces performance of the control allocation strategy. Ways to explicitly account for the actuator dynamics have been found for MPCA [3, 4] and Lyapunov CA [5, 6, 7]. Unfortunately, these methods each suffer from their own limitations. That is, computational load and non-optimal control strategies. Moreover, RLCA relieves the need for a physical control effectiveness model. However, it was found that RLCA lacks in tracking performance, especially when considering highly coupled nonlinear control effectiveness functions for high dimensional systems. This 'curse of dimensionality' remains problematic for the application of RLCA in complex over-actuated systems.

On the other hand, incremental methods [2, 10, 11] heavily rely on output derivative measurements rather than a system dynamics model. Methods have been proposed to account for actuator dynamics in incremental control allocation [3, 10] in separate control loops, however, no previous research has combined the dynamic incremental control allocation problem in a single control loop. Also, the described methods [3, 10] are subject to the limitations stated in Table 3.1. Hence, the combination of INDI with dynamic actuator control allocation techniques remains under-addressed in previous research. Although the INCA framework provides a lot of robustness in control effectiveness model mismatch, over-actuated systems offer far more potential in terms of actuator fault tolerant control. This potential has not yet been reached through earlier control designs. Moreover, incremental control allocation as is has proved useful in dealing with nonlinear non-control affine systems in a robust manner. Hence, taking into account both the inherent advantages as well as the future growth potential in dynamic incremental CA, it is recommended to deepen out this field of research.

In addition, previous research has elaborated on adaptive control allocation for several other CA methods. For example, parameter update laws have been defined for both the actuator dynamics model [7] as well as the control effectiveness model [7, 4, 12]. Other methods rely on fault detection, isolation and control reconfiguration [27]. These approaches can give insight in possible ways to implement FTCA within the incremental nonlinear control allocation framework.

Given the identified research gaps, the resulting research question to be explored is: *How can actuator dynamics and actuator failure modes be effectively accounted for within the incremental nonlinear control allocation framework?* In order to answer this research question, several subquestions can be posed. These include:

- How can the control allocation problem be solved in the actuator nonlinear dynamic inversion control problem?
- How can higher order actuator dynamics be taken into account under output derivative measurement constraints?
- To what extent can the implementation of actuator dynamics in the incremental nonlinear control allocation framework improve the tracking performance?
- What update laws are required to account for actuator failures under the incremental nonlinear control allocation framework?
- To what extent can these update laws increase tracking performance under actuator failure modes?

Part III

Additional Results

4 Miscellaneous topics and results

4.1. INCA and PCH

To omit command saturation problems, Matamoros [2] applied a pseudo-control hedge (PCH) within the INCA-based FCS. The PCH is defined by the difference in virtual control input $\boldsymbol{\nu}$ or $\dot{\boldsymbol{\omega}}_{c,k-1}$ and the estimated obtained virtual control $\hat{\boldsymbol{\nu}}$ or $\hat{\dot{\boldsymbol{\omega}}}_k$. Assuming that no unoptimalities originate from the QP solver, the PCH only depends on actuator dynamics and saturation limits. For INCA the PCH $\boldsymbol{\nu}_h$ can be computed as

$$\boldsymbol{\nu}_h = \dot{\boldsymbol{\omega}}_{c,k-1} - \hat{\dot{\boldsymbol{\omega}}}_k = J^{-1}\nabla_{\boldsymbol{\delta}}\boldsymbol{\Phi}(\mathbf{x}_{k-1}, \boldsymbol{\delta}_{k-1})(\mathbf{u}_{k-1} - \boldsymbol{\delta}_k). \tag{4.1}$$

Next, the hedge is fed back into a reference model as a compensation signal. The reference model $\boldsymbol{\nu}_{\text{rm}}$ is given by

$$\boldsymbol{\nu}_{\text{rm}} = K_{\text{rm}}(\boldsymbol{\omega_c} - \boldsymbol{\omega}_{\text{rm}}) \tag{4.2}$$

with K_{rm} a diagonal matrix. The hedged reference signal fed as a control input into the attitude control system is the state vector of the reference model

$$\boldsymbol{\omega}_{\text{rm}} = \frac{1}{\text{s}}\left(\boldsymbol{\nu}_{\text{rm}} - \boldsymbol{\nu}_h\right). \tag{4.3}$$

In Fig. 4.1 the block diagram of the INCA method with PCH is depicted.

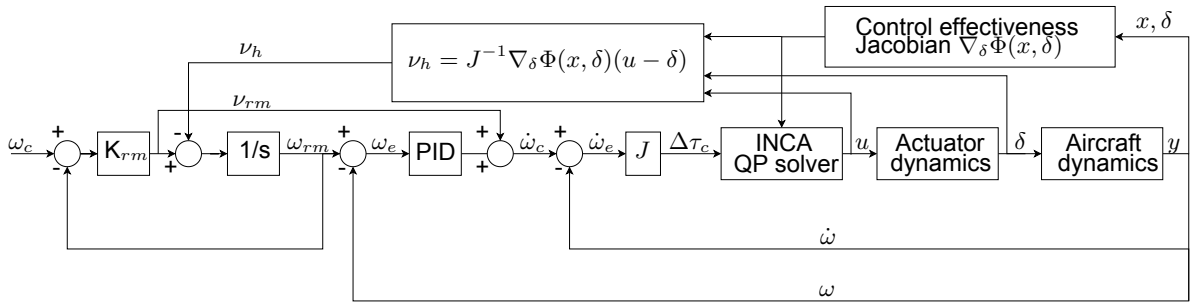


Figure 4.1: INCA control structure with PCH.

The PCH was introduced to act in the case of actuator saturation by avoiding unfeasible virtual command signals. However, the underlying problem with INCA is not primarily actuator saturation limits. These are hard-coded in the QP solver and cannot be violated by the constrained optimization problem. Instead, the QP solver cannot catch up with the virtual control input references as the optimal command inputs are not achieved due to actuator dynamics delays. The outer loop signal is hedged or scaled to preserve closed loop stability with these system delays present.

