Study and Development of an Adaptive Vertical Farm

Thesis Dissertation

Presented and publicly defended on March 24, 2025, for the attainment of the PhD title from

UNIVERSITÀ DI GENOVA
AND
UNIVERSITÉ DE LORRAINE

Ingegneria Matematica e Simulazione (XXXVII Ciclo)
&
Automatique, Traitement du Signal et des Images, Génie Informatique

par

Echrak Chnib

Jury members

President:	Angelo Alessandri	Full Professor, Università di Genova
Referees:	Enrica Zereik	Senior Researcher, CNR
	Naïma Aït Oufroukh	Associate Professor, HDR, Université d'Evry
Examiners :	Catherine Bonnet	Senior Researcher, Inria Paris–Saclay
	Angelo Alessandri	Full Professor, Università di Genova
	Benoît Marx	Associate Professor, HDR, Université de Lorraine
Supervisors :	Patrizia Bagnerini	Associate Professor , Università di Genova
	Ali Zemouche	Associate Professor, HDR, Université de Lorraine
	Mauro Gaggero	Senior Researcher, CNR

École Doctorale IAEM Lorraine

Centre de Recherche en Automatique de Nancy, Université de Lorraine

Dipartimento di ingegneria meccanica, energetica, gestionale e dei trasporti, Università di Genova

Mis en page avec la classe thloria.

"You will not attain knowledge except through six things:
Intelligence, Diligence, Effort, the Bare Necessities,
the Companionship of a teacher, and Patience over time."

– Imam Al-Shafi'i (767–820 CE)

Acknowlegments

This manuscript is the product of thought, but these words are shaped by emotion and gratitude. This thesis was carried out within the framework of a joint PhD program between the University of Genoa (Italy) and the University of Lorraine (France), in collaboration with Space V S.r.l.
I am deeply grateful to the Department of Mechanical, Energy, Management, and Transportation Engineering (DIME) at the University of Genoa and the Centre de Recherche en Automatique de Nancy (CRAN) at the University of Lorraine for providing me with the opportunity, support, and resources necessary to pursue this ambitious and enriching research project.

I would like to sincerely thank my thesis supervisors, Dr. Patrizia Bagnerini and Dr. Ali Zemouche, as well as my thesis co-supervisor, Dr. Mauro Gaggero. I had the privilege of spending three incredible and inspiring years working with them, and I am especially grateful for the passion they shared with me for research, along with their unwavering support, patience, and invaluable guidance throughout this journey. I will never forget the late evenings, the weekend reviews, or the countless insightful discussions that often led to new research ideas and directions. Your energy, rigor, and enthusiasm for research have left a lasting impact on me, both professionally and personally.
Patrizia, thank you for your smile, your positive energy, and your infinite kindness. Ali, thank you for your unwavering support and for always believing in me. Mauro, thank you for your steady presence, your infinite motivation, and the encouragement you brought at every stage of this journey.

I would like to express my heartfelt thanks to Dr. Angelo Alessandri for presiding over the defense jury, and for the many enriching discussions and memorable moments shared over coffee breaks at DIME. I am also sincerely grateful to Dr. Enrica Zereik and Dr. Naima Ait Oufroukh for accepting to review my thesis and for their thoughtful and constructive feedback, which greatly contributed to enhancing the quality of this work. My warm thanks also go to Dr. Catherine Bonnet and Dr. Benoît Marx, for accepting to be part of the jury and for their insightful questions and remarks on my research work.
A special thank you to Mr. Franco Malerba, CEO of Space V S.r.l. and the first Italian astronaut, for

the opportunity to be involved in the Adaptive Vertical Farm project, for his encouragement, and for honoring me with his presence during the defense.
I would like to extend my sincere thanks to Dr. Olivier Sename. My doctoral journey would not have been the same without the opportunity he offered me during my final-year engineering internship, which marked my first step into the world of research and became a stepping stone toward this thesis.

I am deeply grateful to my colleagues, professors, and administrative staff at DIME, CRAN, and IUT Henri Poincaré de Longwy for their constant support and encouragement, which provided an ideal environment for my work.
I extend my heartfelt thanks to my dear colleagues and fellow labmates from both labs in Genoa and Longwy: Hasni, Hichem, Shengya, Huy, Reza, Tiziano, Faezeh, Selsabil, Ayoub, Sofiane, Amir, Mahdi, Karima, and everyone who made my PhD journey as joyful and fulfilling as it was.
An immense and heartfelt thanks goes to Hasni, the first person I met in person on this journey and turned into a lifetime friend, with whom I have shared both laughter and tears over these years. See, we made it!
I would also like to express my immense gratitude to Shengya for traveling from Longwy to Luxembourg, Frankfurt, Milan, and finally Genoa, to support me during my defense. Your presence meant so much to me.

A warm thank you goes to the Istituto Suore del Buon Pastore, and especially to Suora Regina, for welcoming me during my stay in Genoa. You are my Italian "nonne"!
My love and gratitude go to my beloved people, whom I met in the halls of school and who have become lifelong companions: Ones, Wafa, Ameni, and Fakhreddine. Your infinite support, love, and countless video calls and reunions have been a constant source of energy and strength.

This thesis is the result of the love and belief of my family, for whom words can hardly express my gratitude. To my parents, who have always said, ” Knowledge is a light” and pushed me to reach the highest goals. Omi and Baba, I am overjoyed to see the pride in your eyes for your little girl.
To my sister Asma, my first friend and role model, thank you for always having my back over the years; thank you for inspiring me to follow this path in engineering and research.
To my little brother Mohamed, thank you for being my refuge, the best place to escape from stress and anxiety with laughter and joy. I dedicate this thesis to my wonderful family.

List of publications

International Conference Papers

[1] **E. Chnib**, P. Bagnerini, M. Gaggero, A. Zemouche, Dynamic Extended Output-Based Observer for an Adaptive Vertical Farm Quadcopter. *43rd IEEE American Control Conference, ACC 2025*, Denver, Colorado, USA, July 2025.

[2] **E. Chnib**, P. Bagnerini, M. Gaggero, A. Zemouche Parameter Estimation of a Dynamic Growth Model for Lettuce in an Adaptive Vertical Farm. *The 20th IEEE International Conference on Automation Science and Engineering, CASE 2024* , Bari, Italy, August 2024.

[3] **E. Chnib**, P. Bagnerini, M. Gaggero, A. Zemouche, Observer Design by Using a New Output-Based Dynamic Extension Technique. *The 22nd IEEE European Control Conference, ECC 2024*, Stockholm, Sweden, June 2024.

[4] **E. Chnib**, P. Bagnerini, A. Zemouche. LMI based H_∞ observer design for a quadcopter model operating in an adaptive vertical farm. *The 22nd IFAC World Congress, IFAC WC'23*, Yokohama, Japan, July 2023.

[5] P. Bagnerini, **E. Chnib**, M. Gaggero, A. Zemouche. Adaptive Vertical Farm for Space Farming *19th Applied Automation Technology in Teaching and Development, AALE 2022*, Luxembourg, Luxembourg, March 2023.

International Journal Papers

[1] **E. Chnib**, P. Bagnerini, M. Gaggero, A. Zemouche A Data-Driven Black-Box Dynamic Model for Lettuce Growth in an Adaptive Vertical Farm. *IEEE Transactions on Automation Science and Engineering* *(under review)*

[2] **E. Chnib**, P. Bagnerini, M. Gaggero, A. Zemouche State Estimation of an Adaptive Vertical Farm Quadcopter Using an Extended-Output Nonlinear Observer. *International Journal of Robust and Nonlinear Control* *(in preparation)*

Contents

Chapter 2
An Overview on the Theoretical Methods and Tools

Chapter 3
Data-Driven Black-Box Dynamic Models for Crop Growth

Chapter 4
Unmanned Aerial Vehicles (UAVs) in the Adaptive Vertical Farm

List of Figures

List of Tables

Acronyms and Notations

Acronyms

AVF	Adaptive Vertical Farm
VF	Vertical Farm
PA	Precision Agriculture
CEA	Controlled Environment Agriculture
UAV	Unmanned Aerial Vehicle
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying

Notations

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of real positive numbers
$\mathbb{R}^n$	Euclidean real space of dimension n
$\mathbb{R}^{n\times m}$	Set of real matrices of dimension $n \times m$
$\\|x\\|$	Euclidean norm of x, with $x \in \mathbb{R}^n$
$\text{ess } \sup_{t\geq 0} \\|x\\|$	essential supremum of the norm of the signal $x(t) \in \mathbb{R}^n$ for $t \geq 0$
$\\|A\\|$	Induced matrix norm defined as $\sup_{x\in\mathbb{R}^n} \frac{\|A\|}{\|Ax\|}$
$A > 0$	Symmetric positive definite matrix
$A \geq 0$	Symmetric positive semi-definite matrix
$\det(A)$	Determinant of the matrix $A \in \mathbb{R}^{n\times n}$
$\lambda_{\min}(A)$	Smallest eigenvalue of the matrix A
$\lambda_{\max}(A)$	Largest eigenvalue of the matrix A
A^T	Transpose of the matrix A
A^{-1}	Inverse of the matrix A
I_n	An $n \times n$ identity matrix
$0_{n\times m}$	Matrix of dimension $n \times m$ whose entries are all zeros
$diag\ (a_1, \ldots, a_n)$	An $n \times m$ diagonal matrix with a_i as its i-th diagonal element
$col\ (a_1, \ldots, a_n)$	Column vector with elements $(a_1, \ldots, a_n)$
$e_s(i) = (\underbrace{0, \ldots, 0, \overbrace{1}^{i\textbf{-th}}, 0, \ldots, 0}_{s \textbf{ components}})^T \in \mathbb{R}^s$	A vector of the canonical basis of $\mathbb{R}^s,\ s \geq 1$
$blkdiag\ (A_1, A_2, \ldots, A_n)$	Block diagonal matrix of matrices $A_1, A_2, \ldots, A_n$

Introduction

Due to climate change, increasing energy costs, high levels of urbanization, and rapid population increase, ensuring food security is becoming one of the more significant global challenges. The Food and Agriculture Organization of the United Nations [59] projects that by 2050, the world's population will have increased from 7.6 billion in 2024 to about 9.8 billion. This issue is made more critical by urbanization, being a major factor in both environmental change and economic growth. Arable lands are becoming scarcer as metropolitan areas are dramatically increasing, limiting space for conventional farming activities. As a result, traditional agricultural methods, along with the depletion of natural resources such as water and energy, are no longer sufficient to meet the requirements of modern societies over the coming decades.

Accordingly, closed-field agriculture has been evolving from simple covered greenhouses to more sophisticated protected cultivation such as vertical farming, a modern agricultural approach emerged as a sustainable solution [54]. This approach involves growing crops in vertically stacked layers, optimizing, as a result, the cultivated surface per unit of occupied area. In addition, these farms use, often, hydroponic or aeroponic systems growing variety of crops under controlled-environment agriculture (CEA) where the temperature, the humidity and the CO_2 concentration are constantly monitored [128]. As a result, water and energy consumption are decreased compared to traditional horizontal greenhouses. However, during the growth phase, crops typically have a fixed volume of space available in the traditional vertical farms, even though their height varies, resulting in a dynamic occupation of vertical space. Thus, the entire available vertical space is not fully exploited and the energy consumption reduction remains limited since unoccupied spaces are unnecessary air-conditioned and under artificial lighting systems [131]. Hence, improvements in the current vertical farms configuration such as modeling, automation, and real-time decision-making are required to enhance the efficiency and scalability of vertical farms.

A ninnovative vertical farming configuration have arose by 2021 with the goals of simplifying planting and harvesting tasks, while constantly monitoring environmental conditions to maximize efficiency : the Adaptive Vertical Farm (AVF) (See Figure 1). Unlike traditional fixed vertical farms, and based on the "adaptive principle" of the patent developed by Space V S.r.l [47, 71], the AVF is a modular structure that progressively adapts to the growth of the plants included in it, rather than adapting the crops to the structure in order to meet the crops requirements of a variable volume from planting to harvest. In other words, the distance between the shelves is automatically adjusted according to the growth of the plants. Hence, compared to a traditional vertical farm with fixed shelves, the main advantage is the potential to install more shelves within the given vertical space, resulting in a significant increase of the production yield.

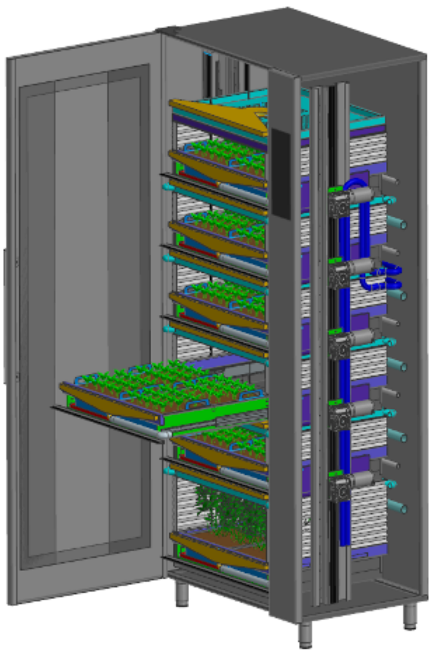


Figure 1 – Sketch of the Adaptive Vertical Farm concept.

Thesis Framework and Objectives

This thesis is conducted within the framework of a joint PhD program (french: convention cotutelle, italian: Convenzioni di cotutela) between the University of Genoa (Italy) and the University of Lorraine (France), in collaboration with Space V S.r.l. spin-off company from the University of Genoa. The project is funded by the "PON Ricerca e Innovazione 2014-2020" through the FSE REACT-EU program. The research was carried out at the Department of Mechanical, Energy, Management, and Transportation Engineering (DIME) at the University of Genoa and the Research Center in Automatic Control of Nancy (CRAN) at the University of Lorraine.
The primary objective of this thesis is part of a broader project aiming at developing the AVF, in order to establish an eco-friendly and resource-efficient farming system as an innovative solution of Precision Agriculture (PA). Aligning with the focuses of the PA field, the key goal of this thesis is to optimize and automate the AVF configuration, including variables such as leaf spacing, indoor environmental variables, and other agronomic factors to improve the crop yield and operational efficiency.

Thesis Contributions

The thesis proposes contributions to the development of the AVF in both the theoretical and computational aspects of PA, particularly in the areas of data-driven models and state estimation :

1. **Enhancing Crop Growth Modeling for Adaptive Vertical Farms Through Data-Driven Approaches :** To optimize the system and the control of shelf movements, an accurate crop

growth model, able to predict the heights of the plants at each time step, is crucial. Dynamic growth models of crops in greenhouse systems are available in the literature, however, many of these models have limitations. They typically depend on a large number of parameters to be tuned, require a solution of a set of Ordinary Differential Equations (ODEs) which may be complex and computationally demanding, are valid for specific crop types, and their accuracy often remains unproven. Thus, we propose the construction of a crop growth model directly from data collected on the field by using a black-box paradigm. The developed model relies on neural networks owing to their simplicity and approximating capabilities. The advantages of the proposed black-box model include :

- the adaptability to different crop types;
- the representation of a variety of dynamic behaviors;
- the reduce computational burden after the training procedure;
- the possibility of being used in innovative vertical farming systems, such as the aforementioned AVF, with very different characteristics than conventional systems.

2. **Contribution to LMI-based Observers for UAVs operating in the Adaptive Vertical Farm :** This contribution addresses the design of a nonlinear observer for UAV-based automation and their potential practical integration into the Adaptive Vertical Farm (AVF) system for different tasks (i.e., real-time monitoring of crop health, environmental conditions, growth patterns, etc..). Since the UAV is operating in a confined space including potential disturbances such as, for instance, variable artificial lighting conditions, close proximity to surfaces or crop density, an accurate and reliable state estimation is crucial for ensuring the UAV's effective and safe operation in the AVF.

   Thus, we propose in this thesis new Linear Matrix Inequality (LMI) based methods ensuring the stability of the observer design, ans as a result, providing practical tool for the operating UAV:

   - The first method investigates an $\mathcal{H}_\infty$ observer design via Linear Matrix Inequalities (LMI) aiming to provide accurate state estimation of the UAV where a novel less conservative LMI condition is applied to solve the $\mathcal{H}_\infty$ circle criterion design.
   - The second method introduces an observer design by using a new output-based dynamic extension technique which allows the decrease of the effect of the measurement noise in the state observer. We propose, hence, a new Linear Matrix Inequality (LMI) condition, ensuring the Input-to-State Stability (ISS) property of the estimation error.

Thesis Organization

The thesis manuscript is organized in different 4 chapters. A brief summary of each of the next chapters is given below:

- **Chapter 1** : introduces the key concepts of vertical farming and precision agriculture in Section 1.2 to motivate the innovation behind the Adapative Vertical Farm presented in Section 1.3 detailing the structure, competitive capacity compared to present solutions, the integrated technologies and its current applications.

- **Chapter 2** : establishes theoretical background on fundamental concepts applied in the continuity of this work. Section 2.2 is a brief overview on Artificial Neural Networks and Data-driven models followed by a brief state-of-the art on the nonlinear observer design in Section 2.3 presenting basic concepts, mainly the stability concept and the estimation for nonlinear dynamic systems.

- **Chapter 3** : is dedicated to the crop growth model in the AVF. It is organised as follow, Section 3.2 is dedicated to the lettuce growth dynamic model considered in this research work. Section 3.3 reports a parameter estimation strategy aiming to tune the dynamic model parameters. The proposed data-driven, black-box model is described in Section 3.4 whereas the simulation results are presented in 3.5.

- **Chapter 4** : addresses the need for accurate state estimation through observer design, for indoor operating UAV in the AVF subject to disturbances and noises such as sensor noise, lighting changes, and space constraints. The chapter is structured as follows: Section 4.2 introduces the an observer design via Linear Matrix Inequalities (LMI) and the $\mathcal{H}\infty$ criterion to ensure accurate state estimation under challenging conditions. Building upon this, Section 4.3 proposes a novel observer design approach using an output-based dynamic extension exploring the design in the continuous-time and discrete-time cases. Section 4.4 showcases the effectiveness of the proposed observer designs through the simulations performed on an indoor quadrotor model.

The thesis concludes with a summary of the contributions and proposes extensions and possible future works.

CHAPTER 1

The Adaptive Vertical Farm: A New Vertical Farming Solution

" We live vertically, so why can't we farm vertically? "

Dickson Despommier
Father of Vertical Farming

Contents

1.1 Introduction

In this chapter, we present in deep the concept of AVF, a forward-thinking solution aimed at addressing the challenges of modern agriculture. In the first part of the chapter, we start with Section 1.2, where we track the outlines of the vertical farming and precision agriculture evolution in order to highlight key developments in these fields and their pivotal role in shaping sustainable farming solutions. This overview provides a context to understand the groundbreaking technological advancement of the AVF, particularly in terms of crop volume occupancy and energy efficiency. The main focus of the second part is on the technical aspects and features of the AVF in Section 1.3, highlighting its key advantages compared to traditional existing vertical greenhouses. Additionally, we discuss the integration of Unmanned Aerial Vehicles (UAVs), emphasizing their role in enhancing farm automation to support PA. Finally, we present existing applications of the AVF in different farming contexts.

1.2 A Brief Overview on Vertical Farming and Precision Agriculture

In this section, we introduce a brief state of the art on two fundamental topics anf field of our research work: Vertical Farming in Subsection 1.2.1 and Precision Agriculture in Subsection 1.2.2 to highlight the innovation of the AVF and its adaptive principle compared to the existant modern farming solutions

1.2.1 Vertical Farming: Concept and Key Considerations

Although the concept of vertical farming has been newly introduced as a sustainable solution to meet the needs of modern societies, as motivated in the introduction of this research work, the roots of the concept can be traced back to the ancient era, around 600 BCE, where " Hanging Gardens of Babylon " are considered an early example of vertical agriculture [13]. However, the modern concept of vertical farming took a more structured form in the 20th century. In the 1910s, Gilbert Ellis Bailey first established the term " vertical farming " as part of a vision to grow food in urban environments using vertical space. His concept focused on creating a method that would not rely solely on traditional soil farming, but instead using hydroponic farming and artificial lightening in a controlled vertical environment to improve the supply of food along with economic and environmental benefits [13, 81]. The concept of vertical farming gained significant attention in the late 20^{th} and 21^{st} centuries as a solution to the challenges posed by urbanization and population growth to optimize space, reduce the need for land and afford for the growing demand and the popularity of organic food [1] impacting the global vertical farming market that reached USD 5.6 billion in 2022 and predicted to exceed USD 35 billion by 2032 1.1. In this context, " Apparatus for the Artificial Cultivation of Plants" is the first detailed technical description of a three-dimensional space for the cultivation of plants independently of seasons and climate conditions [92]. However, Despommier, a key figure in modern vertical agriculture, made an important contribution to the development of the concept of vertical agriculture in its current form with his vision of growing food in urban centers. In 1999, the concept of the " Vertical Farm (VF) " was introduced as a method of food production in multi-storey buildings in the context of a broader concept of " Controlled Environment Agriculture (CEA) ". CEA involves the management of generated indoor climate conditions, such as temperature, humidity, light, and carbone dioxide levels, in order to optimize plant growth and enable year-round production, independent of external conditions. Popularized through his book " The Vertical Farm: Feeding the World in the 21st Century (2010) " [49], Despommier emphasized the potential of the concept to renew urban spaces and reduce the environmental footprint of traditional agriculture. The idea was, first, argued as a contribution to sustainable food production by maximizing yields per square meter, reducing water and pesticide use, and minimizing land requirements, eventually offering a more eco-friendly alternative to conventional farming. However, over the years, the key sustainable benefits of vertical farming has been spanning environmental, social, and economic dimensions. From an environmental perspective, vertical farming has the potential to lower food miles by minimizing transportation distances, reducing water usage with advanced irrigation and recycling techniques, and recycling organic waste, which reduces reliance on landfills. Furthermore, vertical farming plays a role in managing floods, droughts, and seasonal variations, improves energy efficiency through the use of renewable energy sources, and promotes biodiversity by integrating nature into urban settings. From a the social point of view, vertical farming contributes to improving air quality and people's health, making potable water more accessible, creating local jobs, and offering opportunities for social interaction within local farming communities. Economically, vertical farming reduces energy, packaging, and transport fuel costs, offers greater yields, and decreases costs by reducing dependence on fertilizers, herbicides, and pesticides [36,46,73].

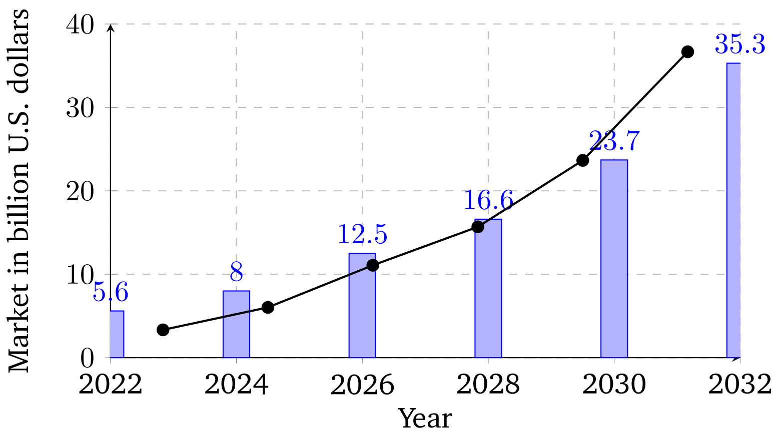


Figure 1.1 – Vertical Farms' Market Growth Projection for 2022-2032 (adapted from [1])

1.2.1.1 General Structure and Types of Vertical Farms

The general architecture of a VF is designed to ensure optimal space and resource use along with efficient and sustainable food production operations. This is achieved thanks to interconnected subsystems ensuring an optimal crop growth regardless of external climate conditions. According to [13,49,125,131], the main subsystems of a VF are the following:

- Infrastructure: VFs, in its simple principle, are vertically stacked cultivation shelves that may constitute multi-storey buildings often located in urban areas where land is scarce.

- Growing Mediums and Irrigation Systems: Unlike traditional greenhouses, VFs rely on soilless growing methods, such as hydroponics, aeroponics, or aquaponics detailed in 1.2.1.2. These farming techniques provide plants with the correct balance of nutrients, water, and pH levels, ensuring optimal growth conditions. This type of farms are closed-loop systems that recycle water.

- Artificial Lighting System: To ensure year-round farming independent from natural sunlight through the photosynthesis operation, VFs rely on light-emitting diode (LED) to control the light spectrum, intensity, and duration based on the crop requirements.

- Climate Control Systems : These systems regulate the climate growth conditions such as temperature, humidity levels and carbon dioxide enrichment to ensure optimal plant growth thanks to monitoring techniques.

Nowadays, as a result of the advancements achieved in the field, other related indoor farming systems based on CEA such as high-density vertical growing (HDVG), building-integrated controlled-environment farms (BICEFs), plant factories with artificial lighting (PFALs) and plant production systems with artificial lighting (PPALs) may be included within the scope of vertical farming. Multi-storey farms, mainly " skyfarms " are less common used terms but refers to agricultural practices conducted in high-rise buildings [13, 92, 125]. To address this multitude of interconnected indoor farming systems with relation with VFs, [165] provided a comprehensive overview of these different vertical farming concepts, detailing their characteristics and classifying them based on influencing factors such as size, density, control systems, arrangement, architectural design, and site selection as represented in Figure 1.2 adapted from [125].

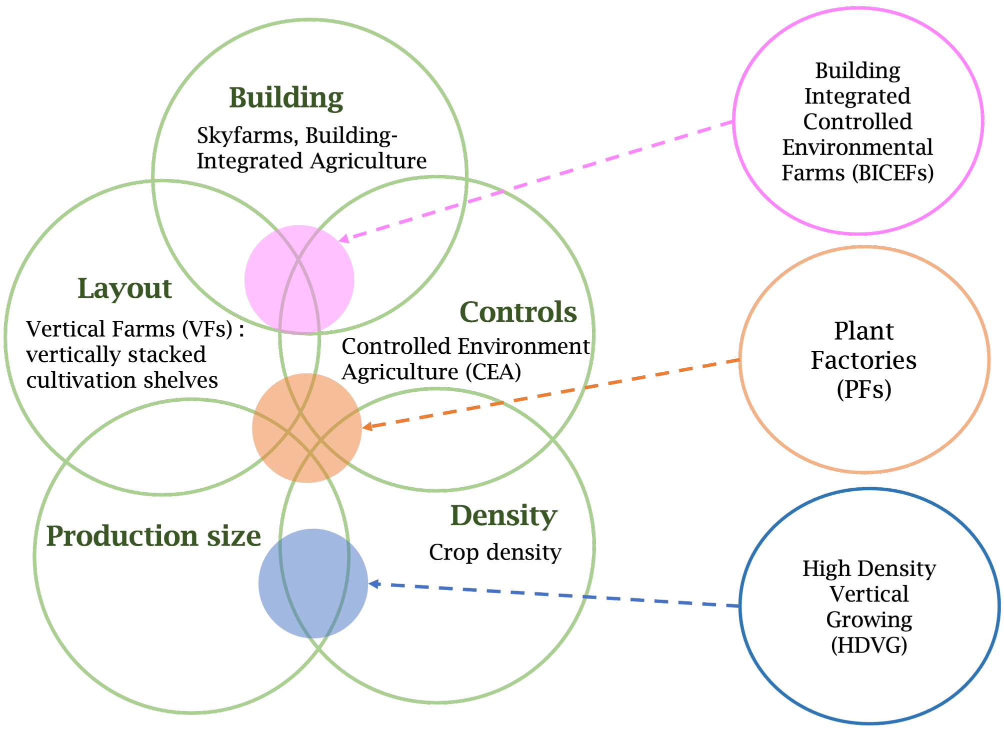


Figure 1.2 – Types of Vertical Farms and Influencing Factors (adapted from [165])

Table 1.1 provides a list of the main leading projects of VFs (installed and in progress) in different cities around the world, mainly in Europe, Asia, and America.

1.2.1.2 Vertical Farming Methods

Over the years, in order to maximize productivity and sustainability of VFs, researchers have been developing and improving farming methods for eventual deployment and better integration into urban planning [55, 56]. The main system paradigms along with their characteristics, key benefits and main technologies are presented in the following and summarized in Figure 1.3 :

1.2.1.2.a Hydroponics

Hydroponics refers to a farming technique that involves growing plants without soil, basing on nutrient-rich water solutions instead. This method allows accurate management of nutrient supply,

Table 1.1 – Summary of existing VF's projects.

Vertical Farm	**Location**
Nuvege	Kyoto, Japan
PlantLab	Den Bosch, Holland
Skygreen	Singapore
FarmedHere	Bedford Park, Illinois, USA
The Plant	Chicago, Illinois, USA
Green Girls Produce	Memphis, Tennessee, USA
Brooklyn Grange	Brooklyn, New York, USA
Gotham Greens	Brooklyn, New York, USA
AeroFarm	Danville, Virginia, USA
iFarm	Dubai, UAE & Riyadh, KSA
Plantagon	Sweden
La Tour Vivante	France
Harvest Green Tower	Vancouver, Canada
Skyfarm	Toronto, Canada

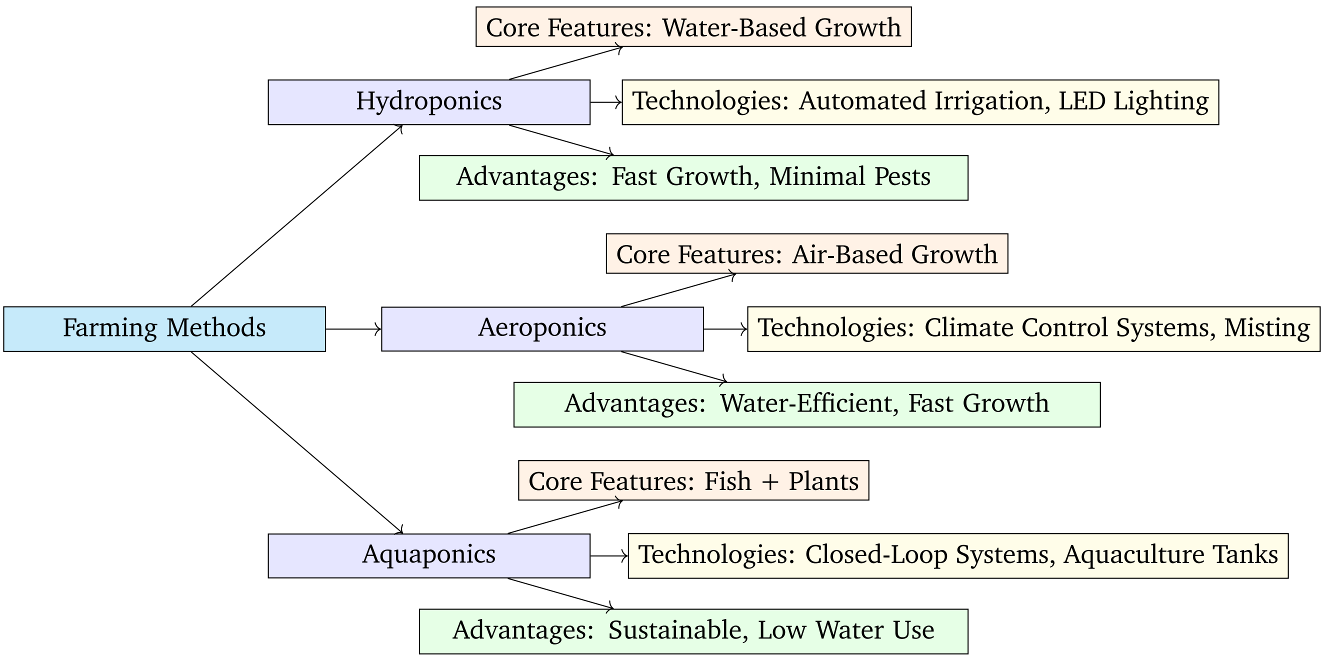


Figure 1.3 – Diagram of Farming Methods: Features, Advantages, and Technologies

allowing crops to obtain the exact mix of necessary nutrients for optimal growth [50]. Figure 1.4 shows the structure of the hydroponics VF in Sweden [96] where the system (A) represents Nutrient Film Technique. For such hydroponic system, each tray has LED illumination above it. Water is pushed into a channel where the plant roots hang. Nutrient water flows (shown by black arrows) continue down the levels, collecting in a reservoir before being recirculated back to the system's top. The hydroponic system (B) is a drip irrigation system where the nutrient water is drip-fed down the column. Wicking substance in the column draws water to the roots. The water that is not consummed by the plants drains out the bottom of the tower and is recirculated.,This arrangement can provide more natural light or be illuminated from the side with LEDs. While the idea of soil-less cultivation dates back to ancient times, it has only become widely adopted in recent years, especially within modern agricultural configurations like VFs and space exploration field where the National Aeronautics and Space Administration (NASA) investigates its viability for cultivating crops (i.e. lettuce, onions, radish, etc..) in controlled settings, mainly the International Space Station (ISS) [167]. Compared to traditional soil-based farming, hydroponics mitigates many of the challenges associated with soil, such as pests. The technique is also water-efficient, requiring significantly less water than conventional farming since the water in hydroponic systems is circulating in a closed-loop, it minimizes waste and supports sustainable agricultural practices, particularly in areas where water resources are scarce. Additionally, hydroponic systems are relatively low-maintenance and labor-efficient, as they eliminate the need for soil cultivation tasks like dirt removal and kneeling. Hydroponics also allows for better control of plant growth conditions. By using inert growing mediums such as nutritive solutions, it enables uniform distribution of nutrients, leading to consistent growth. This precision ensures that plants receive the necessary nutrients without the variability introduced by natural soil conditions. The method is considered " cleaner " since it helps reduce the risk of diseases commonly associated with soil-based agriculture.

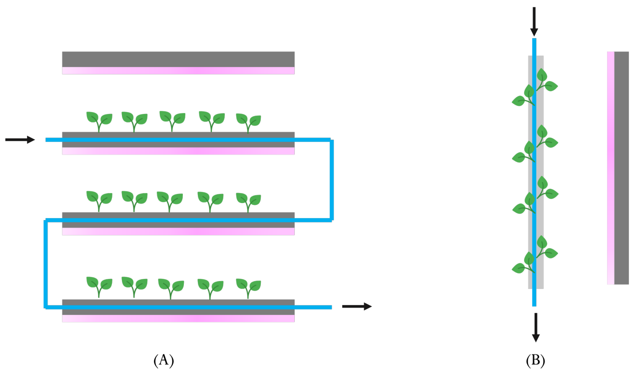


Figure 1.4 – Hydroponic Vertical Farming System (adapted from [70])

1.2.1.2.b Aeroponics

Similar to the hydroponic paradigm, the aeroponic method is soiless farming technique, using a mist or aerosolized nutrient solution applied to the crops' roots. These roots are suspended in the air, allowing them to absorb the nutrients and oxygen necessary for growth. The absence of soil offers several advantages, including increased root oxygenation and faster plant growth [139]. In contrast to hydroponics, depending on nutrient-rich water solution as the growing medium, aeroponics uses air-mist as to deliver essential nutrients through aerosols minimizing, thus, water waste and promoting uniform growth. According to [37], this method reduces water consumption significantly, using up to 90% less water compared to the most efficient hydroponic systems. The system can be employed in various setups, including vertical farming configurations, where plants are stacked in layers to maximize space efficiency along with allowing the crown to grow upward and the roots downward freely. This technique allows, therefore, for better space use, particularly in urban environments where land and water are scarce [13, 139]. One of the leading aeroponic-based VFs is " AeroFarms " for growing leafy greens and herbs [12]. Furthermore, in the context of space farming, XROOTS experiment of the NASA, evaluated the aeroponics technique in order to improve the efficiency of farming in terrestrial greenhouses and controlled environments [89]. Figure 1.5 is an illustration of the concept the aeroponic farming method.

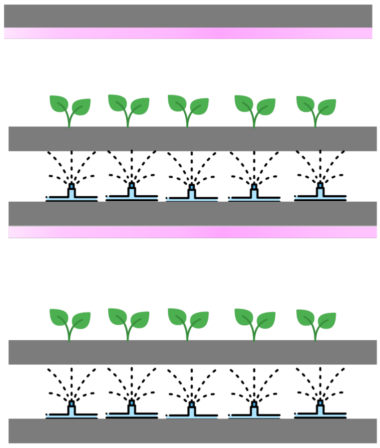


Figure 1.5 – Aeroponic Vertical Farming System

1.2.1.2.c Aquaponics

Aquaponics is farming method combining aquaculture (fish farming) and hydroponics. In aquaponics, fish waste is considered as the organic nutrient source of the growing medium of the plants, which in their turn help filter the water, recirculated, then, to the fish tanks [13]. The symbiotic relationship between the fish and plants creates an enclosed sustainable ecosystem that supports waste and water recycling and reduces the need for chemical fertilizers, pesticides, and herbicides compared to traditional farming systems [164]. Figure 1.6 is a simplified schematic diagram of the principle of this method.

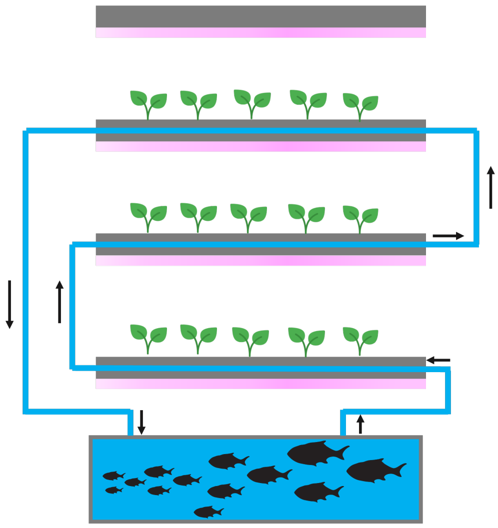


Figure 1.6 – Aquaponic Vertical Farming System

1.2.2 Precision Agriculture: Technological Trends

As a response to the growing need for more efficient and sustainable agricultural practices, Precision Agriculture (PA) appeared in the late 20^{th} century driven by advancements in technology that made possible a more accurate monitoring and a better decision making for farmers, particulary, the development of Global Positioning System (GPS) technology, along with Geographic Information Systems (GIS) enabling accurate mapping and real-time monitoring of field conditions [69, 126].

The primary goal of PA is to make the best use of resources such as water, fertilizers, and pesticides by adapting them to the exact needs of individual crops or fields. This approach ensures " smart" practices that no only improves crop yields but also minimizes waste, environmental impact, and promote sustainability in farming practices [53].

1.2.2.1 Sensors and Actuators

In smart VFs, sensors and actuators play a crucial role in enhancing automation and efficient management [124] . The need for constant human intervention is reduced by collecting critical data on indoor climate parameters or the health of crops. Actuators respond to the collected data by regulating the equipment related to captured parameters, thus maintaining optimal farming conditions. The interaction between actuators and environmental conditions captured by sensors is crucial for achieving optimal crop growth with limited human involvement. To ensure the CEA, the main sensors and actuators present in a VF are listed as following :

- *Temperature and humidity sensors :* monitor the environmental conditions in ambient air, ensuring optimal ranges, i.e. for leafy greens the optimal ranges are $18 - 24°\mathrm{C}$ and $50 - 70\%$.
- $\mathrm{CO_2}$ *sensors :* regulate the concentration of carbon dioxide in the farm, e.g. optimal range for leafy greens is $800 - 1500$ ppm.
- *Light intensity sensors :* control the light spectrum to ensure the appropriate light levels for photosynthesis, e.g. the optimal light range for leafy greens is $50 - 250$ PPFD.
- *Electrical conductivity (EC) :* monitor the dissolved minerals in the growing medium, e.g. the optimal EC range for leafy greens is $1.4 - 1.8$ mS/cm
- *pH sensors :* measure the acidity of the solution, i.e. for leafy greens, the optimal pH range for is $5.5 - 6$.
- *Ion-selective electrodes (ISE) :* evaluate nitrogen concentration in the nutrient solution, e.g. the optimal nitrate range for leafy greens is $100 - 150$ ppm.
- *Dissolved oxygen (DO) :* monitor oxygen levels in the nutrient solution, e.g. the optimal DO range for leafy greens is 78 ppm.
- *Water level sensors:* assess the solution levels in nutrient and stock-solution tanks.
- *Heater:* increases the ambient air temperature to maintain optimal conditions for plant growth.
- *Cooler:* decreases the ambient air temperature when needed to ensure the farm remains within the desired temperature range.
- *Humidifier:* increases ambient air humidity.
- *Dehumidifier:* reduces ambient air humidity.
- *LEDs:* provide the required light intensity for plant growth to mimic sunlight for photosynthesis.
- *Fans:* circulate the ambient air within the farm to ensure proper airflow.
- $\mathrm{CO_2}$ *regulator:* Maintains the reference value $\mathrm{CO_2}$ concentration among the shelves to ensure plant photosynthesis.
- *Nutrient pump:* distributes the needed nutrient solution to the plants' roots.

1.2.2.2 Unmanned Aerial Vehicles (UAVs) in Vertical Farms

Vertical farming merged with UAVs technologies represents a recent pillar of PA, refining farming practices to increase the automation of the farms in the framework of the Industry 5.0. To overcome global issues related to agricultural activities, the convergence of these two fields let open doors to new techniques of dynamic crop monitoring decreasing the number of sensors and actuators and, thus, the cost of the monitoring subsystems. The idea is basically transferred from open-field agriculture where drones are used to provide timely processed data leveraged to support critical crop management decisions. The most frequent applications of UAVs are crop growth monitoring and yield estimation, health monitoring, disease detection, spraying and irrigation where quadcopters are widely used for their simple mechanical structure, swift maneuvrability, and increased payload compared to other UAVs. A detailed review and a comparative study between different types of agricultural drones is proposed by researchers in [136] to highight the best configurtion and the trending applications in the field. The advantages of the UAV integration in the traditional agriculture motivated the use of it in the closed-field farming configurations [125]. For traditional high-scaled greenhouses, Corvus Drones [52] has integrated drones to perform tasks like seed germination and flowering detection that farmers can easily pilot through smartphone application, involving the technology of Internet of Things (IoT). The technology is deployed in VFs to perform similar tasks [26]. In the same context, AeroFarms [12] opened the door forovation in collaboration with Nokia Bell Labs to develop and refine new methodologies through testing with commercial crops. UAVs navigate the VF and capture imagery of crops gathering data on health metrics (i.e. leaf area, stem length, leaf color, spotting, etc). The system is designed to support 5G private mobile network paired with cloud-based data processing to ensure data security in an on-site network environment. Similiraly, iFarm [86] captures imagery to evaluate plant weight, collected images and determine the optimal lighting settings by analyzing power consumption and the rate of leaf expansion as shown in Figure 1.7.