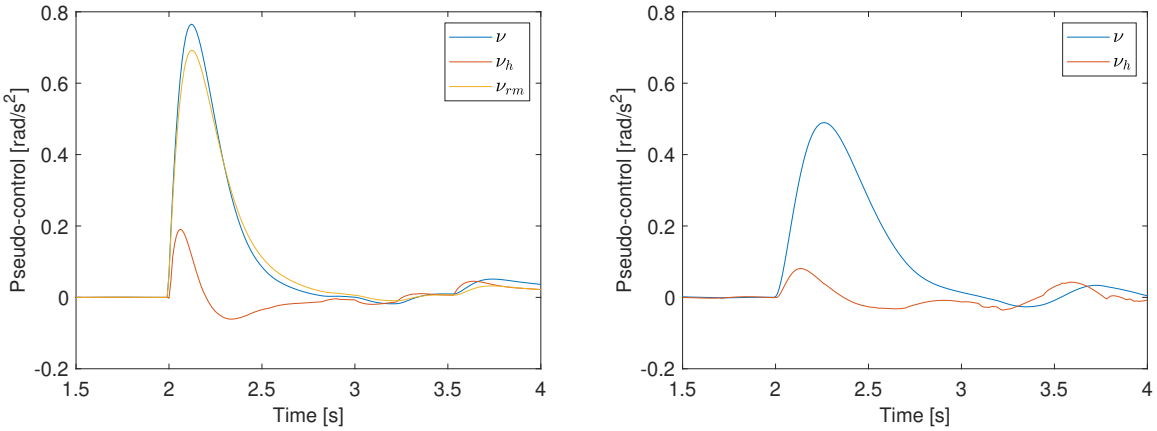
Matamoros [2] did not compute the PCH based on $\mathbf{u}_{k-1}$ and $\boldsymbol{\delta}_k$, but instead compared $\mathbf{u}_k$ with $\boldsymbol{\delta}_k$. However, these signals merely provide a difference between optimal control solution and the current physical actuator deflection (i.e., control increment). Actuator dynamics and saturation limits are not explicitly involved in this quantity. Nevertheless, the signal is still positively correlated with actuator

dynamics delays and saturation events, as larger control increments in general take longer to be achieved. Therefore, the PCH implementation still proved effective.

For this thesis, the INCA PCH was adapted in order to represent the actuator dynamics instead of the control increment needed to achieve optimal actuator deflection. In reality however, the difference in results was found negligible. Hence, if tuned correctly, the control increment turns out to be a good quantity for hedging the virtual control input. However, even the updated PCH implementation with a slightly better theoretical basis, is clearly lacking. The actuator dynamics are always present, also when the reference signal is not that challenging. Therefore, the virtual command is being hedged throughout the entire flight with no true system limitations present. Lastly, the QP constraints on actuator saturation are very conservative if imposed onto the command inputs instead of the actuator positions themselves. To conclude, the PCH method results in performance degradation for INCA while the source of the problem is left untouched: the system delays introduced by the actuator dynamics.

4.1.1. PCH for all-moving wing tips

To illustrate the effects of PCH for INCA, a brief use case with maneuver A will be investigated. A 180 deg/s roll maneuver is initiated at t=2 s.



(a) INCA: PCH with feed-forward

(b) INCA: PCH without feed-forward

Figure 4.2: Roll pseudo-control hedging for maneuver A.

Figure 4.2 depicts the time evolution of the virtual control inputs and pseudo-control hedge. A distinction is made between the PCH method without and with feed-forward of ν_{rm}, as depicted in Fig. 4.1. This feed-forward was introduced by [29] to increase tracking performance. It 'boosts' the signal when large changes in reference states occur, effectively counteracting the hedge ν_h. A large difference between the two virtual control inputs ν can be observed.

Figure 4.3 gives a clear comparison in virtual control inputs between INCA without PCH, with PCH (with and without feed-forward) and D-INCA (without PHC). The PCH with no feed-forward clearly hedges the virtual control input which becomes slightly less steep than found for the outer methods. However, the INCA PCH feed-forward term actually increases the magnitude of the virtual control input, making this controller the most aggressive one. It can be concluded that in the current PCH implementation, not the hedge, but the reference model's feed-forward is dominant. The references for both methods without PCH (INCA and D-INCA) portray a similar slope. The difference in magnitude can be assigned to the tracking performance. For higher tracking errors in ω, the virtual control $\nu = \dot{\omega}_c$ will be higher as well. The virtual control input of the PCH without feed-forward its magnitude does not increase compared to the D-INCA method, because the error is kept artificially low by the reference state ω_{rm} which tends to move towards the estimated obtained angular rate. For INCA with PCH, the error is computed relative to this reference state, whereas for D-INCA the error is computed with respect to the outer loop command input ω_c. The difference between these error definitions is supported by Fig. 4.4.

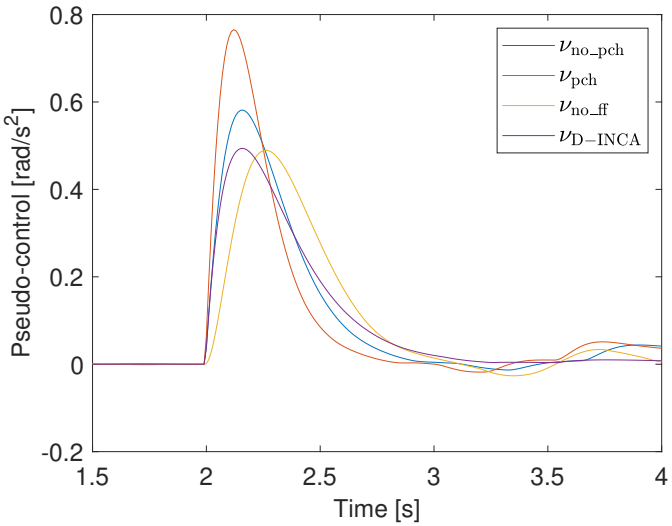
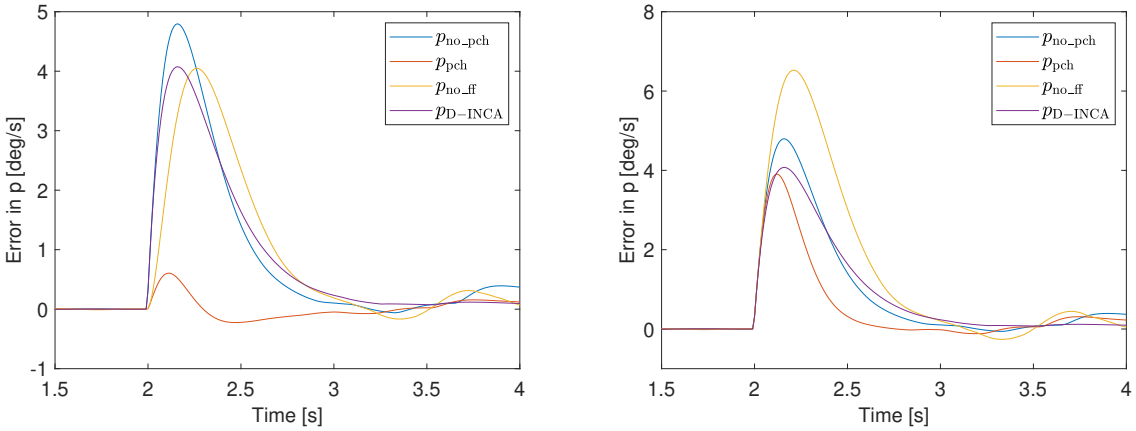


Figure 4.3: Roll virtual control input for maneuver A.