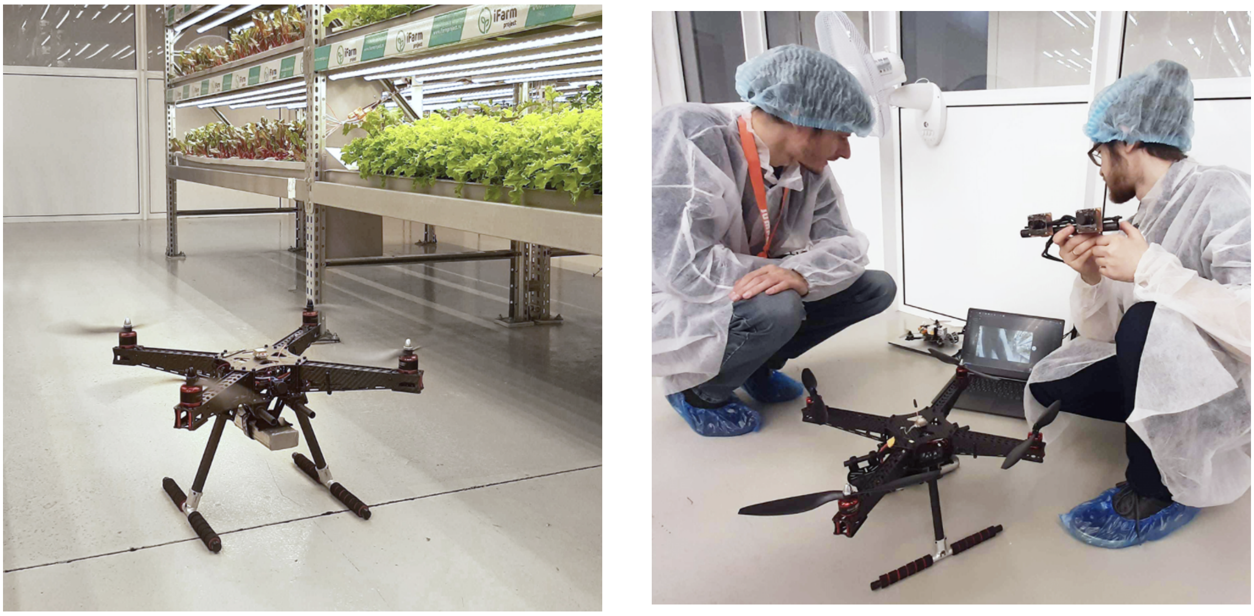


Figure 1.7 – UAV performing in the iFarm [86]

1.2.2.3 Artificial Intelligence (AI) Based Integrated Tools

The automation of vertical farming can be improved using Artificial Intelligence (AI) as a reliable computational tool that enable systems to learn and adjust its behavior based on present data where machine-learning algorithms are employed to identify patterns and predict crop growth state enabling growers to make informed decisions [125]. To maximize crop yield, enhancing crop growth models based on AI-based approximations improves the ability to accurately simulate the growth of large numbers of crops, for which the number of involved parameters is important which may explode the computational costs. The tool is useful for reproduce major patterns and prediction of yield [58]. In the litterature, different AI algorithms are used in vertical farming for plant monitoring, growth prediction, pest detection, and stress identification for different types of crops. Convolutional Neural Networks (CNNs) used in [119] to monitor plant leaf has demonstrated a high accuracy for lettuce disease detection with 86%. For more complex crop patterns, such as height or leaf weight, 3D plant modeling combined with deep segmentation techniques are used on the basil, as proposed in [25]. Other applications such as deep-learning models are used to monitor leaf stress and abnormalities, achieving an accuracy rate range between 83.26% and 94.15% [169, 173]. Furthermore, a neural network approach has been investigated in [6] to develop a model of plant growth prediction using the backpropagation Artificial Neural Network (ANN) method with environmental parameter input at the plant factory. Different network architectures were assessed to confirm the effectiveness of the developed model in predicting lettuce height based on the RMSE validation results. One to seven nodes were used in the experiment and the results showed that the 4-7-1 network architecture was capable of estimating the lettuces height.

1.3 The Adaptive Vertical Farm (AVF)

This section is dedicated to presenting the main focus of this research work: the Adaptive Vertical Farm (AVF) which we aim to develop through this thesis. Subsection 1.3.1 introduces the conceptual design of the AVF, outlining its foundational structure and principles. Subsection 1.3.2 discusses the efficiency and sustainability of the AVF, emphasizing its potential to optimize resource usage and reduce environmental impact. Subsection 1.3.3 addresses the integration of UAVs within the AVF system, highlighting their role in monitoring and maintenance. Finally, Subsection 1.3.4 presents emerging applications that highlight the broader potential of the AVF concept to address contemporary agricultural challenges.

1.3.1 Conceptual Design of the Adaptive Vertical Farm

The idea of AVF is based on the “adaptive principle” introduced by [71], which consists of gradually calibrating the farm structure to the growth of the plants cultivated therein, as shown in Fig. 1.8. The major advantage, compared to the fixed-shelve farm, investigated in 1.2, is the ability to insert more shelves in the same vertical space, which mainly results in an increase of the production yield and, thus, the decrease of the unnecessary air-conditioned volume. The variable number of cultivation shelves has the following requirements (see Figure 1.8):

- the distance between the shelves is adjustable due to an automated mechanism able to vary the position of each shelve according to the crop height during the production cycle;
- the crop height in each shelf is measured in real time by means of proper sensors;
- the airflow is constant and uniform owing to an aeration formwork within each shelf.

The AVF has a modular structure, and Fig. 1.9 sketches the parts of each module, or shelf. The set of technical space for handling, measurement, lighting, and ventilation subsystems is enclosed within the variable height h_m. The variable part that is adapted according to the height of the crop growing is given by h_r. However, the fixed height part always occupies the same part of the total height of

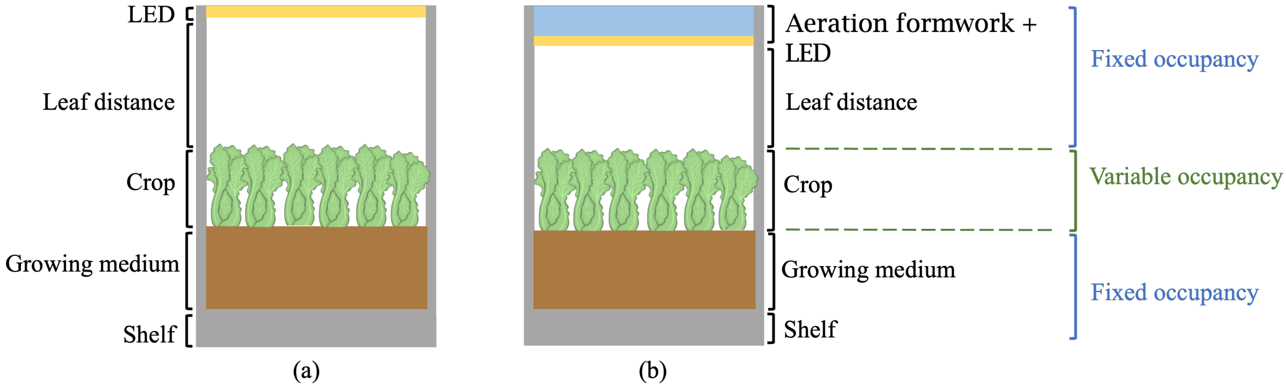


Figure 1.8 – Structure of the fixed-shelf greenhouse (a) and the AVF (b) with respect to vertical occupancy.

the farm. It is composed of the height of the aeration formwork, the artificial LED lighting system, as well as the rest of technical compartment (denoted by h_t), the height of the cultivation substrate (soil, water, root system) h_s, and the leaf distance h_f of the top of the crops from the formwork, ensuring air circulation and proper lighting supply. The leaf distance, i.e., the distance between the top of the plant and the shelf above, is a key parameter of the adaptive greenhouse compared to current industrial vertical farms. The leaf distance is selected in a manner that decreases the volume needed for each crop, thus maximizing the production yield, while maintaining the well-being conditions of the plants (also owing to the aeration formwork of each shelf that dissipates the heat provided by the LED lighting system). The separate micro-conditioning system helps generating different micro-climates for each shelf allowing growing non-homogeneous types of plants on each shelf according to their corresponding temperatures and humidity rates.

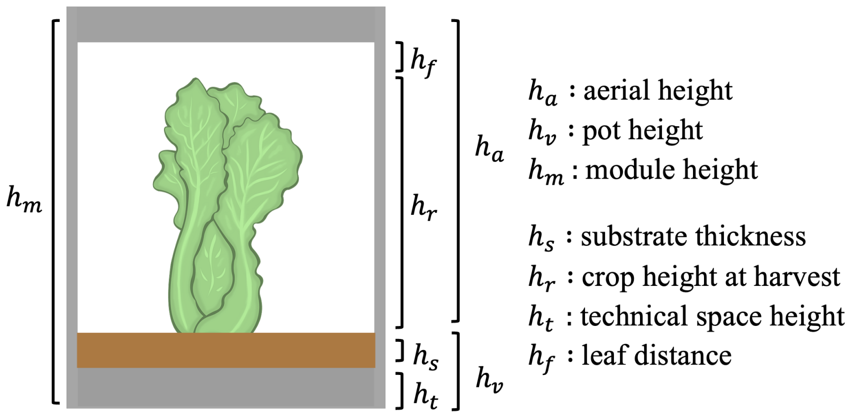


Figure 1.9 – Scheme of the parts of each AVF shelf.

1.3.2 About the Efficiency and the Sustainability of the AVF

In order to study the efficiency of this configuration in terms of production, a preliminary study was performed in [30] at the very early phase of the AVF project. The estimation of the production yield of the AVF is influenced by several factors, including the crop type, which determines the final height at harvest, the duration of the production cycle, and the growth curve. Additionally, technical characteristics of the greenhouse, such as total height, the height of technical compartments (e.g., trays containing the crops, leaf spacing), and the number of shelves that can be inserted, also play a crucial role. Unlike vertical farms (VFs) with fixed shelves, where all shelves can be cultivated simultaneously without sowing constraints, AVFs require an optimal sowing scheduling algorithm to maximize the number of harvests while fully use the total height. To this end, a first algorithm was introduced in [30] to determine the best strategy to maximize production yield, while also considering other factors such as energy efficiency, crop health, and growth rate etc. A comparison with a similar VF of the same size was performed to estimate the increase of production yield resulted from the adaptive technology. In Figure 1.10, the preliminary results contained in [30] are reported.

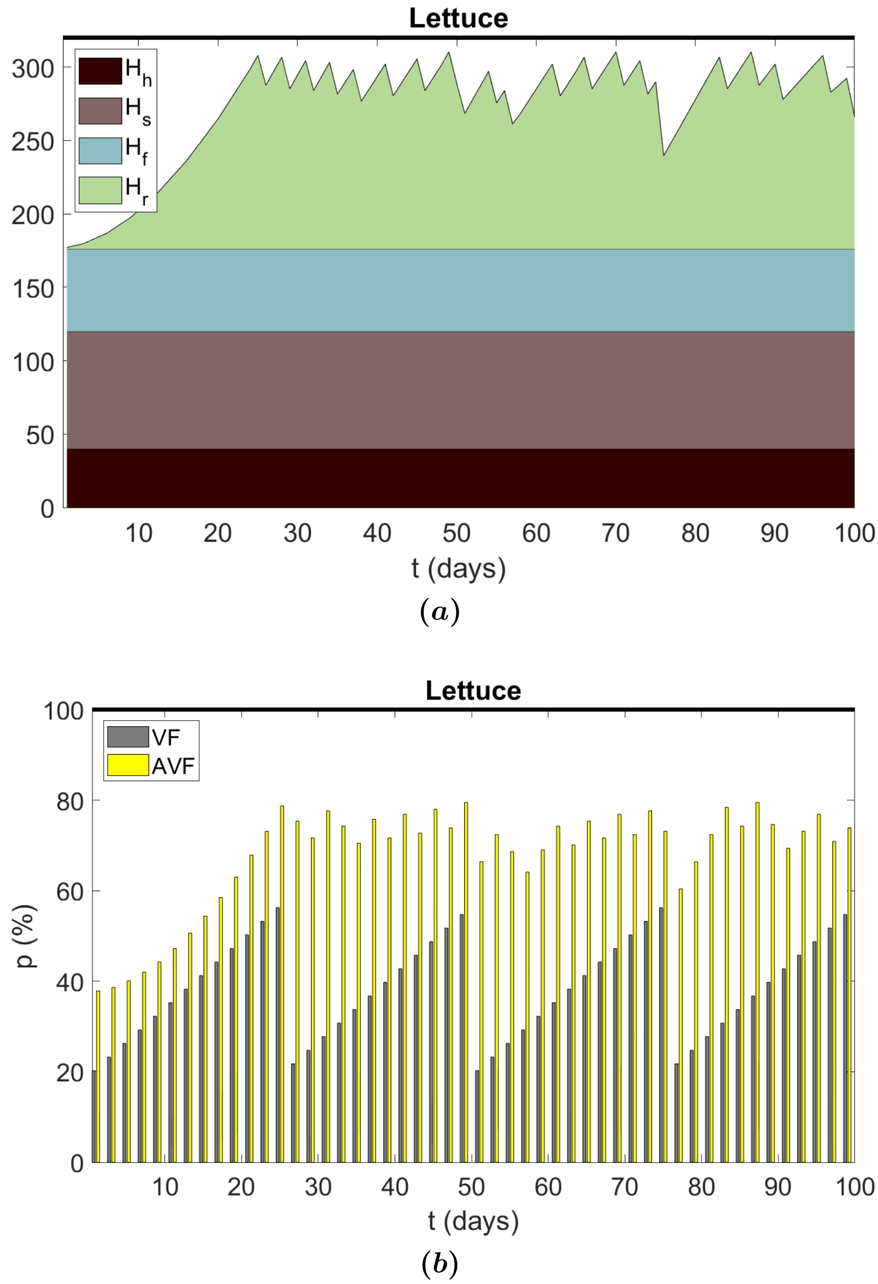


Figure 1.10 – Vertical space occupancy of an AVF (cm) (a) and vertical space occupancy (%) of a fixed shelf VF compared to AVF (b)

For this study, a simulation for lettuce cultivation is considered and presented in Figure 1.10. It

illustrates the total occupancy of the various fixed and variable components for each growing day. The results show that the AVF allows to occupy nearly 100% of the available volume in steady state. As noted in [30], the number of harvests within the simulation time considered of 100 days increases from 12 to 25 compared to a VF of the same size, corresponding to a 108% increase in the production yield. Figure 1.10 also shows the total percentage occupancy (excluding leaf spacing for the sake of simplicity in this preliminary study) of the AVF and of the corresponding fixed-shelf VF. This highlights the ability of AVF to occupy significantly more volume, as more shelves can be inserted at the same total greenhouse height of 310 cm, which naturally leads to a corresponding increase in production yield. As previously stated, the adaptive greenhouse is different from existing industrial VFs since its cultivation shelves are moveable, allowing for dynamic adjustment of the space between the top of the plant and the shelf above (i.e. the leaf distance as presented in Figure 1.8. In order to maximize production yield, this key feature of leaf distance, along with he aeration formwork of each shelf that dissipates the heat produced by the LED lighting system, allow to minimizing the volume required for each crop while maintaining the optimal growth conditions for the crops.
As reported in [27], growth models are crucial to simulate crop performance considering different types of plants and various parameters influencing the growth process. To address these challenges, Chapter 3 focuses on crop growth modeling using data-driven models approach to determine the best configurations regarding leaf distance, light intensity, CO_2 concentration and other aspects related to the CEA.

1.3.3 Integration of the UAVs in the Adaptive Vertical Farm

Figure 1.11 – Illustration of a quadcopter operating in the AVF

In the context of PA, the integration of UAVs, particularly quadcopters as mentionned in 1.2.2, enhance significantly the automation for the AVF aligning with the trend of smart farming observed in the existent open/closed-field agriculture sets [148]. Equipped with sensors, UAVs enable the continuous monitoring, detecting eventual diseases, assessing irrigation and nutrient needs, helping to optimize resources and allowing for a more informed decision-making during production cycle. In more details, real-time data is collected on crop health, environmental conditions and growth patterns, as the drone roams the farm, which is crucial for maintaining well-being conditions of

the plants, as illustrated in Figure 1.11. This technology proves particulary valuable when the AVF concept is scaled up to large, industrial plant factories for a sustainable food production. While stationary sensors enable monitoring specific environmental factors such as temperature, humidity, light intensity, and CO_2 levels, agricultural drones provide a wider perspective owing to their a bird's-eye view of the entire farm, collecting visual data and covering areas that stationary sensors may not reach easily, or where increasing the number of sensors could raise fabrication costs. The " Vertical Take-Off and Landing Autonomous/Semi-autonomous/Remote-Controlled Aerial Agricultural Sensor Platform " [121] is a relevant example of such technology in action.
Moreover, UAVs support the dynamic architecture of the AVF where the leaf distance is the key parameter. Knowing that this parameter varies according to crop height during the growth cycle, UAVs provide an extra layer of data for better comprehensive plant height monitoring. In this framework, authors in [26] proposed a novel approach for assessing plant growth dynamics in fixed-shelve VF containers using crop images captured by UAVs. This method involves processing captured images to calculate the average height of plants in each container, then storing the results in a database enabling comparison of current height measurements with previous data to evaluate growth dynamics. In fact, most of the stationary sensors providing height measurements (e.g. ultrasonic sensors, capacitive sensors, or optical sensors, etc.) require careful positioning to avoid disrupting the plant's natural growth cycle through shading effects, mechanical or electrical interference. Given the dynamic nature of the AVF and the need for real-time, flexible monitoring of crop height, using UAVs with dynamic sensors (e.g. Time-of-Flight (ToF) cameras or stereo vision systems, etc..) is indeed a more suitable and efficient approach. This setup maximizes the efficiency and sustainability of the AVF, while achieving the " affordability - accuracy - scalability" compromise for better performance of such dynamic farming systems.
However, while they present significant advantages, an accurate control of UAVs requires an accurate state estimation that can be affected by external disturbances, such as propellers noise, crop density, light conditions, etc. Thus, in order to ensure the reliability and the efficiency of the AVF, state estimation techniques are essential. Chapter 4 will delve into state UAV estimation ensuring the robustness of the overall system.

1.3.4 Emerging Applications

The concept of the adaptive multilayer greenhouse was originally developed by Germina S.r.l. [157] as part of its efforts to provide sustainable solutions for urban fresh food production. This innovation, which was initially designed for use in horizontal greenhouses, vertical farms, and home cultivation systems, was later adapted for space applications. The technology was acquired by Space V S.r.l. [47], which transferred it to the aerospace sector. This technology is driven by the need to provide astronauts with locally grown fresh food during long-duration missions on the International Space Station (ISS). Space farming has long been a subject of study how microgravity affects plant growth. One of the most famous space cultivation experiments is the " Veggie " system, installed on the ISS in 2014 [66]. The system employs artificial LED lighting to simulate Earth-like conditions for plant growth in microgravity, and has successfully cultivated various crops, including lettuce, radishes, and zinnias. In this context, several studies are conducted to develop the AVF system for a space-based agriculture, where minimizing volume and maximizing resource efficiency are vital for supporting human life in orbit and future lunar settlements. In [132–134], a scheduling approach of seedings based on the solution of mixed-integer linear programming problems is presented to maximize the overall number of sowings, and therefore of harvests, in a given timeframe. The availability of an optimal schedule of seedings enables the implementation of suitable control strategies to vertically move the shelves in order to track the growth of the crops. Space V S.r.l has developed a Proof of Concept (PoC) demonstrator of the AVF, presented in Figure 1.12. It includes three cultivation shelves that adjust dynamically in response to a simulated " accelerated

Figure 1.12 – Proof of Concept of the AVF showcased in the International Astronautical Congress (IAC), Milan 2024

growth" cycle of crops. It is worth to note that while this demonstrator is not intended for flight, several of its subsystems have been designed with space qualification requirements, ensuring the rigorous standards necessary for eventual space applications.

1.4 Conclusion

The AVF represents a promising advancement in addressing future challenges of food security, resource management, and sustainability, both in terrestrial and space environments. Owing to its dynamic architecture and automation during the crop growth cycle compared to fixed-shelves exsisting VFs, the proposed solution is worth to investigate in deep in terms of crop growth and monitoring to ensure its full potential in both terrestrial and space settings.

CHAPTER

2

An Overview on the Theoretical Methods and Tools

" It is not the strongest of the species that survives, nor the most intelligent, but the one most adaptable to change."

Charles Darwin

Contents

2.1 Introduction

This chapter provides the theoretical background necessary for understanding the key concepts that will be applied throughout the continuation of this work. Section 2.2 offers a concise overview of Artificial Neural Networks (ANNs) and Data-driven models, introducing their fundamental principles and highlighting their relevance to the modeling and estimation tasks addressed in Chapter 3 of the present thesis. Following this, Section 2.3 presents a brief state-of-the-art review on nonlinear observer design. This section focuses on the essential concepts of observer theory, with particular emphasis on stability analysis and state estimation for nonlinear dynamic systems. These foundational ideas form the basis for the observer design methods explored later in the work, enabling effective handling of disturbances and uncertainties in system modeling explored in the Chapter 4.

2.2 A Brief Overview on Artificial Neural Networks and Data-driven Models

The Section 2.2 is devoted to introducing a theoretical background for the AI-based contributions to the development of the AVF. It is organised as follows: Subsection 2.2.1 introduces Artificial Neural Networks (ANNs), covering their structure, the role of activation functions, various training methods, and the different types of ANNs commonly used in practice. Subsection 2.2.2 expands the discussion to data-driven models, presenting them as a broader concept that encompasses ANNs and highlighting their versatility and applications in diverse domains.

2.2.1 Artificial Neural Networks (ANN)

Artificial Neural networks (ANNs) are considered as a special class of data-driven models, that have prompted the development of related concepts and fields of Machine Learning (MLs), Artificial Intelligence (AI), and data science. These models are designed to mimic the human brain behavior when processing information by learning from wide amounts of data, identifying patterns, and producing predictions. A neural network consists of multiple layers of interconnected nodes or "neurons," each of which processes data in a hierarchical manner. The architecture typically includes an input layer, one or more hidden layers, and an output layer, with each neuron performing a mathematical operation, usually a weighted sum followed by a nonlinear activation function, before passing its output to the next layer (See Figure 2.1) and detailes in the following :

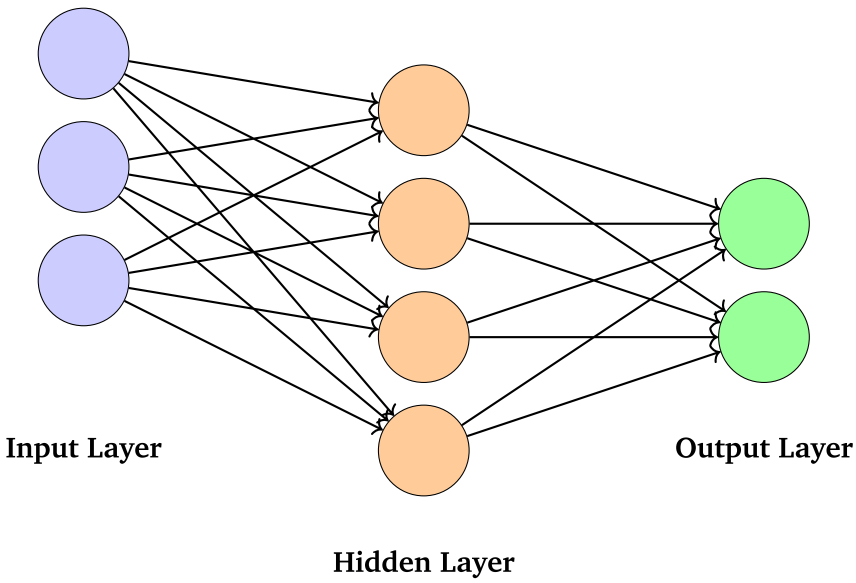


Figure 2.1 – General Architeture Neural Network Architecture

2.2.1.1 Structure on ANN

Since ANNs are computational models inspired by the way biological neural networks in the human brain function, they are composed of several core components interacting to process input data, learn patterns, and make predictions. These components are:

2.2.1.1.a Neurons

Each neuron calculates the weighted total of its inputs and applies an activation function. A neuron's output is determined by:

$$a_j = f\left(\sum_i w_{ij} x_i + b_j\right) \tag{2.1}$$

where:

- x_i are the input values,
- w_{ij} are the weights connecting the i-th input to the j-th neuron,
- b_j is the bias of the j-th neuron,
- f is the activation function,
- a_j is the output of the j-th neuron.