(a) $\omega - \omega_{\text{rm}}$. **(b)** $\omega - \omega_c$.

Figure 4.4: Roll rate tracking errors for maneuver A.

As the tracking performance increases thanks to the feed-forward term, the error with the reference model diminishes almost entirely as depicted in Fig. 4.4a. With no PCH for INCA, $\omega_{\text{rm}} = \omega_c$ holds and INCA cannot hide the error in angular rate tracking behind a reference model. This method yields the greatest errors.

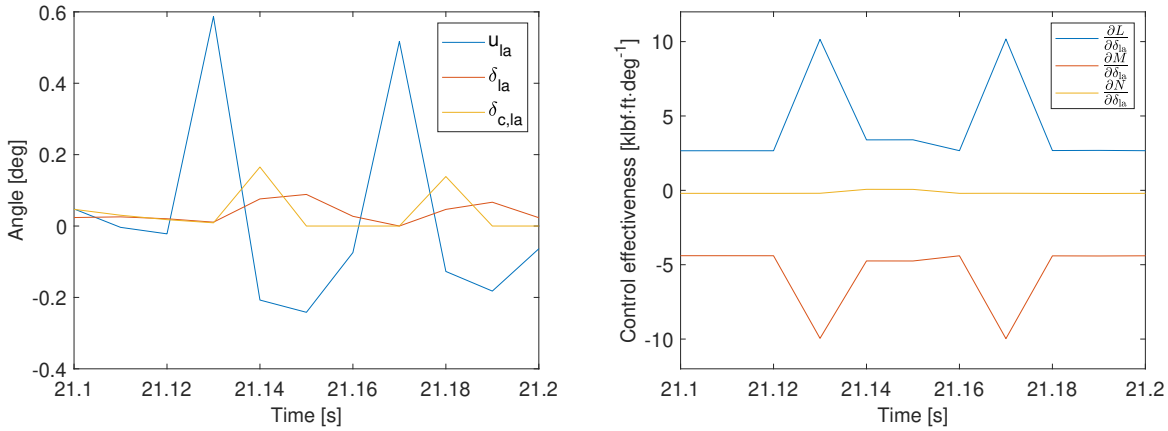
When considering the actual error in attained angular rate, compared to the commanded value, as depicted in Fig. 4.4b, we conclude the PCH with no feed-forward performs the worst. This method also produces the least aggressive inputs and has no inner loop actuator compensation to back this up. Next in line is INCA with no PCH. Thereafter comes D-INCA, which peaks slightly higher than INCA with PCH. However, D-INCA is the first to reach a steady error of zero after the intense maneuver. This is where the knowledge embedded in the onboard actuator dynamics model clearly pays off.

To conclude this topic, the PCH implementation adopted by Matamoros [2] increases the virtual control input under actuator dynamics delays through actuator output feedback and reference model feed-forward. The latter effectively counteracts the hedge itself. The D-INCA method uses an onboard model of the actuator dynamics to mitigate the actuator dynamics induced time delays at command input level instead. This makes it hard to compare the two methods at CA error (i.e., command torque error) level. INCA+PCH with reference model feed-forward compensates before the torque command, whereas D-INCA does

so after the torque signal. Nevertheless, by explicitly modeling actuator dynamics knowledge in the CA optimizer, the closed loop performance is increased as oscillations in attitude rate tracking are diminished. A D-INCA controller with a more accurate actuator dynamics model will likely reduce the angular rate tracking errors even further.

4.2. Control effectiveness modeling and command input chatter

A great portion of actuator tracking errors in can be pinpointed to command input chatter. This chatter is induced by discontinuity in the adopted CEJ model. An example of this phenomenon is depicted in Fig. 4.5 for D-INCA. However, similar events were observed with INCA.



(a) Command input and physical input

(b) CEJ

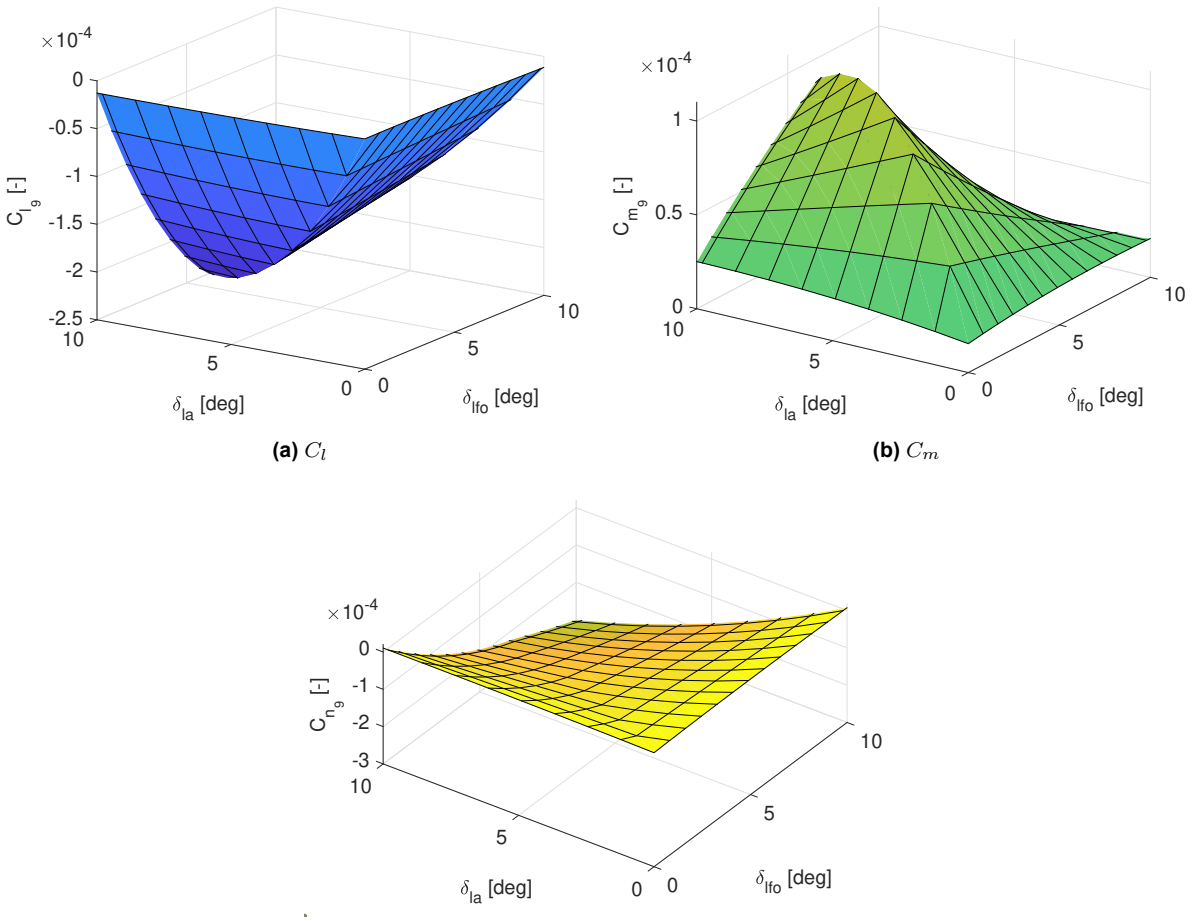
Figure 4.5: Chatter in D-INCA LAMT control solution for maneuver A.

For maneuver A at $t = 21.13$ s, command input chatter occurs at the left all-moving wingtip. The command input follows a jump in the CEJ function, as can be seen in Fig. 4.5. Moreover, the aerodynamic control effectiveness model in Fig. 4.6 portrays a ridge of 0-th order continuity for $C_{l_9} = \text{fcn}(\alpha, \delta_{\text{lfo}}, \delta_{\text{la}})$ and $C_{m_9} = \text{fcn}(\alpha, \delta_{\text{lfo}}, \delta_{\text{la}})$ around the applicable operating point in Table 4.1. The same model for yaw effectiveness is much smoother. From the aerodynamic model, it becomes clear that the partial derivatives with respect to δ_{la} will be discontinuous around the applicable operating point. Also, the aerodynamic model explains why $\frac{\partial L}{\partial \delta_{\text{la}}}$ increases while $\frac{\partial M}{\partial \delta_{\text{la}}}$ decreases at the same time as δ_{la} deflection becomes smaller.

Table 4.1: Chatter example operating point.

t [s]	12.12	12.13
α [deg]	10.9	10.9
δ_{lfo} [deg]	0.0845	0.0534
δ_{la} [deg]	0.0204	0.0108

Particularly, for extremely challenging reference trajectories the chatter problem becomes more prominent as the QP solution space is reduced. This chatter affects the control solution as the first order actuator dynamics model cannot accurately predict the increased second order actuator behavior. This can also be observed in Fig. 4.5a as a mismatch arises between δ_{la} and $\delta_{c,\text{la}}$. Additionally, discontinuity in the CEJ model increases the chance of finding local optima. This makes it hard to predict the total potential of D-INCA without a continuous CEJ model. Last but not least, input chatter poses a problem for practical implementation of INCA due to structural limitations of the actuators. It would be worthwhile to investigate whether a continuous CEJ model would prevent the chatter and increase closed loop performance.



(a) C_l **(b)** C_m

(c) C_n

Figure 4.6: Onboard aerodynamic model sliced at $\alpha = 10.9°$.

4.3. Outer loop angle tracking

In order to assess the performance of INCA, body angular rate commands $\boldsymbol{\omega}_c$ were fed forward as outer loop references. Nevertheless, it is also interesting to observe angle tracking characteristics. In particular, to achieve coordinated turns, sideslip β references are required. Therefore, a sideslip inversion loop was designed by [2] to generate ω_c for maneuvers A, B and C. With this outer loop, β is the reference and the yaw rate r the control input. The other two angular rates, p and q can be controlled directly (maneuver A and B) or rely on their own aerodynamic inversion loops as implemented for maneuver C. The outer loop sideslip inversion control diagram is depicted in Fig. 4.7. The kinematic inversion operation is derived in [2, p. 62] as

$$r_c = \underbrace{\left(\frac{-u}{\sqrt{u^2+w^2}}\right)}_{a_\beta(x)}^{-1}\left[\dot{\beta}_c - \underbrace{\frac{1}{\sqrt{u^2+w^2}}\left(F_x + F_y + F_z + wp_c\right)}_{b_\beta(x)}\right] = a_\beta(x)^{-1}\left[\dot{\beta}_c - b_\beta(x)\right] \tag{4.4}$$

with u and w the longitudinal and vertical body frame velocities. F_x, F_y and F_z indicate the axial, lateral and normal aerodynamic forces within the body frame.

In order to obtain a $\dot{\beta}_c$ for a given sideslip error, a PID is required as depicted in Fig. 4.7. The gains adopted in this PID determine how aggressive the $\dot{\beta}_c$ and thus r_c signals will be. The PID values in [2] for INCA were chosen such that optimal performance was obtained while maintaining closed loop stability.