2.2.1.1.b Layers

ANNs are typically organized into multiple layers, with each layer containing several neurons. Layers of neurons allow ANNs to learn increasingly complex representations of the data.

- **Input Layer:** This is the first layer of the network, where the input features' data are injected into the model. The neurons in the input layer simply pass the raw data to the next layer without any computation or transformation.
- **Hidden Layers:** These intermediate layers are where most of the computation occurs. The neurons in hidden layers apply the activation function to their weighted inputs and produce outputs that are passed to the next layer. Hidden layers are key to enabling the network to model complex patterns and non-linear relationships in the data. A network with multiple hidden layers is referred to as a Deep Neural Network (DNN). The output of each neuron in a hidden layer is calculated according to 2.1.
- **Output Layer:** The final layer in the network, where the predictions are made.

2.2.1.1.c Weights and Biases

- **Weights:** w_{ij} are parameters learned during the training process and set to minimize the loss function.
- **Biases:** b_j are additional parameters added to the weighted sum to enable the model to better fit the data.

2.2.1.2 Activation Functions

Activation functions are essential in artificial neural networks because they introduce non-linearity into the network. Without non-linear activation functions, the network would behave as a simple linear model, regardless of how many layers it has. Some common activation functions include:

2.2.1.2.a Sigmoid Function

A sigmoid activation function outputs values between 0 and 1, making it suitable for binary classification problems:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.2}$$

Its derivative is given by:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$$

2.2.1.2.b ReLU (Rectified Linear Unit):

ReLU is one of the most widely used activation functions because it is computationally efficient and helps mitigate the vanishing gradient problem. It outputs the input if it is positive and zero otherwise.

$$\text{ReLU}(x) = \max(0, x) \tag{2.3}$$

Its derivative is given by:

$$\text{ReLU}'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

2.2.1.2.c Tanh (Hyperbolic Tangent):

Tanh is similar to the sigmoid function, but its outputs range between -1 and 1, making it useful when negative outputs are desired.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.4}$$

Its derivative is given by:

$$\tanh'(x) = 1 - \tanh^2(x)$$

2.2.1.2.d Softmax Function:

Softmax is typically used in the output layer for multi-class classification problems. It transforms raw output values into probabilities by normalizing the output values so that they sum to 1.

$$\text{softmax}(z_j) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{2.5}$$

2.2.1.3 Training Methods

The training technique for a neural network involves adjusting its weights and biases to reduce the difference between predicted and actual outputs. This procedure consists of numerous steps:

2.2.1.3.a Forward Propagation

The input data is transmitted across the network during forward propagation, and each neuron uses activation functions, weights, and biases to determine its output.

2.2.1.3.b Loss Function

The difference between expected and actual values is measured by the means of the loss function. Several loss functions are reported in the liiterature, such as:

- Mean Squared Error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.6}$$

  where N is the number of samples, y_i is the actual value, and $\hat{y}_i$ is the predicted value.

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.7}$$

- Binary Cross-Entropy Loss:

$$\text{Binary Cross-Entropy} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{2.8}$$

  where y_i is the actual binary label (0 or 1), and $\hat{y}_i$ is the predicted probability.

- Huber Loss:

$$\text{Huber Loss} = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{for } |y_i - \hat{y}_i| \leq \delta \\ \delta|y_i - \hat{y}_i| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{2.9}$$

  where δ is a threshold parameter that makes the loss less sensitive to outliers.

2.2.1.3.c Backpropagation

Backpropagation is used to compute the gradients of the loss function with respect to the weights. These gradients are used to update the weights and biases of the network using optimization algorithms.

2.2.1.3.d Optimization Algorithms

During training process, the weights and biases are updated through a variety of optimization methods. Typical optimization techniques include:

- Stochastic Gradient Descent (SGD):

$$w_{i,j,k+1} = w_{i,j,k} - \eta\frac{\partial \text{Loss}}{\partial w_{ij}} \tag{2.10}$$

  where η is the learning rate.

- Adam (Adaptive Moment Estimation): The Adam optimization algorithm adjusts the learning rates based on estimates of the first and second moments of the gradients. It is formulated as follows:

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1)\frac{\partial \text{Loss}}{\partial w_{ij}} \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2)\left(\frac{\partial \text{Loss}}{\partial w_{ij}}\right)^2 \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\
w &= w - \eta\frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}
\end{aligned}
\tag{2.11}
$$

where t the iteration number, m_t and v_t are estimates of the first and second moments of the gradients, β_1 and β_2 are decay rates, ϵ is a constant to prevent division by zero, and ”Loss” refers to the loss function.

2.2.1.4 Types of ANNs

- Single-Layer Perceptron: A basic neural network with no hidden layers. It applies a linear transformation followed by an activation function.

$$\hat{y} = f\left(\sum_i w_{ij}x_i + b_j\right) \tag{2.12}$$

- Multi-Layer Perceptron (MLP): A neural network with one or more hidden layers, allowing it to learn complex patterns. Each layer computes its output as follows:

$$a_j^{(l)} = f\left(\sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right) \tag{2.13}$$

  where l denotes the layer index, and $a_i^{(l-1)}$ are the activations from the previous layer.

As presented previously, the structure allows ANNs to model highly nonlinear relationships and perform several tasks such as image and speech recognition, natural language processing, and time-series prediction, which are challenging for traditional models. The training of ANNs involves adjusting the weights of the connections between neurons to minimize the error in the network's predictions, typically using optimization algorithms like gradient descent. However, when it comes to high-dimensional settings, classical backpropagation, the most widely used method for training neural networks, often exhibits slow convergence, which can significantly affect the efficiency and performance of training, especially when dealing with large-scale and complex data [57]. This challenging issue has motivated research into alternative training algorithms aimed at improving the speed and accuracy of convergence. Several strategies have been proposed to address the limitations of traditional backpropagation, such as adaptive descent step methods and second-order derivative approaches [20, 94]. These techniques adjust the step size during training, allowing for a more efficient optimization process able to overcome the slow convergence associated with high-dimensional data. When talking about ANNs, we typically distinguish two different concepts ” shallow” and ”deep” ones. The distinction is based on the computational units used in the hidden layers of these networks include perceptrons, radial units, and kernel units. So in terms of hidden layers, networks with only one hidden layer are often referred to as “shallow” networks, in contrast to “deep” ones that have two or more hidden layers. While deep networks outperform shallow ones in certain tasks [77], the one-hidden-layer paradigm remains widely employed in various applications (see, e.g., [2, 3, 116, 141] and the references therein). The approximation capabilities of such networks have been extensively explored in theoretical studies, demonstrating that, for various types of computational units, they are universal approximators, i.e., they are capable of accurately approximating any continuous function as desired [34, 142]. Neural networks have been integrated in several and nearly all fields. In control engineering, for instance, ANNs have increasingly become a crucial tool to revolutionize the classical control theory, significantly enhancing system performance by addressing challenges such as nonlinearities, uncertainties, and real-time adaptability. Several trends have emerged in the application of ANNs to control systems. One of the prominent approaches is Neural Network-based Model Predictive Control (MPC), which uses NNs to approximate system models and optimize control inputs in real-time by solving an optimization problem over a prediction horizon. ANNs are being applied to improve MPC, especially when the system dynamics are highly nonlinear or difficult to model [90, 109, 135, 137]. Additionally, Neural Network-based Robust Control focuses on handling model uncertainties and external disturbances where neural networks are used to learn and compensate for model inaccuracies or disturbances. Authors in [93] propose the design of ANN is designed to approximate parameter uncertainties within a robust hybrid tracking control strategy for robotic system. Furthermore, Fault Detection and Isolation (FDI), a technique helps detect and isolate faults in complex systems, ensuring continued reliable operation in complex systems. Integrated neural network can be trained to detect patterns in sensor data that indicate faults or anomalies and to isolate the specific source of the fault [74, 114, 159]. ANNs were also adapted to optimal control to approximate optimal control laws to minimize or maximize a certain performance criterion, mainly for complex systems, reducing reliance on explicit system models [138]. One of the famous ANN-based approach, also, is the use of Extended Kalman Filter (EKF)-based learning algorithms. EKF, originally developed for recursive estimation in control systems, has been adapted for neural network training due to its ability to accelerate the convergence of the optimization process. The advantage of EKF-based training lies in its higher convergence rate compared to standard backpropagation, particularly in challenging high-dimensional spaces. However, this advantage often comes at the cost of increased computational complexity, as matrix inversions are required to compute the EKF gains, which can be computationally expensive and memory-intensive [127]. As a result, significant research efforts have focused on overcoming these computational challenges and improving the flexibility and efficiency of EKF-based learning methods. For instance, specialized EKF techniques have been developed to handle one-dimensional problems more efficiently, and multi-stream EKF methods have been proposed to handle large amounts of data in batch mode, allowing for recursive training on complex datasets [57]. Moreover, EKF has not only been applied to feedforward neural networks but also investigated in the context of recurrent neural networks (RNNs), which are particularly well-suited for processing sequential data. This demonstrates the versatility of EKF as a training method, capable of improving convergence in a range of network architectures [88]. Additionally, a connection has been established between EKF and least squares methods, providing further insight into how these advanced techniques can be applied to neural network training, including regularization methods to prevent overfitting [17]. These efforts reflect a broader trend in the field of neural network research, where the focus is not just on the foundational principles of neural networks, such as backpropagation, but also on improving training methods to handle the increasing complexity and size of datasets encountered in real-world applications. Furthermore, these advancements contribute to the broader trend of making machine learning models more accessible and efficient for researchers and practitioners working with vast amounts of data. By improving convergence rates, reducing computational costs, and enhancing flexibility in training, techniques like EKF-based learning algorithms are poised to play a central role in the future of neural network research and their application across diverse domains, ranging from autonomous systems to natural language understanding and beyond.

2.2.2 Data-driven Models : A broader concept of ANNs

Data-driven models are methods that allow learning patterns or generating predictions about a specific process by relying entirely on data, rather than physical models. ANNs are one of the most widely used types of data-driven models, particularly for tasks involving complex data such as images, time series, or spoken language.Over the recent decades, data collection from industrial processes has become more accessible, thanks to the adoption of distributed control systems (DCS) in most industries. Simultaneously, the expansion of database and data mining technologies has offered strong backing for the development and implementation of data-driven modeling techniques in industrial operations, and thus, the expansion of data-driven techniques for process monitoring, control, and system optimization across various industries [42, 51, 67, 166]. Unlike traditional white-box models, which depend on several parameters governed by explicit physical laws in order to describe system behavior, data-driven models use wide datasets to determine system dynamics, enhancing their flexibility and adaptability. The fundamental idea behind data-driven models is that interactions, behaviors, and patterns within the system are captured in the data itself and may be discovered using algorithms and computational tools. Data-driven can be categorized into three main types: (i) supervised learning, (ii) unsupervised learning, and (iii) reinforcement learning. Supervised learning models are trained on labeled datasets and are useful for prediction tasks such as classification or regression [48]. Unsupervised learning models, on the other hand, focus on finding hidden patterns within data without the use of labels, often used for clustering or dimensionality reduction [33]. Reinforcement learning, which optimizes decisions through feedback from the environment, has become increasingly important in areas like autonomous control and robotics [100]. Through mining and analyzing information from process data, various types of data-based models can be constructed for industrial applications, such as process monitoring, mode clustering, and soft sensing of key performance indices (KPIs) [51, 68, 99, 104, 106, 120, 144, 163, 172] . For example, by projecting high-dimensional process data into a lower-dimensional space, latent variable models such as Principal Component Analysis (PCA) and Partial Least Squares (PLS) can extract key information from the process. These methods form the foundation for developing efficient monitoring systems that can detect abnormal behavior and optimize performance [68, 144]. In addition, data-based regression models have been developed to estimate or predict hard-to-measure variables, often referred to as "soft sensing." Techniques such as Least Squares Regression, Artificial Neural Networks (ANN), Support Vector Regression (SVR), and Gaussian Process Regression (GPR) are commonly used for this purpose [99]. By building effective relationships between easily measurable process variables and those that are difficult to measure, these models enable real-time estimation or prediction, offering substantial time, cost, and labor savings in industrial settings. Data-driven modeling techniques can be further categorized into parametric and non-parametric machine learning algorithms. Parametric algorithms simplify the unknown function to be learned by assuming it follows a known form, characterized by a fixed set of parameters that do not change with the training examples. These algorithms do not require assumptions about the underlying probability distribution of the data, which makes them applicable regardless of the data distribution. Typically, parametric methods involve two steps: selecting the form of the functional relationship and learning the parameters from the training data. Examples include logistic regression, linear discriminant analysis, and naïve Bayes (see, e.g., Chapter 2 of [95] and Chapter 18 of [147]). These methods are fast to train and easy to interpret. However, the choice of a predetermined functional form may limit their ability to accurately represent complex data patterns, making them more suitable for simple, low-dimensional problems. In contrast, non-parametric machine learning models do not rely on strict assumptions about the underlying data mapping and tend to perform better when there is a large amount of data and minimal prior knowledge. These models can fit a broader range of functional forms. However, they generally require more training time and can be harder to interpret. Examples include k-nearest neighbors regression, decision trees, deep Gaussian processes,

and complex neural networks, which are highly nonlinear.
Chapter 3 deploys the black-box capabilities of neural networks to construct a crop growth model for AVFs, based real-world data to overcome the limitations of traditional dynamic growth models.

2.3 Brief Overview on Nonlinear Observer Design

Dynamical systems are at the heart of many fields, including engineering, biology, and economics, where understanding the evolution of of systems over time is essential. However, in real-world applications, it is often impractical or too expensive to measure all the internal states of a system directly. This limitation creates a need for state estimation, a process that infers unmeasured states using available outputs and the known dynamics of the system. A key concept in state estimation is observability, which determines whether the internal states of a system can be fully reconstructed from its outputs.
State estimation is crucial in applications like autonomous vehicles, robotics, and control systems. It enables tasks such as system monitoring, fault detection, and synchronization in multi-agent systems. Common estimation methods include the sliding mode observer [156], the high-gain observer approach [8, 11, 21, 24, 60], sliding mode observer [41, 171], the LMI-based observer techniques [10, 76, 107, 112, 176, 179], the unknown input observer [113], the interval observer [170]. Estimators like the EKF and MHE incorporate statistical or optimization methods to handle noise and uncertainties, making them suitable for systems with significant disturbances. Among these methods, we mention the extended Kalman filter [103, 115, 150, 155] and moving-horizon estimators [14–16, 19, 23].
This section provides an overview of fundamental topics, contributing later to the development of the AVF: stability and state estimation in dynamical systems. Subsection 2.3.1 focuses on stability analysis, introducing key methods such as Lyapunov's techniques and Input-to-State Stability (ISS), which are critical for ensuring robustness in system behavior. Subsection 2.3.2 explores state estimation techniques, covering observability, observer design, and advanced methods like LMI-based observers. Together, these concepts serve as the foundation for developing reliable estimation schemes.

2.3.1 The Stability Concept for Dynamic Nonlinear Systems

Determining whether a system will revert to its equilibrium state or deviate from it in response to minor perturbations is crucial for real-world applications: this idea is the core of the stability concept. In this section, we first introduce the main definitions and principles of stability, in particular Lyapunov's methods, which provide powerful tools for analyzing the stability of nonlinear systems. Let us consider the following autonomous system described by the following differential equation:

$$\dot{x} = f(x), \tag{2.14}$$

where $x \in D \subset \mathbb{R}^n$, and D is a domain containing the origin. We assume that the function f governing the system's dynamics is piecewise continuous for time t and locally Lipschitz with respect to the state x. This implies that there exists a constant $k > 0$ such that:

$$\|f(x_1) - f(x_2)\| \leq k\|x_1 - x_2\|, \tag{2.15}$$

for all $x_1, x_2 \in D$. Additionally, we assume that $f(x_0)$ is bounded, where x_0 represents the initial condition at $t = 0$.

Definition 2.3.1 ([91, 149, 168]). *An equilibrium point x^* of the system (2.14) is a state where the system remains at rest if it starts from that state. Mathematically, this is expressed as:*

$$x(0) = x^* \Rightarrow x(t) = x^* \quad \forall t \geq 0.$$

Equivalently, x^* is an equilibrium point if and only if:

$$f(x^*) = 0, \quad \forall t \geq 0.$$

Without loss of generality, we will focus on the stability of the origin ($x = 0$) for the autonomous system (2.14). This simplifies the analysis while still capturing the essential behavior of the system.

2.3.1.1 Types of Stability

Stability can be classified into various types based on the system's response to perturbations from its equilibrium state. These classifications include, mainly, stability, asymptotic stability, and exponential stability.

Definition 2.3.2 ([105]). *The origin is said to exhibit different types of stability based on the following criteria:*

1. ***Stability Criteria:*** *The origin is a stable equilibrium point if, for every $\epsilon > 0$, there exists a $\delta = \delta(\epsilon)$ such that:*

   $$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \epsilon, \quad \forall t \geq 0.$$

   If this condition is not met, the origin is considered unstable.

2. ***Asymptotic Stability:*** *The origin is asymptotically stable if it is stable, and in addition, δ can be chosen such that:*

   $$\|x(0)\| < \delta \Rightarrow \lim_{t \to \infty} \|x(t)\| = 0.$$

   This means that solutions not only remain close to the origin but also converge to the origin as time goes to infinity.

3. ***Exponential Stability:*** *The origin is locally exponentially stable if there exist constants $\alpha \geq 1$ and $\beta > 0$ such that:*

   $$\|x(t)\| \leq \alpha \|x(0)\| \exp(-\beta t), \quad \forall t \geq 0, \ \forall x(0) \in \mathcal{B}_r.$$

   If $\mathcal{B}_r = \mathbb{R}^n$, we say that the origin is globally exponentially stable. This indicates that solutions decay to the origin at an exponential rate.

In conclusion, an equilibrium point is considered stable if small perturbations do not lead to significant departures from the point. It is asymptotically stable if these deviations remain small and gradually diminish to zero over time. Exponential stability represents a more stringent form of asymptotic stability, where the convergence rate is exponential. The reported definitions underscore the system's behavior in response to disturbances. However, directly applying these fundamental definitions to establish stability typically requires finding explicit solutions to the system, which can be difficult or infeasible. To address this challenge, alternative approaches, such as Lyapunov's methods, are used to analyze stability witout the need of applying explicit solutions.

2.3.1.2 Lyapunov's Methods for Stability Analysis

Lyapunov's methods offer effective techniques for assessing the stability of equilibrium points without the need for explicit solutions to the system's differential equations. The two primary approaches are the indirect method of Lyapunov and the direct method of Lyapunov.

2.3.1.2.a Lyapunov's Indirect Method

Theorem 2.3.3. *([85, 149]) Let $x = 0$ be an equilibrium point for the nonlinear system (2.14), where $f : D \to \mathbb{R}^n$ is continuously differentiable and D is a neighborhood of the origin. The linearized system around the origin is given by:*

$$\dot{x} = f(x) \Rightarrow \dot{x} = Ax,$$

where $A = \left.\frac{\partial f}{\partial x}\right|_{x=0}$ is the Jacobian matrix evaluated at the origin.
The stability of the origin can be classified as follows:

1. *The origin is locally asymptotically stable if all eigenvalues of A have negative real parts $Re(\lambda_i) < 0$.*

2. *The origin is unstable if at least one eigenvalue of A has a positive real part $Re(\lambda_i) > 0$.*

3. *If some eigenvalues have zero real parts $Re(\lambda_i) = 0$ and the rest have non-positive real parts, the system's behavior is more complex:*

   - *If $Re(\lambda_i) \leq 0$ for all i, the origin may be marginally stable.*
   - *Otherwise, the origin is unstable.*

In numerous scenarios, this theorem is a useful tool for stability analysis since it provides a criterion for local stability based on the linearized system. However, when the linear approximation is not sufficient to determine stability, Lyapunov's direct method can be applied.

2.3.1.2.b Lyapunov's Direct Method

The direct method of Lyapunov relies on the principle of energy dissipation. In physical systems, if the total energy of the system steadily decreases over time, an equilibrium state will be reached. This concept can be mathematically formulated using a scalar function $V(x)$, which is commonly interpreted as the total energy of the system.

Theorem 2.3.4 (Lyapunov Function and Lyapunov Stability [105])**.** *Let $x = 0$ be an equilibrium point for the nonlinear system (2.16), and let D be a neighborhood of the origin. Consider a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ satisfying the following conditions:*

- *$V(x) = 0$ if and only if $x = 0$.*

- *$V(x) > 0$ if $x \neq 0$ (i.e., $V(x)$ is positive definite).*

- *The time derivative of V along the trajectories of the system is non-positive:*

$$\dot{V}(x(t)) = \frac{d}{dt}V(x(t)) = \nabla V(x) \cdot f(x) \leq 0, \quad \forall x \neq 0.$$

Then $V(x)$ is called a Lyapunov function, and the system is stable.

- *The system is asymptotically stable if $\dot{V}(x) < 0$ for all $x \neq 0$.*

- *The system is globally asymptotically stable if, in addition, $V(x) \to +\infty$ as $\|x\| \to +\infty$ (also known as "radial unboundedness").*

Lyapunov's direct method provides a powerful way to prove stability without solving the differential equations explicitly. By constructing an appropriate Lyapunov function, one can conclude the stability of the system's equilibrium. However, it is worth noting that we cannot conclude about the instability of a system since we have only sufficient conditions, except in special cases such as LTI systems.

2.3.1.3 Input-to-State Stability (ISS) and its Extensions

In real-world applications, control systems are usually subject to noise, disturbances in control inputs and errors in measurements or estimates. Therefore, it is desirable for a system to be not only stable but also input-to-state stable (ISS): robust with respect to external disturbances and uncertainties. Hence, the concept of ISS extends the classical notion of stability to situations where the system is not only subjected to initial state deviations but also to external inputs. ISS was first introduced by Sontag in the 20^{-th} in his work [151]. The mentionned research work on the development of ISS provided a framework for analyzing systems where the state remains bounded in response to bounded inputs. This concept has since been extended and applied to a variety of dynamical systems, becoming a fundamental tool in control theory for assessing stability under non-negligible disturbances.
Let's consider the following differential equation system:

$$\dot{x} = f(x, u), \tag{2.16}$$

where $x \in \mathbb{R}^n$, where $u : \mathbb{R}_+ \to \mathbb{R}^m$ is a Lebesgue measurable essentially bounded external input and f is a Lipschitz continuous function. This ensures that there exists a unique absolutely continuous solution of the system. with input $u : [0, \infty) \to \mathbb{R}^n$ (also called "controls" or "disturbances" depending on the context) being measurable locally essentially bounded maps.
ISS extends the global asymptotically stability (GAS) property to nonlinear systems. Intuitively, ISS means that the system's behavior remains bounded when the inputs are bounded and tends towards an equilibrium point when the inputs approach zero [153]. ISS is closely related to the concept of stability under perturbations (total stability), which is well-studied in the classical dynamical systems literature. ISS requires that the system remains GAS up to an error term dependent on the magnitude of the input u, measured via the essential supremum norm:

$$\|u\|_\infty := \operatorname*{ess\,sup}_{t \geq 0} \|u(t)\|. \tag{2.17}$$

Definition 2.3.5 ([154]). *System is called ISS, if there exist $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that for all initial values x_0, all perturbation functions u and all times $t \geq 0$, the following inequality holds:*

$$|x(t, x_0; u)| \leq \beta(|x_0|, t) + \gamma(\|u\|_\infty, t) \tag{2.18}$$

ISS can be characterized via ISS Lyapunov functions. The ISS Lyapunov function provides a powerful tool for analyzing the stability of nonlinear systems subject to input disturbances. It guarantees that the state trajectories of the system remain bounded, provided that the inputs are also bounded, and the Lyapunov gain ensures that the inputs do not cause the system state to grow faster than a certain rate.

Theorem 2.3.6 ([153]). *A system is ISS if and only if it admits an ISS Lyapunov function, i.e. a smooth function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and $\mathcal{K}_\infty$-functions α_1, α_2 and $\mathcal{K}$- functions α_3, γ such that*

$$\alpha_1|x| \leq V(x) \leq \alpha_2|x| \tag{2.19}$$

and

$$\dot{V}(x) \leq -\alpha_3|x| \tag{2.20}$$

hold for all $x \in \mathbb{R}^n, u \in \mathbb{R}^m$ so that $|x| \geq \gamma(|u|)$; the function γ is called Lyapunov gain.

Remark 2.3.7. *A smooth function V is an ISS-Lyapunov function for (2.16) if and only if there exist $\alpha_i, i = 1, .., 4$ such that (2.20) holds, and*

$$\dot{V}(x) \leq -\alpha_3|x| + \alpha_4|u| \tag{2.21}$$

The ISS property of an observer ensures that its estimation error is bounded and decays over time, despite the presence of input and measurement disturbances. The observer is said to be ISS if the error between the estimated state and the true state of the system remains bounded in the presence of input and measurement disturbances. This is captured by equation (2.22), which states that the error $|x(t) - \hat{x}(t)|$ between the true state $x(t)$ and the estimated state $\hat{x}(t)$ is bounded by a function of the error and time, $\beta(|x(t) - \hat{x}(t)|, t)$, as well as the supremum of the magnitudes of the input disturbance $v(s)$ and the measurement disturbance $w(s)$ up to time t.

Definition 2.3.8 (ISS-observer [152]). *An observer is said to be Input-to-State Stable (ISS) if the estimation error satisfies*

$$|x(t) - \hat{x}(t)| \leq \beta\left(|x(0) - \hat{x}(0)|, t\right) + \gamma_1\left(\sup_{s\in[0,t)} |v(s)|\right) + \gamma_2\left(\sup_{s\in[0,t)} |w(s)|\right) \tag{2.22}$$

for all $t \geq 0$, for some $\beta \in \mathcal{KL}$, and $\gamma_1, \gamma_2 \in \mathcal{K}$.

2.3.2 Estimation in Dynamical Systems

In dynamical systems, estimation plays a crucial role in determining the internal state of a system based on observable outputs. Accurate state estimation is necessary for the control and optimization of dynamic processes, especially when some states are unmeasured or unavailable. While the terms "observer" and "estimator" are often used interchangeably, there is a subtle distinction in their use. An observer specifically refers to a system designed to estimate the unmeasured internal states of a system based on the system dynamics and available output measurements. On the other hand, a state estimator refers more broadly to any algorithm used to estimate the state of a system, which can include observers as well as other estimation techniques like the Kalman filter.
This section explores the concepts and techniques used in state estimation, including the principle of observability, the design of the most-used state observers, and advanced methods such as LMI-based observers.

2.3.2.1 Observability

The observability of a process is a fundamental concept in the field of state estimation. Indeed, to reconstruct the inaccessible states of a system, it is necessary to know, a priori whether the variables are observable or not. Observability stands for the possibility of reconstructing the full trajectory from the observed data, that is, from the output trajectory in the uncontrolled case, or from the couple (output trajectory, control trajectory) in the controlled case. This property is fundamental for both linear and nonlinear systems and dictates whether an estimator or observer can function effectively.

2.3.2.1.a Observability of Nonlinear Systems

Let us consider the class of nonlinear systems described by :

$$\begin{cases} \dot{x} &= f(x, u), \\ y &= h(x, u), \end{cases} \tag{2.23}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$.

Definition 2.3.9. *Observability of nonlinear systems (Geometric Condition) [32]*

- *Indistinguishability: Let $y_u^0(t) \to x^0$ and $y_u^1(t) \to x^1$. Then x^0 and x^1 are indistinguishable if $y_u^0(t) = y_u^1(t),\ \forall\, t \geq 0,\ \forall\, u$. Otherwise, x^0 and x^1 are distinguishable.*

- *The system (2.23) is observable in x^0 if x^0 is distinguishable from any $x \in \mathbb{R}^n$. In addition, the system (2.23) is observable if $\forall\, x^0 \in \mathbb{R}^n$, x^0 is distinguishable.*

Definition 2.3.10 (Observability of nonlinear systems (rank condition))**.** *We say that system (2.23) is observable if the following rank condition holds:*

$$\operatorname{rank}\left(dh,\, dL_f h,\, \ldots,\, dL_f^{n-1} h\right)^{\top} = n$$

where $dL_f^k h$ is given by:

$$dL_f^k h = \left(\frac{\partial L_f^k h}{\partial x_1}, \frac{\partial L_f^k h}{\partial x_2}, \ldots, \frac{\partial L_f^k h}{\partial x_n}\right)$$

and $L_f^k h$ denotes the k-th Lie derivative of h along the vector field f, recursively defined by:

$$L_f^0 h := h, \quad L_f^k h := L_f\left(L_f^{k-1} h\right), \quad \text{for } k \geq 1.$$

The (first) Lie derivative of a scalar function $h : D \to \mathbb{R}$ along a vector field $f : D \to \mathbb{R}^n$ is defined as:

$$L_f h(x) = \sum_{i=1}^{n} f_i(x) \frac{\partial h}{\partial x_i}(x).$$

For more details about the observability of nonlinear systems, one can refer to [32, 84, 91, 129, 149].

2.3.2.1.b Observability of Linear Systems

Now, let's consider the observability of linear systems, which are easier to analyze due to their linearity. Linear systems are described by the following equations:

$$\begin{cases} \dot{x} &= Ax + Bu, \\ y &= Cx, \end{cases} \tag{2.24}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control and $y \in \mathbb{R}^p$ is the measured output. The matrices A, B, C are constant matrices of appropriate dimensions. The Kalman observability criterion provides a straightforward test for the observability of linear systems. If the system is observable, then it is possible to reconstruct the entire state vector from the input and the output data.

Definition 2.3.11 (Kalman observability criterion [80])**.** *the system (2.24) is said to be observable if and only if*

$$\operatorname{rank}\Big(\mathcal{O}(A, C)\Big) \triangleq \operatorname{rank}\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} = n \tag{2.25}$$

$\mathcal{O}(A, C)$ is called Kalman observability matrix (of size $np \times n$).

2.3.2.2 State Estimators

State estimators are fundamental tools in control theory, robotics, signal processing, and other fields that involve dynamic systems, as they allow for accurate state reconstruction in the presence of noise, uncertainty, and incomplete measurements.

2.3.2.2.a Standard Kalman Filter

we start, first, by introducing the standard Kalman filter in the case of of continuous and discrete Linear Time varying (LTV) systems. Actually, being a fundamental design for EKF, the standard offers a simple yet optimal solution for estimating the state of linear systems in the presence of stochastic disturbances [73, 103, 122]. Based on a linear state model that relates the evolution of the true signal to the noisy or incomplete measurements received over time. Essentially, Kalman filtering provides a means to estimate unknown variables by processing noisy measurements over successive time steps. The Kalman filtering algorithm depicted by Figure2.2 provides estimates of unknown variables given the measurements observed over time.

- **Case of continuous Linear Time varying (LTV) systems:** For the LTV system of the form

$$\begin{cases} \dot{x} &= Ax + Bu + v(t), \\ y &= Cx + w(t), \end{cases} \tag{2.26}$$

where $v(t)$ and $w(t)$ are two Gaussian white noises of zero expectation, and non-correlated covariances Q and R, respectively.

$$\dot{\hat{x}} = A\hat{x} + Bu + PC^T R^{-1}(y - C\hat{x}), \tag{2.27}$$

$$\sum_{i=1}^{n} E(e_i(t)^2) \quad = \quad E(e^T(t)e(t))$$

$$\dot{\hat{x}} = A(t)\hat{x} + B(t)u + PC^T(t)R^{-1}(y - C(t)\hat{x}), \tag{2.28}$$

where P is a symmetric and definite positive solution of the following Riccati equation:

$$\dot{P} = AP + PA^T + Q - PC^T R^{-1} CP.$$

For the LTV system of the form

$$\begin{cases} \dot{x}_{x+1} &= A_t x_t + B_t u_t + v_t, \\ y_t &= C_t x_t + w_t, \end{cases} \tag{2.29}$$

where v and w are two Gaussian white noises of zero expectation, and non-correlated covariances Q and R, respectively. Consider the best possible predictor model and estimating that the value that the noise will take at the following instant will be zero, the best prediction that we can make is given by

$$\hat{x}_{t+1/t} = A\hat{x}_{t/t} + Bu_t.$$

As this prediction must be the same as:

$$\hat{x}_{t+1/t} = A\hat{x}_{t/t-1} + Bu_t + K_t(y_t - C\hat{x}_{t/t-1}).$$

We conclude that

$$\hat{x}_{t/t} = \hat{x}_{t/t-1} + A^{-1}K_t(y_t - C\hat{x}_{t/t-1})$$

- **Case of Discrete Linear Time varying (LTV) systems:** For discrete-time LTV system of the form :

$$\begin{cases} x_{k+1} &= A_k x_k + B_k u_k + v_k, \\ y_k &= C_k x_k + w_k, \end{cases} \tag{2.30}$$

where x_k is the state vector at time step k, u_k is the control input at time step k, y_k is the measurement output at time step k, v_k represents the process noise, w_k represents the measurement noise assumed to be zero-mean Gaussian with covariances Q_k, R_k, respectively.

The prediction for the state and the error covariance is given by:

$$\hat{x}_{k+1/k} = A_k\hat{x}_k + B_k u_k \tag{2.31}$$

$$P_{k+1/k} = A_k P_k A_k^T + Q_k \tag{2.32}$$

where $\hat{x}_{k+1/k}$ is the predicted state at time step $k+1$, $P_{k+1/k}$ is the predicted state covariance at time step $k+1$, A_k is the state transition matrix at time step k, B_k is the control input matrix at time step k, and Q_k is the process noise covariance at time step k.

The state estimate is updated based on the measurement y_k at time step k, and the Kalman gain K_k is computed as follows:

$$K_k = P_{k+1/k}C_k^T\left(C_k P_{k+1/k} C_k^T + R_k\right)^{-1} \tag{2.33}$$

such as the updated state estimate is:

$$\hat{x}_{k+1} = \hat{x}_{k+1/k} + K_k(y_k - C_k\hat{x}_{k+1/k}) \tag{2.34}$$

and the updated error covariance is computed as:

$$P_{k+1} = \left(P_{k+1/k}^{-1} + C_k^T R_k^{-1} C_k\right)^{-1} \tag{2.35}$$

2.3.2.2.b Extended Kalman Filter (EKF)

The Extended Kalman Filter (EKF), one of the most interesting applied observers. It is an extension of the standard Kalman filter designed to handle nonlinear systems. The algorithm depicted in Figure 2.3, is based on the idea of replacing the state and output matrices A, C of the linear system (2.26) or (2.29) by the Jacobians of the system non-linearities in question.

- **Case of continuous Linear Time varying (LTV) systems:**

  Consider the following nonlinear system:

$$\begin{cases} \dot{x} & = f(x,u) + v(t), \\ y & = h(x,u) + w(t), \end{cases} \tag{2.36}$$

  The EKF is expressed as follows:

$$\dot{x} = f(\hat{x},u) + PH(\hat{x},u)R^{-1}(y - h(\hat{x},u)) \tag{2.37a}$$

$$\dot{P} = F(\hat{x},u)P + PF(\hat{x},u)^\top + Q - PH(\hat{x},u)^\top R^{-1} H(\hat{x},u)P, \tag{2.37b}$$

  where

$$F(\hat{x},u) = \frac{\partial f}{\partial x}(\hat{x},u),$$

  and

$$H(\hat{x},u) = \frac{\partial f}{\partial x}(\hat{x},u).$$

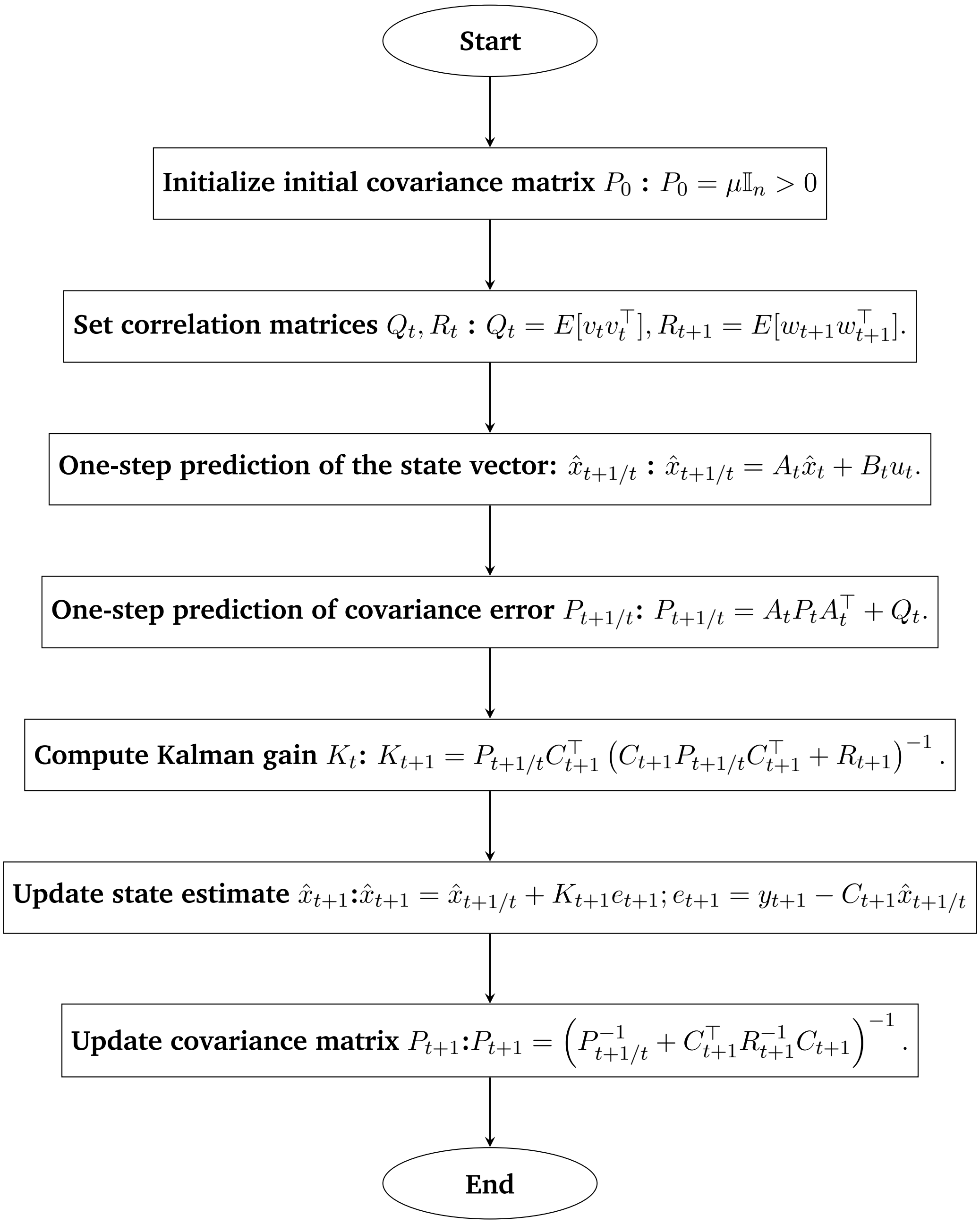


Figure 2.2 – Algorithm of the Standard Kalman Filter Algorithm

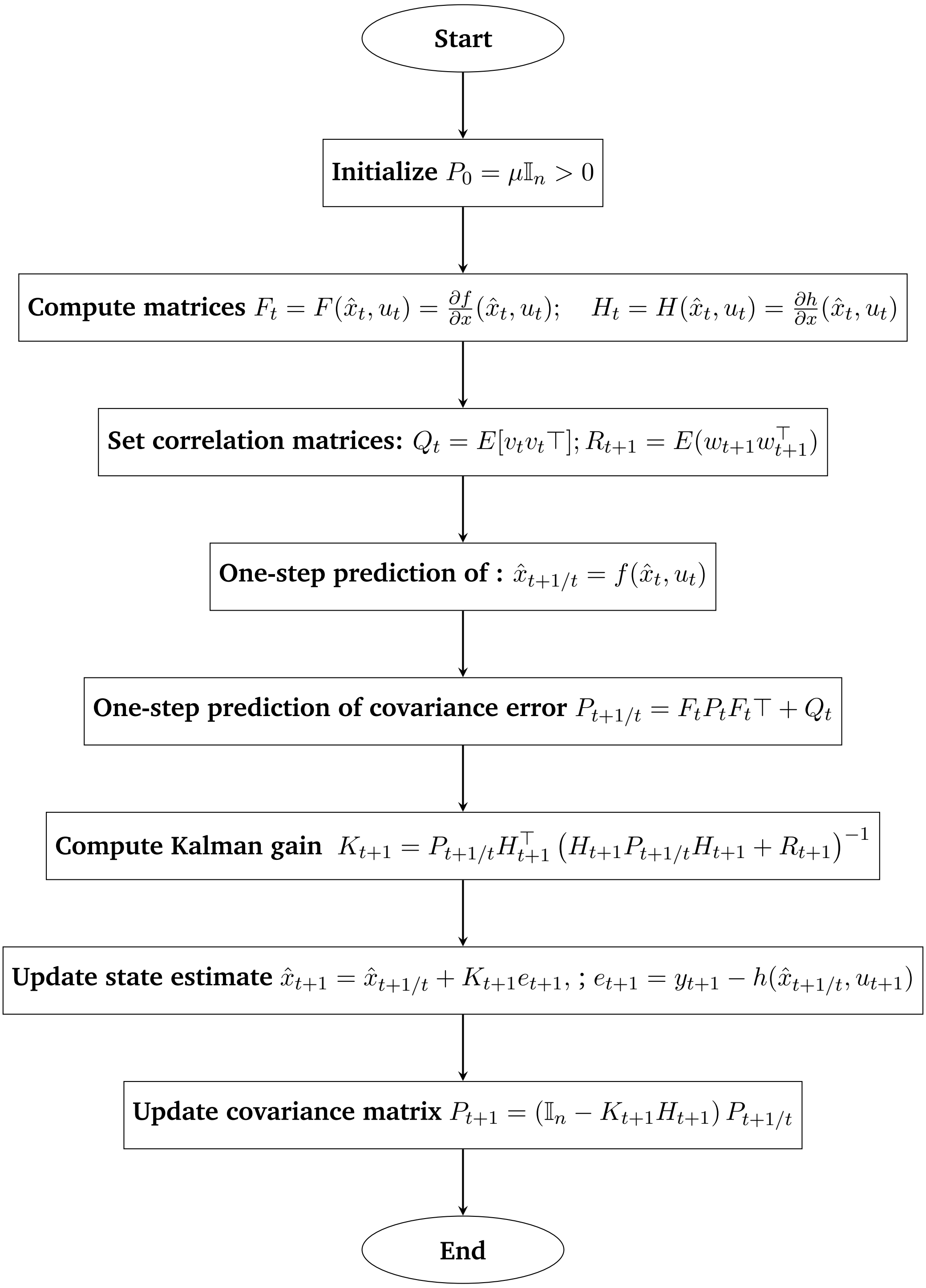


Figure 2.3 – Algorithm for the Extended Kalman Filter

- **Case of discrete Linear Time varying (LTV) systems:**

  Consider the following nonlinear system in discrete time:

$$\begin{cases} x_{k+1} &= f(x_k, u_k) + v_k, \\ y_k &= h(x_k, u_k) + w_k, \end{cases} \tag{2.38}$$

  where v_k and w_k represent the process and measurement noise, modeled as zero-mean Gaussian white noise sequences with covariances Q_k and R_k, respectively.