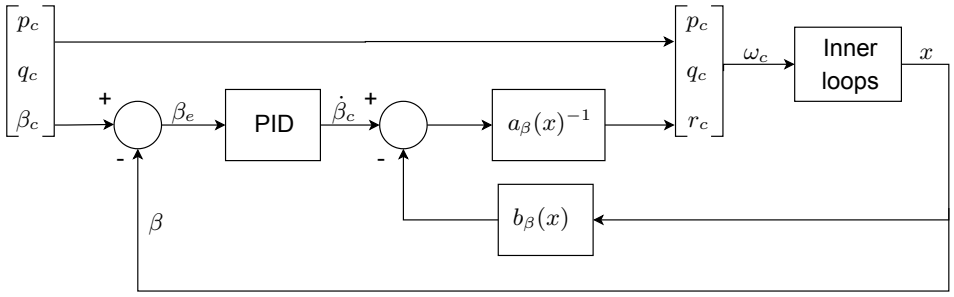


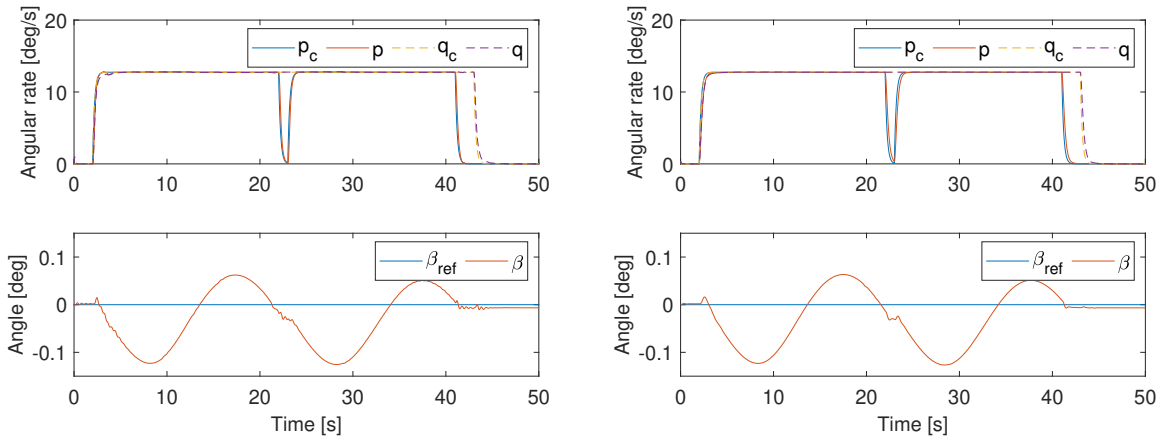
Figure 4.7: Sideslip inversion outer loop.

This section will elaborate on outer loop PID tuning for coordinated turns with INCA and D-INCA. The use case of maneuver A will be used again in order to display the differences in PID tuning requirements between CA methods. The adopted outer loop gains are included in Table 4.2.

Table 4.2: Outer loop PID β gains for maneuver A.

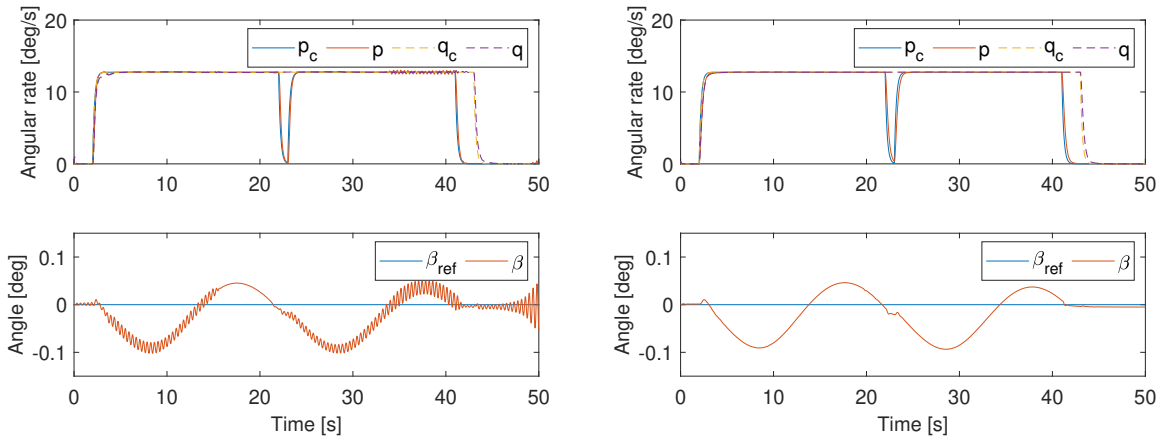
β controller	nominal	high	extreme
$K_{P\beta}$	15	20	25
$K_{I\beta}$	0	0	2
$K_{D\beta}$	1	5	10

Figures 4.8 to 4.10 show the sideslip reference tracking with INCA and D-INCA for three different sets of PID gains. It can be seen that tracking for the first two sets of gains is very similar. Adopting a more stressing PID to achieve higher performance, INCA is showing unstable behavior in Fig. 4.8. Even further increasing the PID gains for the zero sideslip reference, the INCA solution becomes singular. Whereas, D-INCA's tracking performance with respect to β keeps increasing slightly.



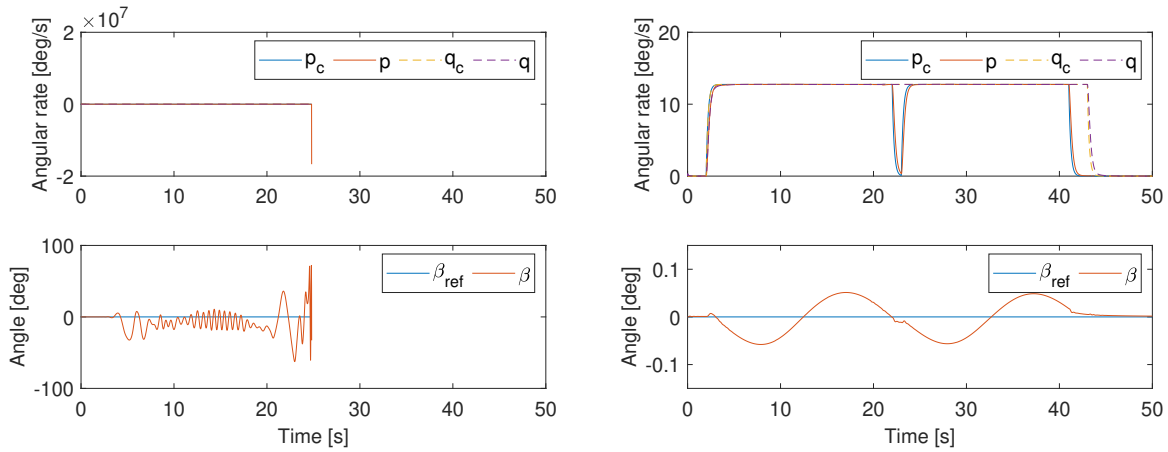
(a) INCA **(b)** D-INCA

Figure 4.8: Outer loop reference tracking maneuver A with nominal β PID gains [15, 0, 1].



(a) INCA **(b)** D-INCA

Figure 4.9: Outer loop reference tracking maneuver A with β PID gains [20, 0, 5].



(a) INCA **(b)** D-INCA

Figure 4.10: Outer loop reference tracking maneuver A with β PID gains [25, 2, 10].