  The EKF is implemented as follows:

$$\hat{x}_{k+1/k} = f(\hat{x}_k, u_k) + K_k(y_k - h(\hat{x}_k, u_k)), \tag{2.39}$$

  where $\hat{x}_{k+1/k}$ is the predicted state at time step $k+1$, and K_k is the Kalman gain. The covariance of the estimation error evolves according to the Riccati equation:

$$P_{k+1/k} = F(\hat{x}_k, u_k) P_k F(\hat{x}_k, u_k)^T + Q_k, \tag{2.40}$$

  where P_k is the covariance matrix of the estimation error at time k, and $F(\hat{x}_k, u_k) = \frac{\partial f}{\partial x}(\hat{x}_k, u_k)$ is the Jacobian matrix of the system dynamics with respect to the state.

2.3.2.3 State Observers

State observers are a specific type of state estimator designed to estimate the unmeasured internal states of a system. In other words, state observer is a system that provides an estimate of the internal state of a given real system, from measurements of the input and output of the real system. It is typically computer-implemented. A state observer is a typically computer-implemented system that estimates the internal state of a given real system, from measurements of the input and output of the real system. Observers are needed since the full state can not be measured or is too expensive to measure in many applications. In addition, some variables in many applications have to be estimated and can not be measured due to the unavailability of sensors at any cost.

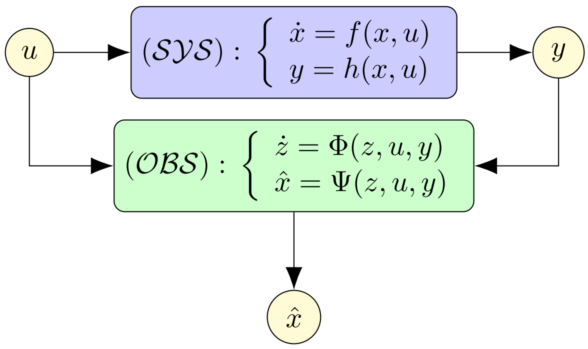


Figure 2.4 – State observer principle

Definition 2.3.12 (State Observer [108], [174]). *Consider the following dynamical system:*

$$(\mathcal{SYS}) : \begin{cases} \dot{x} = f(x,u), \\ y = h(x,u) \end{cases} \tag{2.41}$$

The dynamical system

$$(\mathcal{OBS}) : \begin{cases} \dot{z} = \Phi(z,u,y), \\ \hat{x} = \Psi(z,u,y) \end{cases} \tag{2.42}$$

is said to be a local asymptotic observer for the system (2.41) if the following conditions hold:

- *If the initial states are equal, i.e., $x(0) = \hat{x}(0)$, then the observer perfectly reproduces the state trajectory: $x(t) = \hat{x}(t)$ for all $t \geq 0$.*
- *There exists a subset $\Omega \subseteq \mathbb{R}^n$ such that for any initial estimation error $x(0) - \hat{x}(0) \in \Omega$, the estimation error asymptotically vanishes:*

$$\lim_{t \to +\infty} \|x(t) - \hat{x}(t)\| = 0.$$

Some particular cases include:

- *If the estimation error $\|x(t) - \hat{x}(t)\|$ converges to zero exponentially, the observer is said to be an* ***exponential observer****.*
- *If the convergence property holds for all possible initial conditions, i.e., $\Omega = \mathbb{R}^n$, then the observer is said to be a* ***global observer****.*

2.3.2.3.a Luenberger Observer

The Luenberger observer is a classical and widely used technique for state estimation in linear systems [101, 102, 117, 130]. It estimates the internal states of a system by combining a model of the plant with a correction term based on output feedback. This correction is performed using a feedback gain that adjusts the state estimates according to the difference between the actual system output and the predicted output. While the Luenberger observer provides a simple and effective framework for state estimation, it is not inherently optimal. However, under stochastic assumptions âĂî such as the presence of Gaussian white noise âĂî and when the observer gain is chosen to minimize the mean square estimation error, the Luenberger observer leads to the Kalman filter, which is optimal in the least-squares sense. We consider the dynamic model of a linear system defined as follows:

$$\begin{cases} \dot{x} &= Ax(t) + Bu(t), \\ y &= Cx(t), \end{cases} \tag{2.43}$$

where, at time t, $x(t)$ is the plant's state; $u(t)$ is its inputs; and $y(t)$ is its outputs. Luenberger's theory of observation is essentially based on pole placement techniques. The Luenberger observer for the system (2.43):

$$\dot{\hat{x}} = A\hat{x} + Bu(t) + K(y(t) - C\hat{x}).$$

The dynamics of the estimation error $e(t) = x(t) - \hat{x}$ has the expression

$$\dot{e}(t) = (A - KC)e(t).$$

Using a pole placement technique, it suffices to choose the gain K of the observer so that the eigenvalues of the matrix $A - KC$ are in the left complex half-plane.

2.3.2.3.b Standard High-Gain Observer

The high-gain observer is based on the concept of applying a high gain to the error between the observed output and the actual output. This approach forces the observer to converge quickly to the true state values, even if the initial estimation error is large. We recall the basic high gain observer as in [62]. Consider the class of nonlinear systems described by the equations:

$$\begin{cases} \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} x_2 + f_1(x_1) \\ x_3 + f_2(x_1, x_2) \\ \vdots \\ x_n + f_{n-1}(x_1, \dots, x_{n-1}) \\ f_n(x) \end{bmatrix} \\ y = x_1 \end{cases} \tag{2.44}$$

where $x(t) \in \mathbb{R}^n$ is the system state and $y(t) \in \mathbb{R}^p$ is the output measurement vector. For the sake of brevity, we consider the system (2.44) without control input. We assume that the functions $f_i : \mathbb{R}^i \longrightarrow \mathbb{R}, i = 1, \ldots, n$ are respectively γ_{f_i}–Lipschitz with respect to their arguments, and the Lipschitz constraint is assumed to be global. That is there exists $\gamma_{f_i} > 0, i = 1, \ldots, n$ such that

$$\Big|f_i(x_1 + \Delta_1, \ldots, x_i + \Delta_i) - f_i(x_1, \ldots, x_i)\Big| \leq \gamma_{f_i} \sum_{j=1}^{i} |\Delta_j| \, , \forall \Delta_i \in \mathbb{R} \, . \tag{2.45}$$

If the Lipschitz assumption is not global and the system (2.44) admits a positively invariant compact set on which f is Lipschitz, then we can apply the *Hilbert* projection theorem [123, 178] or the *Kirszbraun–Valentine* extension theorem [162, 177] to extend f to a globally Lipschitz function. The reader can also find details on this extension in [63]. To simplify the developments, we write system (2.44) under the form:

$$\begin{cases} \dot{x} = Ax + f(x) \\ y = Cx \end{cases} \tag{2.46}$$

where

$$C = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}, \; f(x) = \begin{bmatrix} f_1(x_1) & \ldots & f_n(x) \end{bmatrix}^\top \tag{2.47}$$

and the state matrix A is defined by

$$(A)_{i,j} = \begin{cases} 1 & \text{if} \quad j = i + 1 \\ 0 & \text{if} \quad j \neq i + 1 \end{cases} . \tag{2.48}$$

Let us introduce the following linear transformation

$$\zeta = \mathbb{T}_\tau x, \text{ where } \mathbb{T}_\tau \triangleq \operatorname{diag}\left(\frac{1}{\tau}, \ldots, \frac{1}{\tau^n}\right) \tag{2.49}$$

which transforms (2.44) into

$$\dot{\zeta} = \tau A \zeta + \mathbb{T}_\tau f(\mathbb{T}_{\frac{1}{\tau}} \zeta) \tag{2.50}$$

where the relation $\mathbb{T}_\tau^{-1} = \mathbb{T}_{\frac{1}{\tau}}$ is used. Note that $\tau \geq 1$ is a tuning parameter that adjusts the observer gain, and is typically chosen to be large in order to ensure fast convergence of the estimation error. Now consider the following observer corresponding to (2.50):

$$\dot{\hat{\zeta}} = \tau A \hat{\zeta} + \mathbb{T}_\tau f(\mathbb{T}_{\frac{1}{\tau}} \hat{\zeta}) + L\left(y - C\mathbb{T}_{\frac{1}{\tau}} \hat{\zeta}\right) \tag{2.51}$$

where L, independent from τ, is the constant observer gain to be determined. Then the dynamics of the estimation error $e_\zeta = \zeta - \hat{\zeta}$ is expressed as

$$\dot{e}_\zeta = \tau\Big(A - LC\Big) e_\zeta + \Delta f \tag{2.52}$$

where

$$\Delta f \triangleq \mathbb{T}_\tau \Big[f(\mathbb{T}_{\frac{1}{\tau}} \zeta) - f\big(\mathbb{T}_{\frac{1}{\tau}} \hat{\zeta}\big)\Big] . \tag{2.53}$$

From the Lipschitz condition (2.45), the equivalence of norms in $\mathbb{R}^n$, the structure of $\mathbb{T}_\tau$ in (2.49), and the fact that $\tau \geq 1$, we can show as in [18] that there is a constant $k_f > 0$, independent of τ, such that

$$\|\Delta f\| \leq k_f \|e_\zeta\|. \tag{2.54}$$

Now, we can recall the following standard theorem from high-gain methodology.

Theorem 2.3.13 ([18]). *Let $P = P^\top > 0$ and $\mathcal{X}$ be matrices of appropriate dimensions, and $\tau > 0$ is a scalar, such that the following conditions hold:*

$$A^T P + PA - C^\top \mathcal{X} - \mathcal{X}^\top C + \lambda \mathbb{I}_n < 0, \tag{2.55}$$

$$\tau > \max\left(1, \frac{2k_f \lambda_{\max}(P)}{\lambda}\right). \tag{2.56}$$

Then the observer (2.51) corresponding to (2.50), with $L = \mathcal{P}^{-1}\mathcal{X}^\top$, converges exponentially towards zero. Moreover, the estimated state $\hat{x}(t) = \mathbb{T}_{\frac{1}{\tau}}\hat{\zeta}(t)$ exponentially converges to the state $x(t)$ of the original system (2.44).

The key advantage of the high-gain approach is that it guarantees the existence of an exponentially convergent observer, achieved by adjusting a single parameter, which must be sufficiently large [61, 64]. However, the necessity for a large gain poses a significant drawback. Specifically, high-gain observers are highly sensitive to output measurement noise due to the large tuning parameter, which can become excessively high in systems with higher dimensions and nonlinearities with large Lipschitz constants. To address this issue, numerous studies have proposed various solutions : a time-varying high-gain observer can be designed where the gain starts high and gradually decreases over time [35]; a high-gain observer with limited gain power [24]; LPV/LMI-Based High-Gain Observer [9]. In [9], a new class of observers called the HG/LMI observer, was introduced by combining the standard high-gain methodology with the LPV/LMI technique [176]. This new observer offers the benefit of reducing the tuning parameters compared to previous high-gain observers, without resorting to saturation functions or filtering mechanisms.

2.3.2.3.c LMI-based Observers

LMI (Linear Matrix Inequality)-based observers are designed using optimization techniques, where the observer gain is derived by solving a set of LMIs. These LMIs are typically derived from stability and performance conditions (e.g., Lyapunov stability). The LMI approach uses convex optimization to find observer gains that meet certain criteria. We consider a class of nonlinear systems without a linear state part. Indeed, we will see later that the linear part is not necessary for the design methods we develop. For simplicity of presentation, we consider, without loss of generality, that the nonlinear function depends only on the system state, and the output is linear. The extension to nonlinear output is straightforward. The class of systems we treat in this section is described by the following equations:

$$\begin{cases} \dot{x} = \Psi(x) \\ y = Cx \end{cases} \tag{2.57}$$

where the nonlinear function $\Psi : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is assumed to be γ_Ψ-Lipschitz, i.e.:

$$\Big\|\Psi(x) - \Psi(y)\Big\| \leq \gamma_\Psi \Big\|x - y\Big\|, \quad \forall\, x, y \in \mathbb{R}^n \tag{2.58}$$

This part is devoted to the reformulated Lipschitz property from [176]. This reformulation is necessary for our developed LPV approach. A useful definition and two lemmas are presented.

Definition 2.3.14. *Consider two vectors*

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n \text{ and } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n.$$

For all $i = 0, ..., n$, we define an auxiliary vector $X^{Y_i} \in \mathbb{R}^n$ corresponding to X and Y as follows:

$$\begin{cases} X^{Y_i} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ x_{i+1} \\ \vdots \\ x_n \end{pmatrix} \text{ for } i = 1, ..., n \\ X^{Y_0} = X \end{cases} \tag{2.59}$$

Lemma 2.3.15 ([176]). *Consider a function $\Psi : \mathbb{R}^n \longrightarrow \mathbb{R}$. Then, for all*

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n \text{ and } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$$

there exist functions $\psi_j : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$, $j = 1, ..., n$ so that

$$\Psi(X) - \Psi(Y) = \sum_{j=1}^{j=n} \psi_j\Big(X^{Y_{j-1}}, X^{Y_j}\Big) e_n^T(j)\Big(X - Y\Big) \tag{2.60}$$

Proof. The proof consists of rewriting $\Psi(X) - \Psi(Y)$ as

$$\Psi(X) - \Psi(Y) = \sum_{j=1}^{j=n} \Big[\Psi\Big(X^{Y_{j-1}}\Big) - \Psi\Big(X^{Y_j}\Big)\Big]$$

Now, defining the functions ψ_j by

$$\psi_j\Big(X^{Y_{j-1}}, X^{Y_j}\Big) = \begin{cases} 0 & \text{if } x_j = y_j \\ \frac{\Psi\left(X^{Y_{j-1}}\right) - \Psi\left(X^{Y_j}\right)}{x_j - y_j} & \text{if } x_j \neq y_j \end{cases} \tag{2.61}$$

we can write

$$\begin{aligned} \Psi(X) - \Psi(Y) &= \sum_{j=1}^{j=n} \Big[\psi_j\Big(X^{Y_{j-1}}, X^{Y_j}\Big)\Big] (x_j - y_j) \\ &= \sum_{j=1}^{j=n} \Big[\psi_j\Big(X^{Y_{j-1}}, X^{Y_j}\Big) e_n^T(j)\Big] \Big(X - Y\Big) \end{aligned} \tag{2.62}$$

□

Lemma 2.3.16 ([176]). *Considering the function $\Psi : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, the two following items are equivalent:*

- *Lipschitz property : Ψ is γ_Ψ-Lipschitz with respect to its argument, i.e.:*

$$\Big\|\Psi(X) - \Psi(Y)\Big\| \leq \gamma_\Psi \Big\|X - Y\Big\|, \;\; \forall\, X, Y \in \mathbb{R}^n \tag{2.63}$$

- *Reformulated Lipschitz property : for all* $i, j = 1, ..., n$, *there exist functions*

$$\psi_{ij} \ : \ \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \ \mathbb{R}$$

and constants $\underline{\gamma}_{\psi_{ij}}$ *and* $\bar{\gamma}_{\psi_{ij}}$, *so that* $\forall \ X, Y \in \mathbb{R}^n$,

$$\Psi(X) - \Psi(Y) = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} \psi_{ij} H_{ij} \Big(X - Y \Big) \tag{2.64}$$

and

$$\underline{\gamma}_{\psi_{ij}} \leq \psi_{ij} \leq \bar{\gamma}_{\psi_{ij}} \tag{2.65}$$

where

$$\psi_{ij} \triangleq \psi_{ij} \Big(X^{Y_{j-1}}, X^{Y_j} \Big) \ \text{ and } \ H_{ij} = e_n(i) e_n^T(j)$$

In fact, Lemma (2.3.16) provides a best less conservative Lipschitz condition. Indeed, the reformulation (2.64)-(2.65) allows treatment of the nonlinearity with the best precision and exploits all the interesting properties of the nonlinearity of the investigated system. For instance, the Lipschitz condition (2.15) does not distinguish between the nonlinearities $\Psi(x) = \sin(x_2)$ and $\Psi(x) = \tanh(x_2)$. However, with the reformulation (2.65), we can see the difference. Indeed, for $\Psi(x) = \sin(x_2)$ we have $\underline{\gamma}_{\psi_{11}} = -1$ and for $\Psi(x) = \tanh(x_2)$, we have $\underline{\gamma}_{\psi_{11}} = 0$. Consider the following Luenberger observer:

$$\dot{\hat{x}} = \Psi(\hat{x}) + L\Big(y - C\hat{x} \Big) \tag{2.66}$$

The dynamic of the estimation error $e = x - \hat{x}$ is then given by:

$$\dot{e} = \Big[\Psi(x) - \Psi(\hat{x}) \Big] - LCe \tag{2.67}$$

Since $\Psi(.)$ is γ_Ψ-Lipschitz, then following Lemma 2.3.16 there are functions

$$\psi_{ij} \ : \ \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \ \mathbb{R}$$

and constants $\underline{\gamma}_{\psi_{ij}}$ and $\bar{\gamma}_{\psi_{ij}}$, so that

$$\Psi(x) - \Psi(\hat{x}) = \Big[\sum_{i=1}^{i=n} \sum_{j=1}^{j=n} \psi_{ij} H_{ij} \Big] e \tag{2.68}$$

and

$$\underline{\gamma}_{\psi_{ij}} \leq \psi_{ij} \leq \bar{\gamma}_{\psi_{ij}} \tag{2.69}$$

where

$$\psi_{ij} \triangleq \psi_{ij} \Big(x_t^{\hat{x}_{j-1}}, x^{\hat{x}_j} \Big)$$

is defined as in (2.61). By replacing Ψ by Ψ_i (the i^{th} component of Ψ). In the sequel, for simplicity of presentation, we use only ψ_{ij} instead of $\psi_{ij}\Big(x^{\hat{x}_{j-1}}, x^{\hat{x}_j} \Big)$.

Now, define the matrices

$$\Theta = \Big(\psi_{ij} \Big)_{ij} \tag{2.70}$$

and

$$\mathcal{A}\Big(\Theta\Big) = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} \psi_{ij} H_{ij} \tag{2.71}$$

Consequently, the dynamics (2.67) can be rewritten as

$$\dot{e} = \Big[\mathcal{A}\Big(\Theta\Big) - LC\Big]e \tag{2.72}$$

According to (2.69), the matrix parameter Θ belongs to a bounded convex set $\mathcal{H}_n$ for which the set of vertices is defined by:

$$\mathcal{V}_{\mathcal{H}_n} = \Big\{\Phi \in \mathbb{R}^{n\times n} \;:\; \Phi_{ij} \in \Big\{\underline{\gamma}_{\psi_{ij}}, \bar{\gamma}_{\psi_{ij}}\Big\}\Big\}. \tag{2.73}$$

The following theorem, which provides LMI conditions for observer design of Lipschitz systems.

Theorem 2.3.17 ([176]). *The observer (2.66) is asymptotically convergent if there exist a positive definite matrix $\mathcal{P}$, a matrix $\mathcal{R}$ of appropriate dimension so that the following LMI conditions hold:*

$$\mathcal{A}\Big(\Phi\Big)^T\mathcal{P} + \mathcal{P}\mathcal{A}\Big(\Phi\Big) - C^T\mathcal{R} - \mathcal{R}^TC < 0,\ \forall\, \Phi \in \mathcal{V}_{\mathcal{H}_n} \tag{2.74}$$

Hence, the observer gain is given by

$$L = \mathcal{P}^{-1}\mathcal{R}^T.$$

Proof. Consider the standard Lyapunov function

$$V\Big(e\Big) = e^T\mathcal{P}e.$$

Calculating the derivative of the Lyapunov function along the trajectories of (2.72), we obtain

$$\dot{V} = e^T\overbrace{\Bigg(\Big[\mathcal{A}\Big(\Theta\Big) - LC\Big]^T\mathcal{P} + \mathcal{P}\Big[\mathcal{A}\Big(\Theta\Big) - LC\Big]\Bigg)}^{\Omega(\Theta)}e$$

Consequently, $\dot{V} < 0$ for all $e \neq 0$ if

$$\Omega(\Theta) < 0,\ \ \forall\, \Theta \in \mathcal{H}_n \tag{2.75}$$

Therefore, from the convexity principle [40] and the notation $\mathcal{R} = L^T\mathcal{P}$ we deduce that inequality (2.75) holds if LMI (2.74) is verified for all $\Phi \in \mathcal{V}_{\mathcal{H}_n}$. This means that $\dot{V} < 0,\ \forall e \neq 0$, which ends the proof.

□

It is clear that from the computational complexity point of view, the LPV method is less interesting, viewing the cost of more demanding LMIs to be solved. In fact, in the LPV method, we have to solve 2^{n^2} LMIs for n-dimensional nonlinear vectors. But, despite that, the LPV method provides less restrictive LMI synthesis conditions. On the other hand, despite the cost of more demanding LMIs, this does not play an important role in the feasibility of the proposed LMIs. In general, computational complexity plays a role in online methods for real-time applications.
In the scope of this thesis, LMI-based observers are further investigated, mainly in Chapter 4, to provide accurate state for the UAV operating in the AVF. More general and less conservative LMI conditions presented in [7] are recalled in order to, first design a robust observer-based $\mathcal{H}_\infty$ via LMIs, along with a second LMI-based observer design ensuring the Input-to-State Stability (ISS) property of the estimation error.

2.4 Conclusion

In conclusion, this chapter has provided the necessary theoretical foundation for understanding the key concepts and methodologies that support the work presented in the subsequent chapters. and sets the stage for the practical application of these methodologies in the crop growth modeling and the design and implementation of robust observer systems for the UAV in a Vertical Farming environment, which will be explored in the following chapters.

CHAPTER 3

Data-Driven Black-Box Dynamic Models for Crop Growth

Contents

3.1 Introduction

Given that the cultivation shelves move vertically according to the growth of crops (see Figure 3.1), having at disposal an accurate growth model of crops, able to provide the actual height of plants at each time instant of the growth cycle, is fundamental to ensure the dynamic architecture of the AVF as motivated in 1.3. Furthermore, in order to achieve the best performance for the AVF in terms of resources management and production yield, an optimal schedule of seedings is fundamental. In [27, 30, 132, 133], a scheduling approach of seedings based on mixed-integer linear programming is proposed to maximize the number of sowings, and consequently, the number of harvests, in a given timeframe (i.e. the production cycle). An optimal seeding schedule enables the implementation of control schemes allowing the adjustment of between-shelf height according to crop growth. Hence, the accuracy of the crop growth model, informing about the plant height at each time step is a key factor for effective scheduling. Accurate crop models allow also the optimization of greenhouse indoor climate for an efficient resource management including leaf area, light intensity, CO_2 concentration, and other key agronomic factors, thus, they are worth investigating in order to assess the overall AVF performance. In this research work, without loss of generality, we focus on the case of lettuce, which is one of the most considered crops in indoor cultivation due to its short growth cycle (usually between 30 and 60 days depending on the variety), high market demand, ease of cultivation, space efficiency given its compact form, low light requirements, and sensitivity to environmental factors such as temperature, humidity, and nutrient levels, which can be accurately controlled in indoor farming systems allowing for multiple harvests per year and consistent production. As a commonly grown crop in vertical farming systems, lettuce mirrors the growth dynamics of leafy greens, herbs, and microgreens, which allows the model to generalize across a range of similar crops. Crop production in VFs differs from the standard horizontal greenhouses, giving the different indoor climate conditions. This leads to several studies on the lettuce production within VFs such as plant factories (readers can refer to [110] and [118]). In the literature, most studies on the lettuce production in vertical farms are based on the growth models that consider the dry mass (also denoted by dry weight), i.e., the weight of the dried shoot, as the key crop growth indicator being a more reliable indicator compared to fresh weight, i.e. the weight of a plant immediately after it is harvested, without any dehydration. In fact, dry mass provides perspectives on crop height during the growth cycle considering the accumulation of dry mass which affects structural components like stems, leaves, and roots, thus enabling to estimate the height of the crop at different time instants during the production cycle, and therefore it enables the correct scheduling of seedings, and control strategies for the vertical movement of shelves, as discussed previously. According to [110], different dry mass-based crop models are available such as the 3-dimensional crop models involving leaf angles and light spectrum tracing as models based on the representation of the CO_2 [65, 140]. However, the most considered crop growth model in VFs is the one developed by Van Henten [82], which is a two-state dynamic model depending on a pair of ordinary differential equations (ODEs). Nevertheless, the crop growth model considered in [82] has several limitations, as highlighted below:

- It is specifically designed for lettuce cultivation, making it unsuitable for other crops.
- The model relies on several parameters, which may affect its prediction capabilities [83, 158]
- The model requires solving a pair of ordinary differential equations (ODEs), which can be computationally intensive, especially when integrated into optimization procedures. This is particularly challenging when coupling the model with an optimal seeding schedule to maximize production within the constraints of available vertical space [**?**, 28, 29, 31].
- Originally developed for traditional horizontal greenhouses, the model's application to vertical farming systems may not be entirely accurate. While some studies address this issue

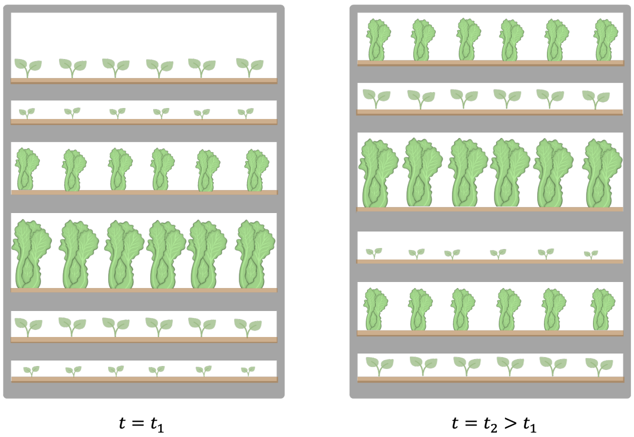


Figure 3.1 – Example of cultivation shelf composition for the growth of lettuce in an AVF at two different time steps t_1 and t_2.

in the context of vertical farms [110, 111, 118], the model's adaptation to AVFs, with its dynamic-shelve system, remains not investigated.

In the Chapter 3, and in order to address the aforementioned shortcomings, we propose, firstly, a parameter tuning through an identification procedure in order to enhance the crop height prediction. Secondly, we investigate the development of a model of crop growth based on data that can be collected on field, deploying the black-box paradigm as motivated in 1.2.2.3.
The Chapter 3 is organised as follow, Section 3.2 is dedicated to the lettuce growth dynamic model proposed by [82]. Section 3.3 reports the proposed parameter estimation strategy published in [45]. The proposed data-driven, black-box model is described in Section 3.4 whereas the simulation results are presented in 3.5.

3.2 Dynamic Model for Lettuce Growth

In this section, we briefly report the crop growth model proposed in [82] for lettuce. The key variable defining the growth is the dry weight, which indirectly influences the plant height. As pointed out also in [158, 161], at any time t, the dry weight is made up of two components: the structural dry weight, referred to in the following as $x_1(t)$ (measured in $g{\cdot}m^{-2}$), which encompasses cell walls and cytoplasm, and the non-structural dry weight, denoted by $x_2(t)$ (measured in $g{\cdot}m^{-2}$), which includes glucose, sucrose, and starch.
The dynamic evolution of the plant growth for $t \in [0, T]$, where $T > 0$ is a given horizon length, can be captured by the following continuous-time dynamic model characterized by a state variable:

$$\underline{X}(t) := \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^\top \in \mathbb{R}^2, \tag{3.1}$$

such as:

$$\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} = r_{\text{gr}}(t)\, x_1(t), \tag{3.2}$$

$$\begin{aligned} \frac{\mathrm{d}x_2(t)}{\mathrm{d}t} &= c_{\text{ch2o}}\, \phi_{\text{phot}}(t) - r_{\text{gr}}(t)\, x_1(t) - \phi_{\text{resp}}(t) \\ &\quad - \frac{1 - c_y}{c_y} r_{\text{gr}}(t)\, x_1(t), \end{aligned} \tag{3.3}$$

Equations (3.2)–(3.3) are ODEs involving several functions of time and constant parameters. In more detail, $r_{\text{gr}}(t)$ is the specific growth rate, $\phi_{\text{phot}}(t)$ is the gross canopy photosynthesis, and

$\phi_{\text{resp}}(t)$ is the maintenance respiration. The constant parameters c_{ch2o} and c_y express the conversion rate of CO_2 into CH_2O and the yield factor measuring the non-structural dry mass wastes due to respiration and photosynthetic activities, respectively.
In more detail, the specific growth rate $r_{\text{gr}}(t)$ at time t is given by:

$$r_{\text{gr}}(t) = c_{\text{gr,max}} \frac{x_2(t)}{c_\gamma x_1(t) + x_2(t)} c_{\text{q10,gr}}^{\frac{u(t)}{10}-2}, \tag{3.4}$$

where c_γ is the growth rate coefficient and $c_{\text{q10,gr}}$ is the measure of growth rate sensitivity to canopy temperature. The quantity $u(t)$ represents the temperature, serving as a system input.
The gross canopy photosynthesis rate $\phi_{\text{phot}}(t)$ is described by the following relationship:

$$\phi_{\text{phot}}(t) = \left(1 - e^{-c_k c_{\text{lar}}(1-c_\tau)x_1(t)}\right) \phi_{\text{phot,max}}(t), \tag{3.5}$$

where c_k, c_{lar}, and $\phi_{\text{phot,max}}(t)$ express the light extinction coefficient, the structural leaf area ratio, and the gross CO_2 assimilation rate, respectively. The parameter c_τ is the root dry mass ratio to the overall dry mass of the crop, dependent on the crop type. Equation (3.5) includes the quantity $\phi_{\text{phot,max}}(t)$, which describes the response of canopy photosynthesis to photosynthetically active radiation, carbon dioxide concentration in the greenhouse air, and temperature, i.e.:

$$\phi_{\text{phot,max}}(t) = \frac{\varepsilon(t) u_{\text{par}}(t) \sigma(t) \left(u_{\text{co2}}(t) - \Gamma(t)\right)}{\varepsilon(t) u_{\text{par}}(t) + \sigma(t) \left(u_{\text{co2}}(t) - \Gamma(t)\right)}, \tag{3.6}$$

where $u_{\text{par}}(t)$ and $u_{\text{co2}}(t)$ are inputs denoting the incident photosynthetically active radiation and the CO_2 concentration, respectively, while $\varepsilon(t)$ is the light use efficiency, which is given by

$$\varepsilon(t) = c_\varepsilon \frac{u_{\text{co2}}(t) - \Gamma(t)}{u_{\text{co2}}(t) + 2\Gamma(t)}, \tag{3.7}$$

where c_ε is the light use efficiency at very high CO_2 concentrations. The quantity $\Gamma(t)$ in (3.6) and (3.7) depends on the temperature, as follows:

$$\Gamma(t) = c_\Gamma \, c_{\text{q10},\Gamma}^{(u(t)-20)/10}, \tag{3.8}$$

where c_Γ is the CO_2 compensation point at 20°C. The quantity $\sigma(t)$ in (3.6) is the leaf conductance, which is described by the expression

$$\frac{1}{\sigma(t)} = \frac{1}{c_{\text{bnd}}} + \frac{1}{c_{\text{stm}}} + \frac{1}{c_{\text{car}}(t)}, \tag{3.9}$$

where c_{bnd}, c_{stm}, and $c_{\text{car}}(t)$ are the boundary layer, the stomatal, and the carboxylation conductances, respectively. The latter depends on temperature variation within a range between 5°C and 40°C, as described by the following polynomial:

$$c_{\text{car}}(t) = c_{\text{car},1} u(t)^2 + c_{\text{car},2} u(t) + c_{\text{car},3}, \tag{3.10}$$

where $c_{\text{car},1}$, $c_{\text{car},2}$, and $c_{\text{car},3}$ are proper constants.
Lastly, the maintenance respiration $\phi_{\text{resp}}(t)$ is given by:

$$\phi_{\text{resp}}(t) = \left(c_{\text{resp,sht}} \, (1 - c_\tau) + c_{\text{resp,rt}} \, c_\tau\right) x_1(t) c_{\text{q10,resp}}^{\frac{u(t)}{10}-\frac{5}{2}}, \tag{3.11}$$

where $c_{\text{resp,sht}}$ and $c_{\text{resp,rt}}$ are shoot and root maintenance respiration coefficients at 25°C, respectively.

Table 3.1 – Parameters and simulation values of the considered lettuce growth model (3.12) according to [83].

Parameter	Value	Unit of measurement
c_{ch2o}	0.68	-
c_y	0.8	-
c_γ	1	-
c_ε	$17 \cdot 10^{-6}$	$g \cdot J^{-1}$
c_Γ	$7.32 \cdot 10^{-3}$	$g \cdot m^{-3}$
c_k	0.9	-
c_τ	0.07	-
c_{lar}	$75 \cdot 10^{-3}$	$m^2 \cdot g^{-1}$
$c_{\text{gr,max}}$	$5 \cdot 10^{-6}$	s^{-1}
$c_{\text{q10,}\Gamma}$	2	-
$c_{\text{q10,gr}}$	1.6	-
$c_{\text{q10,resp}}$	2	-
$c_{\text{resp,sht}}$	$3.47 \cdot 10^{-7}$	s^{-1}
$c_{\text{resp,rt}}$	$1.16 \cdot 10^{-7}$	s^{-1}
c_{bnd}	0.004	$m \cdot s^{-1}$
c_{stm}	0.02	$m \cdot s^{-1}$
$c_{\text{car, 1}}$	$-1.32 \cdot 10^{-5}$	$m \cdot s^{-1} \cdot {}^\circ C^{-2}$
$c_{\text{car, 2}}$	$5.94 \cdot 10^{-4}$	$m \cdot s^{-1} \cdot {}^\circ C^{-1}$
$c_{\text{car, 3}}$	$-2.64 \cdot 10^{-3}$	$m \cdot s^{-1}$
$\alpha_{\text{co2,1}}$	1.464	$g \cdot m^{-3}$
$\alpha_{\text{co2,2}}$	1.281	$g \cdot m^{-3}$
$\alpha_{t,1}$	18.75	$^\circ C$
$\alpha_{t,2}$	11.25	$^\circ C$
$\alpha_{\text{par,1}}$	125	$W \cdot m^{-2}$
$\alpha_{\text{par,2}}$	125	$W \cdot m^{-2}$

The overall dynamic model (3.2)–(3.11) can be compactly expressed as:

$$\frac{\mathrm{d}\underline{X}(t)}{\mathrm{d}t} = f(\underline{X}(t), \underline{U}(t), t, \underline{P}), \tag{3.12}$$

where $\underline{U}(t) := \text{col}[u(t), u_{\text{co2}}(t), u_{\text{par}}(t)] \in \mathbb{R}^3$ collects all the system inputs $u(t)$, $u_{\text{co2}}(t)$, and $u_{\text{par}}(t)$ at time t, $\underline{P} \in \mathbb{R}^{19}$ includes all the parameters in (3.2)–(3.11), and $f : \mathbb{R}^2 \times \mathbb{R}^3 \times [0, T] \times \mathbb{R}^{19} \to \mathbb{R}^2$ is a vector field capturing the nonlinear relationship among $\underline{X}(t)$, $\underline{U}(t)$, t, and $\underline{P}$.

Table 3.1 summarizes all the parameters included in the vector $\underline{P}$, along with their numerical values, as reported in [83]. We observe that the dynamic model (3.2)–(3.11) depends on numerous parameters. Each value may influence the temporal behavior of the state variables $x_1(t)$ and $x_2(t)$. We refer the reader to the sensitivity analysis [83] to fully understand the parameters' influence on the crop growth. Hence, in order to have an accurate model that is able to effectively predict the crop growth, it is crucial to properly estimate such parameters based on the available data.

3.3 Parameter Estimation of the Dynamic Growth Model for Lettuce in the AVF

In this section, we perform parameter estimation in order to effectively predict the dry mass production and thus the crop growth whithin the AVF. As mentionned previously, the idea behind carrying out a precise parameter tuning is that the existant estimates of the multiple parameters are outdated [72, 83, 158, 161]. To our knowledge, few are the works that have investigated parameter calibration for indoor lettuce production, mainly [87].
According to the sensitivity analysis performed in [83], studying the impact of parameter changes on growth dynamics, some parameters have more significant effect on the dry mass production than others. Mainly, the yield factor c_y, the light extinction coefficient c_k, the structural leaf area ratio c_{lar}, the light use efficiency c_ε, and the maximum growth rate $c_{\text{gr,max}}$. For convenience, it is assumed that $c_{\text{gr,max}}$ and c_ε are known as mentioned in [82]. As a result, we investigate estimating the following subset of parameters of $\underline{P}$, gathered in the vector $\underline{p}$:

$$\underline{p} := \begin{bmatrix} c_y & c_k & c_{\text{lar}} \end{bmatrix}^\top \in \mathbb{R}^3. \tag{3.13}$$

It is worth to note that a broader set of parameters might be considered without conceptual issues. Let $\underline{X}^{\text{meas}}(t)$ represent the dry mass measured at time t, and for simplicity, let $\underline{X}(t, \underline{p})$ denote the dry mass predicted by the dynamic model in equation (3.12), based on a specific value of the parameter vector $\underline{p}$. The parameter estimation approach is based on solving an optimization problem designed to minimize the discrepancy between the measured output and the estimated output over time. This difference is averaged over a specified number L of realizations of the measured variables $\underline{X}^{\text{meas}}(t)$. The choice of L different realizations is made to obtain parameters that are able to reproduce a variety of dynamic behaviors rather than only a specific one. In more detail, let $\underline{X}^{\text{meas}(l)}(t)$ and $\underline{X}^{(l)}(t, \underline{p})$ be the measured output and estimated one for the l-th realization, $l = 1, \ldots, L$. We perform a parameter identification in the sense of obtaining the best estimation $\hat{\underline{p}}$ of $\underline{p}$ resulting from the minimization of a least squares cost that fits the available data of the dry mass at different time instants.
The selection of L different realizations is intended to ensure that the estimated parameters can capture a wide range of dynamic behaviors, rather than focusing solely on a single specific behavior. Specifically, let $\underline{X}^{\text{meas}(l)}(t)$ and $\underline{X}^{(l)}(t, \underline{p})$ represent the measured and estimated outputs, respectively, for the l-th realization, where $l = 1, \ldots, L$. The parameter identification process aims to find the optimal estimation $\hat{\underline{p}}$ of $\underline{p}$, which is obtained by minimizing a least squares cost function. This cost function fits the available data of the dry mass at different time points. Thus, optimization problem to be solved is the following:

$$\hat{\underline{p}} := \arg \min_{\underline{p} \in [\underline{p}_{\text{min}}, \underline{p}_{\text{max}}]} J(\underline{p}), \tag{3.14}$$

where $\underline{p}_{\text{min}} \in \mathbb{R}^3$ and $\underline{p}_{\text{max}} \in \mathbb{R}^3$ are proper lower and upper bounds for $\underline{p}$, respectively, and the cost function $J(\underline{p})$ is defined as follows:

$$J(\underline{p}) := \frac{1}{L} \frac{1}{|\mathcal{T}|} \sum_{l=1}^{L} \sum_{t \in \mathcal{T}} \left\| \underline{X}^{\text{meas},(l)}(t) - \underline{X}^{(l)}(t, \underline{p}) \right\|^2, \tag{3.15}$$

where $\mathcal{T}$ is a set including time instants in the interval $[0, T]$ for which measurements of the dry weight are available. In this context, we can assume, for instance a uniform discretization of the interval $[0, T]$, with sampling time Δt, with $|\mathcal{T}|$ is the number of elements in $\mathcal{T}$. The minimization in (3.14) is performed subject to equations (3.2)–(3.10) reported in Section 3.2 for all $l = 1, \ldots, L$.

Even though the calibration can be performed efficiently through the identification process, predetermining the model structure might restrict its predictive capabilities, potentially making it incapable of capturing some complex crop behaviors. Therefore, the investigation of a black-box model in Section 3.4 mimicking the crop dynamics is a fundamental step to represent a large variety of dynamic behaviors for different crop types and varieties and reduce computational burden owing to the flexibility and approximation capabilities of these structures as mentioned in Section 2.2.

3.4 Black-Box Model for Lettuce Growth

The building block of the data-driven crop growth model consists of feedforward neural networks, sometimes referred to as multilayer perceptrons. These networks constitutes a fundamental and extensively-used type of artificial neural networks, which are the basis of numerous machine learning applications across different fields. They are characterized by interconnected layers of artificial neurons, also called nodes or computational units, arranged sequentially. The information flows unidirectionally from the input to the output layer through one or more hidden layers, giving rise to the term "feedforward". In the hidden layer of such networks, examples of computational units include radial basis functions, kernel units, and sigmoidal functions. Among the various possibilities, we focus on one-hidden-layer networks, a specific type characterized by a single input layer, one output layer, and one hidden layer.
Theoretical results abound for one-hidden-layer feedforward neural networks, with a crucial aspect being the universal approximation property [34]. This property refers to the capability of such networks to approximate any well-behaved, unknown function that generates a set of data with arbitrary accuracy, provided an appropriate number of computational units in the hidden layer is selected. Additionally, under suitable regularity assumptions on the unknown mapping generating the data, these networks ensure uniform approximations characterized by upper bounds with a number of parameters growing at most polynomially with the dimension of the input of the function to be approximated [34, 98, 146].
We use neural networks to approximate the unknown mapping that, given as input the structural and non-structural dry weights, the system inputs given by the temperature, the incident phosynthetically active radiation, and the CO_2 concentration at a given time instant, generates as output the state vector at the next step. The aforementioned procedure implies a discretization procedure. In more detail, we assume that the interval of interest $[0, T]$ is sampled in $N+1$ discrete time instants $0, \Delta t, 2\Delta t, \ldots, N\Delta t$, where Δt is the distance between two consecutive points. With a little abuse of notation, in the following we will refer to such discrete time instants as $n = 0, 1, 2, \ldots, N$, by dropping Δt. Toward this end, let $\underline{X}_n = \underline{X}(n\Delta t) \in \mathbb{R}^2$ be the state of the system, i.e., the structural and non-structural dry weights at the time step n, and let $\underline{U}_n = \underline{U}(n\Delta t) \in \mathbb{R}^3$ be the system inputs, i.e., the temperature, the incident phosynthetically active radiation, and the CO_2 concentration at the same time step. The goal is to approximate the mapping

$$(\underline{X}_n, \underline{U}_n, n) \mapsto \underline{X}_{n+1}, \quad n = 0, 1, \ldots, N-1.$$

The black-box version of the dynamic model (3.12) is therefore given by

$$\underline{X}_{n+1} = \gamma(\underline{X}_n, \underline{U}_n, n, \underline{W}), \quad n = 0, 1, \ldots, N-1, \tag{3.16}$$

where γ is a one-hidden-layer feeforward neural network and $\underline{W} \in \mathbb{R}^p$ is the vector of its parameters (sometimes called weights in the neural network palance). The parametrized function γ is given by

$$\gamma(\underline{X}_n, \underline{U}_n, n, \underline{W}) = \sum_{i=1}^{\nu} \underline{c}_i \phi\left(\sum_{j=1}^{6} \underline{a}_{ij} \tilde{X}_{n,j} + \underline{b}_i\right) + \underline{c}_0, \tag{3.17}$$

where $\underline{\tilde{X}}_n = \text{col}(\underline{X}_n, \underline{U}_n, n) \in \mathbb{R}^6$, $\tilde{X}_{n,j}$ is its j-th component, $j = 1, \ldots, 6$, ν is the number of computational units in the hidden layer (usually referred to as neurons), ϕ is the activation or basis function, and $\underline{c}_i$, $\underline{c}_0$, $\underline{b}_i$, and $\underline{a}_{ij}$ are the neural weights included in the vector $\underline{W}$. Without loss of generality, in this work we will focus on sigmoidal activation functions ϕ, as this is quite a common choice in one-hidden-layer networks.

We observe that the application of the black-box model (3.16) allows to avoid the integration of the ODE (3.12), as the former is constructed to provide as output directly the state at the next time step. As pointed out in the introduction, this represents an important advantage in terms of computational requirements, especially in the case the model is used within an optimization algorithm, which requires several, repeated solutions with different state and input vectors at the various iterations. This is the case of the AVF, where an optimal scheduling of seedings is required to improve the overall efficiency. The model provides forecasts of the height of the plants (throught the information on the dry weight), which is essential to avoid exceeding the total height of the greenhouse.