The main advantage of D-INCA is that the actuator model-based approach seems to be more effective in mitigating actuator dynamics induced time delays. Therefore, it presents more stability margin and can handle more aggressive outer loop references ω_c. When performing coordinated turns, this means it can track the zero sideslip reference more accurately. Presumably, the conservative actuator constraints in INCA, which are imposed on $\mathbf{u}$ rather than $\boldsymbol{\delta}$, play a big role in this. For the current thesis, the same outer loop PID gains were assumed for INCA, D-INCA and AD-INCA. However, outer loop angle reference tracking increases if we consider more challenging PID gains for D-INCA. Whereas, if we encounter instability with INCA, it may be beneficial to reduce the outer loop references for angle tracking. To conclude, any outer loop angle tracking controller should be tuned based on the inner loop system's performance. Ultimately, for a fair judgement of β tracking capability with maneuver A, Fig. 4.8a should be compared with Fig. 4.10b.

4.4. Higher order dynamic INCA

(A)D-INCA is limited as the onboard actuator dynamics model has to be of first order. For some applications with higher order actuator dynamics, a first order representation may be sufficient. However, a higher order actuator dynamics model will likely be more accurate and increase closed loop performance. Therefore, this section will briefly elaborate on second order actuator dynamics modeling. For the implementation of higher order actuator dynamics within the D-INCA control scheme, the example of second order actuator

dynamics will be used. Let's first define the second order actuator dynamics model as

$$\begin{bmatrix} \mathbf{z}_{1_{k+1}} \\ \mathbf{z}_{2_{k+1}} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{z}_{1_k} \\ \mathbf{z}_{2_k} \end{bmatrix} + \mathbf{B}\mathbf{u}_k. \tag{4.5}$$

Where the state vector contains the discrete time actuator positions and rates

$$\begin{bmatrix} \mathbf{z}_{1_k} \\ \mathbf{z}_{2_k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\delta}_k \\ \frac{\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k}{\Delta t} \end{bmatrix}. \tag{4.6}$$

Substituting the states in Eq. (4.6) into the second order discrete state-space model in Eq. (4.5), the following model is obtained

$$\begin{bmatrix} \boldsymbol{\delta}_{k+1} \\ \frac{\boldsymbol{\delta}_{k+2} - \boldsymbol{\delta}_{k+1}}{\Delta t} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ * & * \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_k \\ \frac{\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k}{\Delta t} \end{bmatrix} + \begin{bmatrix} 0 \\ * \end{bmatrix} \mathbf{u}_k \tag{4.7}$$

with the asterisk indicating the entries in state and input equation that depend on the second order dynamics. Moreover, in line with the incremental D-INCA approach, the state space model can be rewritten as follows

$$\begin{bmatrix} \boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k \\ \frac{\boldsymbol{\delta}_{k+2} - 2\boldsymbol{\delta}_{k+1} + \boldsymbol{\delta}_k}{\Delta t} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ * & * \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_k \\ \frac{\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k}{\Delta t} \end{bmatrix} + \begin{bmatrix} 0 \\ * \end{bmatrix} \mathbf{u}_k - \begin{bmatrix} \boldsymbol{\delta}_k \\ \frac{\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k}{\Delta t} \end{bmatrix}. \tag{4.8}$$

The left-hand side now corresponds to the discrete-time increments in actuator position and actuator rate. The analogous continuous time increments are described through

$$\Delta\boldsymbol{\delta} = \boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k \quad \text{and} \quad \Delta\dot{\boldsymbol{\delta}} \approx \frac{\boldsymbol{\delta}_{k+2} - 2\boldsymbol{\delta}_{k+1} + \boldsymbol{\delta}_k}{\Delta t}. \tag{4.9}$$

The two left-hand side increments in Eq. (4.9) need to correspond with some information in the control effectiveness model, i.e., a model is required of the form

$$\dot{\boldsymbol{\tau}} = \Phi_2(\boldsymbol{\delta}, \mathbf{x}, \dot{\boldsymbol{\delta}}, \dot{\mathbf{x}}), \tag{4.10}$$

including an outer loop reference for $\dot{\boldsymbol{\tau}}$. Subsequently, the directional derivative with respect to $\boldsymbol{\delta}$ and $\dot{\boldsymbol{\delta}}$ yields a higher order control effectiveness Jacobian

$$\Delta\dot{\boldsymbol{\tau}} = \nabla_{\boldsymbol{\delta},\dot{\boldsymbol{\delta}}} \Phi_2(\boldsymbol{\delta}, \mathbf{x}, \dot{\boldsymbol{\delta}}, \dot{\mathbf{x}}) \begin{bmatrix} \Delta\boldsymbol{\delta} \\ \Delta\dot{\boldsymbol{\delta}} \end{bmatrix}. \tag{4.11}$$

Firstly, the right-hand side of Eq. (4.11) needs to be solved for the outer loop reference $\Delta\dot{\boldsymbol{\tau}}$. Next, the obtained solution for $\Delta\boldsymbol{\delta}$ and $\Delta\dot{\boldsymbol{\delta}}$ could be used to find the values for $\boldsymbol{\delta}_{k+1}$ and $\boldsymbol{\delta}_{k+2}$ through Eq. (4.9). Lastly, $\mathbf{u}_k$ should be chosen such as to adhere to this solution through the system dynamics described by Eq. (4.8).

To conclude, a higher order actuator model for D-INCA would impose significant requirements on the control effectiveness model and the outer loop reference. Another challenge includes the required knowledge on actuator rates and dealing with potentially noisy signals. In addition, the actuator dynamics input equation $\mathbf{B}$ becomes a vector, making inversion impossible. The fact that both a $\boldsymbol{\delta}_{k+2}$ and $\boldsymbol{\delta}_{k+1}$ need to be chosen at time step k may also lead to problems as these values may not match for subsequent time steps. Freezing the solution for $\boldsymbol{\delta}_{k+2}$ for the next time step could be considered in order to solve this problem. Alternatively, a piecewise model structure may be explored to better capture the second order dynamics. Furthermore, the implementation of a second order actuator dynamics model for D-INCA is deemed out of scope of the current thesis.

Part IV

Closure

5

Conclusion

This chapter aims to answer the research questions posed in Chapter 1. These questions are repeated below for convenience, starting with the main research question:

Research Question

How can actuator dynamics and online actuator failure modes in non-control affine nonlinear over-actuated systems be effectively accounted for within the incremental nonlinear control allocation framework?

The sub-questions were:

1. How can actuator dynamics be taken into account within the INCA control loop under output derivative measurement constraints and possible onboard model limitations? And
2. to what extent can the compensation of actuator dynamics in the incremental nonlinear control allocation framework improve the tracking performance?
3. What update laws are required to account for actuator failures under the incremental nonlinear control allocation framework? And
4. to what extent can these update laws increase tracking performance under actuator failure modes?

5.1. Sub-question 1

The first sub-question can be answered as follows. Through an actuator dynamics model within the CA solver we can optimize for the command input instead of the physical input. By adopting an incremental approach, a first order actuator dynamics model can be used with no need for higher order derivative output feedback than already required for INDI or INCA. Higher order actuator dynamics can be partly accounted for through a first order lag model as presented within the ICE FCS. A higher order actuator dynamics model may increase tracking performance. However, it also imposes higher requirements on the control effectiveness model. The implementation of such a higher order actuator dynamics D-INCA is recommended future research.

Moreover, INCA was implemented in combination with a discontinuous CEJ model. This yields problems for INCA as well as (A)D-INCA. First and foremost, discontinuity in the CEJ model results in undesired command input chatter, during which second order actuator behavior becomes dominant. The first order onboard actuator dynamics model cannot accurately predict these second order phenomena, resulting in performance degradation. Also, input chatter is a problem for any practical implementation of INCA due to structural limitations of the actuators. Furthermore, CEJ discontinuity may cause actuators to be stuck in local optima between two discontinuous splines (potentially even with different signs). This makes it hard to predict the total potential of D-INCA without a continuous CEJ. These findings leave the need for a higher order, i.e. first order or higher, continuity aerodynamic control effectiveness function. The implementation of such a model is highly recommended for future research.

5.2. Sub-question 2

An earlier adopted method to compensate for actuator dynamics delays includes a PCH with reference model feed-forward. The latter increases the virtual control input under actuator dynamics delays through actuator position feedback. D-INCA on the other hand, compensates at command input level through an onboard actuator dynamics model. It also presents a better definition of the constrained QP problem. Namely, INCA imposes the actuator saturation limits as QP constraints on the command inputs instead of the physical inputs. Whereas, D-INCA constrains the command inputs based on the model-based estimated physical actuator deflections. To conclude, it was found that D-INCA compensates for the actuator dynamics in a more effective way than INCA in combination with reference model feed-forward. By accounting for actuator dynamics, D-INCA mitigates inner loop time delays. Therefore, stability margins are increased. This advantage allows for more challenging outer loop references and outer loop gains. In terms of flight control, this results in better trajectory tracking performance throughout the flight envelope. In addition, it can be concluded that the robustness against CEJ mismatch is not decreased with respect to the baseline INCA scheme.

5.3. Sub-question 3

Thirdly, a recursive least squares estimator with adaptive forgetting factor can be used to estimate the actuator dynamics online. Moreover, the incremental scheme allows for adaptation of the actuator dynamics model for each control increment. Robustness of the parameter estimator was achieved through an adaptive forgetting factor, preventing diverging parameter estimates even with little to no excitation. Unfortunately, the RLS estimator is not able to identify a first order model from a second order system. Since the D-INCA framework only allows for first order actuator dynamics compensation, a model-order-reduction step is needed for the RLS estimator to be implemented within the AD-INCA frame. For this thesis, a Matlab built-in balanced truncation routine was adopted to obtain a first order actuator dynamics model. The computational load from the Matlab routine is high, because it needs to be called as an extrinsic function in Simulink. However, this is a drawback introduced by Matlab and not necessarily a practical problem. To overcome the issue, the algorithm may be modeled as a native S-function within Simulink as part of future research. Alternatively, a way to implement second order actuator dynamics directly in the QP optimizer could be further investigated. Unfortunately, this step would impose high requirements on control effectiveness modeling.

5.4. Sub-question 4

AD-INCA allows for stuck actuators as well as underdamped actuators to be identified. Hereby nullifying steady state tracking errors and increasing performance after actuator failure events. However, the step in performance is minor compared to the performance leap from INCA to D-INCA. This is likely due to the fact that INCA and D-INCA already exhibit great robustness. Due to the incremental approach, the mismatch between the optimal control deflection and the actual control deflection after actuator failure remains bounded by the actuator saturation limits and the sample time. Actuator failure events also introduce increased command input chatter. This is likely due to a reduced solution space, making it more likely for the optimizer to stay in local optima and to hop between discontinuous splines.

A failure event resulting in zero torque was not investigated as it cannot be modeled through the actuator dynamics. Instead, such a case should be observed by the FCS and fed back to the control effectiveness model rather than the actuator dynamics model. Hence, implementation of such fault tolerance requires a different approach which may be part of future research.

6

Recommendations

This final chapter provides a brief overview of the primary recommendations for the future continuation of this research.

Rec 1: Continuous control effectiveness Jacobian
The lack of a continuous CEJ model results in chatter in the command input solution for INCA, D-INCA and AD-INCA. Discontinuity in the spline model may increase the chances of the solver getting stuck in local optima. In addition, second order actuator dynamics phenomena become more dominant during chatter. Hence, for D-INCA and AD-INCA it will be harder to capture the actuator dynamics with a first order discrete-time model. Lastly, command input chatter introduces problems for real-life implementation of the CA methods, due to structural limitations of the effectors. Therefore, the implementation of a continuous CEJ model is recommended for future research.

Rec 2: Higher order D-INCA
It was observed that the first order discrete-time actuator dynamics model is limited in presenting the second order actuator dynamics. During nominal actuator behavior, the model may overestimate the system response, resulting in smaller command inputs than required for the optimal physical inputs. Additionally, overshoot phenomena as observed with (faulty) underdamped actuators cannot be captured by the model. Solving the problem presents significant requirements on the control effectiveness function as well as the outer loop reference. Nevertheless, the implementation of a higher order D-INCA will increase the performance of the CA method even further. This makes it a good candidate for future research.

Rec 3: Native S-function balanced truncation routine
The RLS estimator in AD-INCA cannot directly estimate a first order model from the second order actuator dynamics. Therefore, a second order difference equation model structure was adopted. On the other hand, (A)D-INCA only accept first order actuator dynamics models to compensate for the actuator dynamics delays. This mismatch yields the need for a model-order-reduction from the second order model estimate to a set of first order model parameter estimates. Currently, the reduced-order-model is computed with a built-in Matlab balanced truncation routine. However, this approach is computationally costly as the algorithm needs to be called as an extrinsic function from Simulink. This practical problem could be solved by implementing the balanced truncation algorithm as a native S-function in Simulink instead.

Rec 4: Zero torque actuator fault tolerance
The actuator failure events in this research include underdamped and stuck effectors, as these situations can be represented by an actuator dynamics model. Another fault mode that was not considered is that of zero torque. This failure case should be observed by the FCS and fed back to the CEJ. The implementation of an adaptive CEJ for zero torque fault tolerance is recommended for future research.

References

[1] K. M. Dorsett et al. *Innovative Control Effectors (ICE)*. Tech. rep. 1996. URL: https://apps.dtic.mil/sti/pdfs/ADB212813.pdf.

[2] I. Matamoros. "Nonlinear Control Allocation for a High-Performance Tailless Aircraft with Innovative Control Effectors An Incremental Robust Approach". 2017.

[3] P. de Heer. "Incremental nonlinear control allocation for an aircraft with distributed electric propulsion". 2021.

[4] K. Chatrath et al. "Vehicle Dynamics Control using model predictive control allocation combined with an adaptive parameter estimator". In: *SAE International Journal of Connected and Automated Vehicles* 3.2 (2020). DOI: 10.4271/12-03-02-0009.

[5] T. A. Johansen. "Optimizing Nonlinear Control Allocation". In: *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)* (2004). DOI: 10.1109/cdc.2004.1429240.

[6] J. Tjønnås et al. "Optimizing Nonlinear Adaptive Control Allocation". In: *IFAC Proceedings Volumes* 38.1 (2005), pp. 1160–1165. DOI: 10.3182/20050703-6-cz-1902.00850.

[7] J. Tjønnås et al. "Optimizing Adaptive Control Allocation with actuator dynamics". In: *Modeling, Identification and Control: A Norwegian Research Bulletin* 29.2 (2008), pp. 69–76. DOI: 10.4173/mic.2008.2.4.

[8] P. S. de Vries et al. "Reinforcement learning-based control allocation for the innovative control effectors aircraft". In: *AIAA Scitech 2019 Forum* (2019). DOI: 10.2514/6.2019-0144.

[9] H. J. Tol et al. "Nonlinear multivariate spline-based control allocation for high-performance aircraft". In: *Journal of Guidance, Control, and Dynamics* 37.6 (2014), pp. 1840–1862. DOI: 10.2514/1.g000065.

[10] X. Li et al. "A method to compensate interaction between actuator dynamics and control allocator under Incremental Nonlinear Dynamic Inversion Controller". In: *IOP Conference Series: Materials Science and Engineering* 428 012048 (2018). DOI: 10.1088/1757-899x/428/1/012048.

[11] R. Steffensen et al. "Nonlinear dynamic inversion with actuator dynamics: An incremental control perspective". In: *Journal of Guidance, Control, and Dynamics* 46.4 (2023), pp. 709–717. DOI: 10.2514/1.g007079.

[12] R. Baggi et al. "Hierarchical Dynamic Control Allocation for over-actuated aircraft: Methodology and flight tests on a scaled-down model". In: *2022 IEEE Conference on Control Technology and Applications (CCTA)* (2022). DOI: 10.1109/ccta49430.2022.9966141.

[13] T. A. Johansen et al. "Control Allocation—A survey". In: *Automatica* 49.5 (Mar. 2013), pp. 1087–1103. DOI: 10.1016/j.automatica.2013.01.035.

[14] D. Enns. "Control allocation approaches". In: *Guidance, Navigation, and Control Conference and Exhibit* (1998). DOI: 10.2514/6.1998-4109.

[15] O. Härkegård. "Dynamic control allocation using constrained quadratic programming". In: *Journal of Guidance, Control, and Dynamics* 27.6 (2004), pp. 1028–1034. DOI: 10.2514/1.11607.

[16] H. Yang et al. "Research and experiment on dynamic weight pseudo-inverse control allocation for spacecraft attitude control system". In: *2019 Chinese Control Conference (CCC)* (2019). DOI: 10.23919/chicc.2019.8865693.

[17] A.J. Ostroff et al. "Enhanced NDI strategies for reconfigurable flight control". In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)* (2002). DOI: 10.1109/acc.2002.1024492.

[18] S. Sieberling et al. “Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction”. In: *Journal of Guidance, Control, and Dynamics* 33.6 (2010), pp. 1732–1742. DOI: 10.2514/1.49978.

[19] P. Acquatella et al. “Robust nonlinear spacecraft attitude control using incremental nonlinear dynamic inversion.” In: *AIAA Guidance, Navigation, and Control Conference* (2012). DOI: 10.2514/6.2012-4623.

[20] M. Naderi et al. “Guaranteed feasible control allocation using model predictive control”. In: *Control Theory and Technology* 17.3 (2019), pp. 252–264. DOI: 10.1007/s11768-019-7231-9.

[21] M. Oppenheimer et al. “Methods for compensating for control allocator and actuator interactions”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit* (2004). DOI: 10.2514/6.2004-5168.

[22] A. Tavasoli et al. “Comparison of static and dynamic control allocation techniques for Integrated Vehicle Control”. In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 7180–7186. DOI: 10.3182/20110828-6-it-1002.00398.

[23] R. de Castro et al. “Lyapunov-based control allocation for over-actuated nonlinear systems”. In: *2019 American Control Conference (ACC)* (2019). DOI: 10.23919/acc.2019.8814326.

[24] P. Kolaric et al. “Optimal Dynamic Control Allocation with guaranteed constraints and online reinforcement learning”. In: *Automatica* 122 (2020), p. 109265. DOI: 10.1016/j.automatica.2020.109265.

[25] B. Smit et al. “Adaptive incremental nonlinear dynamic inversion flight control for consistent handling qualities”. In: *AIAA SCITECH 2022 Forum* (2022). DOI: 10.2514/6.2022-1394.

[26] E. J. Smeur et al. “Adaptive incremental nonlinear dynamic inversion for attitude control of Micro Aerial Vehicles”. In: *AIAA Guidance, Navigation, and Control Conference* (2016). DOI: 10.2514/6.2016-1390.

[27] A. Cristofaro et al. “Fault-tolerant control allocation for overactuated nonlinear systems”. In: *Asian Journal of Control* 20.2 (2017), pp. 621–634. DOI: 10.1002/asjc.1619.

[28] S. S. Tohidi et al. “Adaptive control allocation for constrained systems”. In: *Automatica* 121 (2020), p. 109161. DOI: 10.1016/j.automatica.2020.109161.

[29] P. Simplício et al. “An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion”. In: *Control Engineering Practice* 21.8 (Aug. 2013), pp. 1065–1077. DOI: 10.1016/j.conengprac.2013.03.009.