Clearly, the model (3.16) requires a proper training procedure for the parameter vector $\underline{W}$. In more detail, the following procedure can be used to construct the model starting from data collected on the field:

(i) acquire data on the structural and non-structural dry weights as well as on the system inputs at different discrete time instants n and the corresponding values of the state at the next time step $n+1$. Let $\underline{\tilde{X}}_n^{(l)} = \text{col}(\underline{X}_n^{(l)}, \underline{U}_n^{(l)}, n^{(l)}) \in \mathbb{R}^6$ be the l-th measure of the input data for γ and let $\underline{X}_{n+1}^{(l)} \in \mathbb{R}^2$ be the corresponding output, for $l = 1, \ldots, L$, where L is the number of collected samples;

(ii) apply a training method to approximate the collected pairs $\left(\underline{\tilde{X}}_n^{(l)}, \underline{X}_{n+1}^{(l)}\right)$, $l = 1, \ldots, L$.

Roughly speaking, the training procedure consists of searching for the optimal values of the parameter vector $\underline{W}$ that minimize the quadratic difference between the measures and the output of the function γ, i.e.,

$$\underline{W}^{\star} = \arg \min_{\underline{W} \in \mathbb{R}^p} \sum_{l=1}^{L} \left(\underline{X}_{n+1}^{(l)} - \gamma(\underline{X}_n^{(l)}, \underline{U}_n^{(l)}, n^{(l)}, \underline{W}) \right)^2. \tag{3.18}$$

Several efficient algorithms and software packages exist to solve the optimization problem (3.18). Examples are given by the Levenberg-Marquardt method or the backpropagation one [79]. As it is well-known, the training procedure may be computationally intensive, but it is made offline, i.e., before operating the greenhouse. After the training, the application of the black-box model reduces to the computation of the algebraic relationship (3.16), which enables large savings as compared to the numerical solution of the ODE (3.12).

3.5 Simulation Results

In this section, we investigate the numerical results of the proposed approaches in Section 3.3 and Section 3.4. We first assess the identifiability of the parameters included in the vector $\underline{p}$ within the dynamic model (3.12) starting from measurements of the dry weight. And then, we test the effectiveness of the proposed data-driven, black-box modeling approach to forecast the crop growth in two case studies.

3.5.1 Parameter Estimation

To showcase the effectiveness of the parameter identification process, numerical results are based on results relying on both synthetic and real-world data measurements. Since a prototype of the

AVF is newly constructed with no cultivation experiment conducted within yet, no data is directly collected on the field, the test is, first, based on simulated data to acquire a wide variability of available measurements. Then, we perform parameter identification to predict crop growth using field-collected data originated from the measurements performed during the third International Autonomous Greenhouse Challenge performed in 2022 at Wageningen University & Research (WUR), in Bleiswijk, The Netherlands. All tests were carried out on a computer equipped with Apple M1 CPU, 16 GB of RAM, and a Matlab R2022b version. The measurements are collected in a publicly-available dataset [4]. The two rounds of simulations and corresponding results are reported in Subsections 3.5.1.1 and 3.5.1.2, respectively:

3.5.1.1 Results with a Synthetic Dataset

A set of synthetic data is generated for both the structural and non-structural dry weight representing the common crop growth behaviors. For this purpose, we first notice that the dry mass production defined by (3.2)–(3.10) is dependent on the system inputs $\underline{U}(t)$, i.e., temperature, carbon dioxide concentration, and photo-synthetically active radiation. Therefore, a set of significant values of the state $\underline{X}(t)$ to be used for parameter estimation can be generated by properly varying $\underline{U}(t)$. In this work, we adopted the approach detailed in [83], which consists in first generating an uncorrelated zero mean white noise $w(t) \in [-1, 1]$ and then transforming it into time sequences to cover a wide range of input amplitudes and frequencies, as follows:

$$u(t) = \alpha_{t,1} + \alpha_{t,2} w(t), \tag{3.19}$$

$$u_{\text{co2}}(t) = \alpha_{\text{co2},1} + \alpha_{\text{co2},2} w(t), \tag{3.20}$$

$$u_{\text{par}}(t) = \alpha_{\text{par},1} + \alpha_{\text{par},2} w(t), \tag{3.21}$$

for $t \in [0, T]$, where $\alpha_{t,1}$, $\alpha_{\text{co2},1}$, and $\alpha_{\text{par},1}$ are the average values of the input signals $u(t)$, $u_{\text{co2}}(t)$, and $u_{\text{par}}(t)$, respectively, while $\alpha_{t,2}$, $\alpha_{\text{co2},2}$, and $\alpha_{\text{par},2}$ are the amplitude of the variations around the aforementioned input values. The numerical values of such parameters are reported in Table 3.1.
Let $\underline{p}^* := \begin{bmatrix} 0.8 & 0.9 & 0.075 \end{bmatrix}^\top \in \mathbb{R}^3$ be the vector of "true" parameters (such values are those reported in Table 3.1), used to simulate the behavior of the state variables starting from $L = 100$ initial conditions randomly extracted according to uniform distributions in the ranges $[0.25, 25]$ and $[0.1, 10]$ for $x_1(0)$ and $x_2(0)$, respectively. The trajectory corresponding to the l-th initial condition was denoted by $\underline{X}^{(l)}(t)$ for $t \in [0, T]$ and $l = 1, \ldots, L$, with $\underline{U}(t)$ defined as in (3.22)–(3.24) for each time $t \in [0, T]$. Since the dynamics of the dry mass production are slow, the simulations were performed over $T = 5 \cdot 10^6$ s, i.e., about 50 days. The set $\mathcal{T}$ of time instants where we supposed to have measures of the dry weight was obtained by uniformly sampling the interval $[0, T]$ with 100 points.
It is worth noting that, while the synthetic data generation approach can replicate expected crop growth behavior, it may not fully capture every possible real-world scenario. For example, real-world conditions may introduce additional complexities like sudden environmental changes, or non-ideal growing conditions, which are not accounted for in the model. Additionally, while the noise-driven variation covers a wide range of scenarios, it may still fail to replicate the full diversity of possible real-world inputs. This may introduce a potential bias in the data. To avoid overfitting, we ensure that the data covers a wide range of possible input variations. However, further validation against real-world data would be necessary to fully assess the model's robustness and ability to adapt to unpredictable real-world conditions.
The obtained growth curves were saved and used as measurements $\underline{X}^{\text{meas}(l)}(t)$, $l = 1, \ldots, L$, to implement the parameter estimation procedure described in Section 3.3. In this regard, the lower and upper bounds for the parameters were fixed to $\underline{p}_{\text{min}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$ and $\underline{p}_{\text{max}} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top$, respectively, whereas the initial values of the parameters within the optimization procedure were taken equal to $\underline{p}_0 = \begin{bmatrix} 0.2 & 0.5 & 0.5 \end{bmatrix}^\top$.

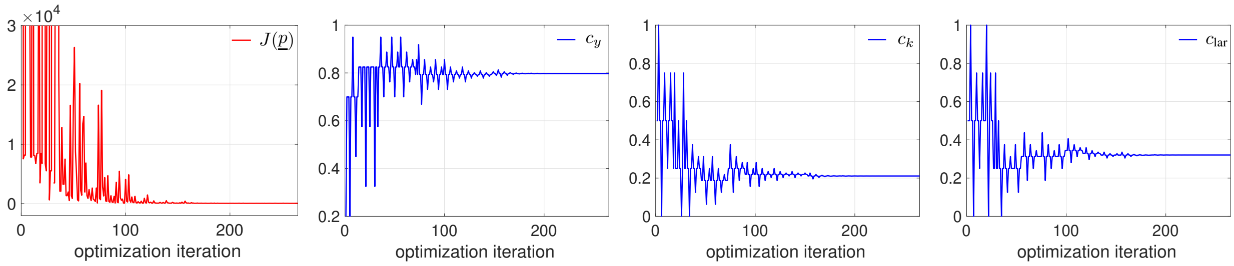


Figure 3.2 – Trends of the cost function J and of the estimated parameters in $\underline{p}$ at the various iterations of the optimization procedure based on synthetic data.

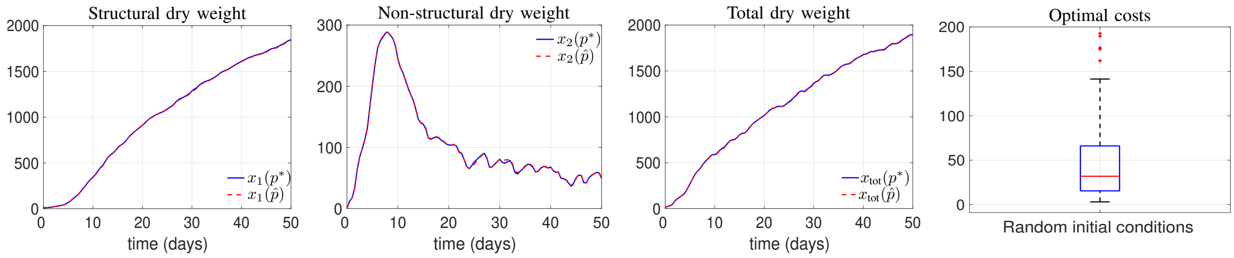


Figure 3.3 – Example of behavior of the structural dry weight, non-structural dry weight, and total dry weight compared to a reference trajectory obtained with the "true" parameters $\underline{p}^*$ for a randomly-selected initial condition among the considered 100 synthetic ones, and boxplot of the optimal costs computed over the various trajectories.

Figure 3.2 depicts the behavior of the cost $J(\underline{p})$ and of the identified parameter vector $\underline{p}$ at the various iterations of the optimization process. After an oscillating behavior, the cost and the parameters converged to stationary values, which were taken equal to the optimal estimates $J(\underline{\hat{p}})$ and $\underline{\hat{p}}$, respectively. In more detail, the optimal parameters obtained by the optimization algorithm were $\underline{\hat{p}} := \begin{bmatrix} 0.8 & 0.21 & 0.32 \end{bmatrix}^\top$, while the optimal cost $J(\underline{\hat{p}})$ was equal to 61.3.

The accuracy of the obtained optimal parameters $\underline{\hat{p}}$ was evaluated by simulating the crop growth by starting again from 100 initial conditions, but different from the ones used for parameter estimation in (3.14). In more detail, we randomly generated $M = 100$ initial conditions by considering again random extractions from uniform distributions in the ranges $[0.25, 25]$ and $[0.1, 10]$ for $x_1(0)$ and $x_2(0)$, respectively. Figure 3.3 reports a randomly-extracted trajectory of the state variables together with the trajectory obtained with the "true" parameters $\underline{p}^*$, used as a reference. We observe a satisfactory overlap of the two trajectories, thus proving the accuracy of the estimation, also for initial conditions different than the ones used during the optimization process. The effectiveness of the estimation is confirmed also by the boxplot of the optimal cost function over the considered different $M = 100$ random initial condition used for assessing performances (see the last plot in Figure 3.3). The median of the cost (the red, horizontal line of the boxplot), i.e., the difference between the trajectories obtained with the optimal parameters $\underline{\hat{p}}$ and the "true" ones, is equal to 31.8, which is quite a satisfactory result. In other words, the parameters resulting from the optimization have a good capability of approximating various behaviors of the system.

If we compare the optimal estimates $\underline{\hat{p}}$ and the "true" parameters $\underline{p}^*$, we note that they are quite different, except for the value of c_y, which is perfectly estimated. This reveals that some parameters of the dynamic model (3.12) may be not correctly estimated starting from available measurements on the dry weight. However, the optimal cost is reduced, and the approximation of the trajectories is satisfactory. This means that there exist different values of the parameters that provide similar results in terms of trajectories.

3.5.1.2 Results with a Dataset of Real-World Measurements

As said, we considered the dataset [4] containing measurements of crop growth and greenhouse conditions collected during the third International Autonomous Greenhouse Challenge (See Figure 3.4). Such data were collected on a horizontal greenhouse rather than an AVF, but they can be considered representative of the growth of lettuce in an indoor, fully controlled environment. In fact, the AVF is at present just a concept, and prototypes are currently under construction. Thus, it is not possible to actually measure the growth of the plants and use them as reference to guide the estimation procedure. For this reason, we adopted the dataset [4] as a benchmark.

Figure 3.4 – Real images of lettuce growth taken from the dataset [4]. From top to bottom, we have pictures of the crop on the day of transplanting, during the cultivation cycle, and on the harvest day for the team "CVA".

The dataset was generated during a lettuce growing experiment conducted in six different compartments of the greenhouse. Each compartment had a total area of $96\ \mathrm{m}^2$, and a growing area of $16\ \mathrm{m}^2$.

During the experiment, two cultivation cycles of lettuce (variety "Lugano") were conducted by five different participating teams (named "CVA", "DigitalCucumbers", "Koala", "MondayLettuce", and "VeggieMight"). Each team had at disposal a different compartment, with distinct controlled climates, which can be set independently by the various teams. The sixth compartment was cultivated by a group of WUR researchers (we considered it as a sixth team participating in the challenge). The dataset contains time-series data on controlled indoor climate for each compartment, along with crop images and measurements on the total dry weight, height, and diameter of plants. In more detail, the indoor temperature $u(t)$, the photosynthetically active radiation $u_{\text{par}}(t)$, and the indoor CO_2 concentration $u_{\text{co2}}(t)$ introduced in Section 3.1 were collected via standard sensors in 5-minute intervals, as discussed in [5]. To mimic the commercial approach in which cultivators solely monitor and assess crop production visually, daily images were taken in 15-minute intervals. In particular, the dataset includes the collected RGB and depth data, with a resolution of 1080×1920 pixel for both types of images. Measurements of total dry weight, height, and diameter were obtained for six weeks with a 7-day interval using a destructive measurement method on six individual lettuce plants sampled from each compartment.

For the aim of parameter estimation considered in this work, we performed a data pre-processing using Matlab, with the goal of expressing numerical values in the same units of measurements used by the dynamic model (3.2)–(3.10), ensuring a complete dataset without missing data, and reducing measurement noise. Since the initial dataset of the total dry weight measurements was obtained for 6 weeks with a sampling time of 7 days, a linear interpolation technique was applied to set a uniform sampling time of 5 minutes, from the first day after transplanting to the harvest day. In this way, we aligned data of the plants dry weight with the available information on the control inputs $u(t)$, $u_{\text{par}}(t)$, and $u_{\text{co2}}(t)$. On the overall, we had 10081 data points for each variable for each team participating in the challenge. In other words, using the terminology introduced in Section 3.3, we had $L = 6$ different realizations $\underline{X}^{\text{meas}(l)}(t)$ for $l = 1, \ldots, 6$ and $t = 0, 1, \ldots, 10080$. Furthermore, to address missing values in the indoor greenhouse climate and total dry weight or diameter measurements, a linear interpolation was used as imputation technique to estimate missing values through neighboring data points. Lastly, noise reduction was performed by smoothing data through an average over a 100-element sliding window over all the available time series. The optimization (3.14) was performed by choosing again $\underline{p}_0 = \begin{bmatrix} 0.2 & 0.5 & 0.5 \end{bmatrix}^\top$ as initial guess of the parameters, and bounds $\underline{p}_{\text{min}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$ and $\underline{p}_{\text{max}} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\top$. Figure 3.5 depicts the behavior of the cost $J(\underline{p})$ and of the identified parameter vector $\underline{p}$ at the various iterations of the optimization process. Likewise for the case of synthetic data described in Section 3.5.1.1, after an oscillating behavior, the cost and the parameters converged to stationary values, which were considered as the optimal estimates $J(\hat{\underline{p}})$ and $\hat{\underline{p}}$, respectively. In more detail, the optimal parameters obtained by the optimization algorithm were $\hat{\underline{p}} := \begin{bmatrix} 0.15 & 0.07 & 0.36 \end{bmatrix}^\top$, while the optimal cost $J(\hat{\underline{p}})$ was equal to 239.1. Figure 3.6 reports an example of trajectory for the total dry weight (dashed, red line) and the corresponding estimate provided by the model with the identified parameter vector $\hat{\underline{p}}$ (blue line). Likewise in the case of the use of synthetic data, we observe a satisfactory overlap of the two trajectories, thus proving the accuracy of the estimation. The effectiveness of the estimation is confirmed also by the boxplot of the optimal cost function over the available test trajectories (see the last plot in Figure 3.6). The median of the cost is equal to 509.6. As expected, the value of the optimal costs when using real data are higher than the one obtained with synthetic data. The overlap between the dashed, red curves and the blue ones in Figures 3.3 and 3.6 underscores some limits of the identified model to approximate the variations present in real-world growth dynamics. This discrepancy is not surprising, as dealing with real, field-collected data with their intrinsic higher variability nature is much more difficult compared to the controlled nature of synthetic data. Consequently, for practical applications, it will be crucial to investigate in detail the theoretical properties of the crop growth model, for instance the robustness to account for the

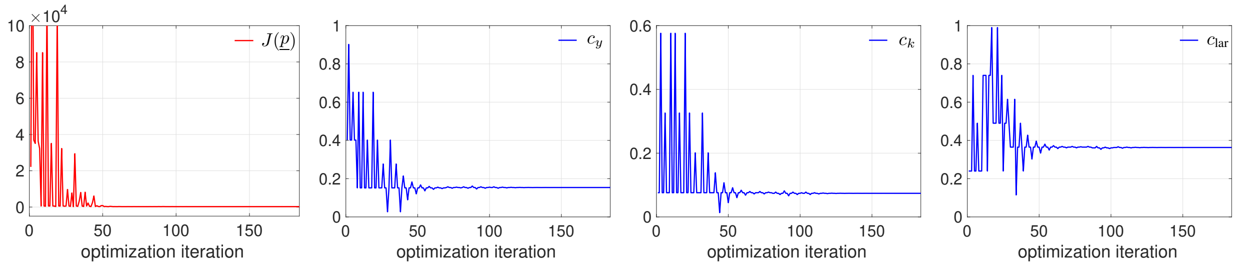


Figure 3.5 – Trends of the cost function J and of the estimated parameters in $\underline{p}$ at the various iterations of the optimization procedure based on real data.

inherent variability and uncertainties of real-world growth processes. In other words, we expect that a better field-data-based parameter estimation is possible.

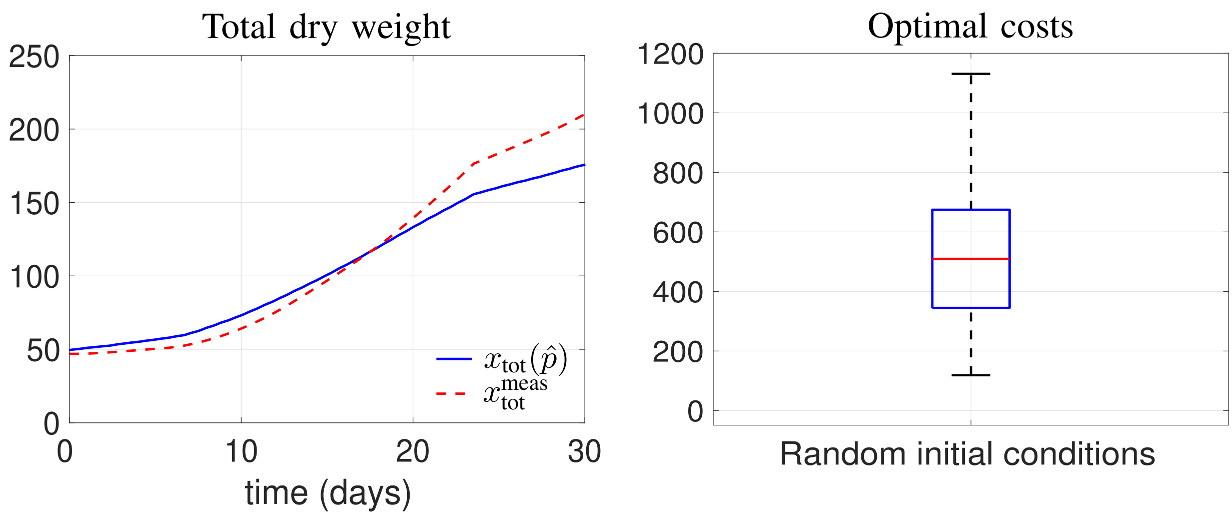


Figure 3.6 – Example of behavior of the total dry weight compared to a reference trajectory taken from the dataset [4] and boxplot of the optimal costs computed over the various trajectories.

3.5.2 Black-Box Model

This section assesses the data-driven crop growth modeling approach. After validating the capability of the black-box model 3.16 to approximate crop growth dynamics using synthetic data, we used again the dataset [4] of real measurements of the dry weight for a complete and comprehensive assessment of performances. All tests were carried out on a computer equipped with Apple M1 CPU, 16 GB of RAM, and a Matlab R2022b version.

3.5.2.1 Results with a Synthetic Dataset

In this section, we test the effectiveness of the proposed data-driven, black-box modeling approach to forecast the crop growth in two case studies. we generated the pairs $\left(\underline{\tilde{X}}_n^{(l)}, \underline{X}_{n+1}^{(l)}\right)$, $l = 1, \ldots, L$, synthetically through simulations starting from the model (3.12) introduced in Section 3.2, with the goal of first verifying the feasibility of the idea of approximating crop growth from data, and then of using (3.12) as a reference to check the accuracy of the black-box model. Hence, in order to collect the data pairs $\left(\underline{\tilde{X}}_n^{(l)}, \underline{X}_{n+1}^{(l)}\right)$, $l = 1, \ldots, L$, we numerically solved offline the ODE (3.12) in the interval $[0, T]$ for different initial conditions and system inputs. The value of T was chosen

equal to $5 \cdot 10^6$ seconds, i.e., about 57 days (such a large value was motivated by the slowness of the dynamics of the dry mass production). In particular, after performing a discretization of the interval into $N = 100$ time steps, we used a fourth-order Runge-Kutta method [180] for the numerical solution of (3.12). In other words, a sampling time equal to $\Delta t = 5 \cdot 10^4$ seconds was chosen for the temporal discretization. We selected a total of $G = 1000$ different initial conditions $\underline{X}_0^{(g)}$, $g = 1, \ldots, G$, randomly extracted according to uniform distributions in the ranges $[2, 3]$ and $[0.5, 1.5]$ for the dry weight and the non-structural dry weight, respectively.
Concerning the system inputs collected in the vector $\underline{U}_n$, $n = 0, 1, \ldots, N-1$, i.e., temperature, carbon dioxide concentration, and photo-synthetically active radiation, we adopted the approach detailed in [83] to generate a set of significant values. It consists in first generating an uncorrelated, zero-mean white noise w_n, $n = 0, 1, \ldots, N-1$, and then in transforming it into time sequences to cover a wide range of input amplitudes and frequencies, as follows:

$$u_n = \alpha_{t,1} + \alpha_{t,2} w_n, \tag{3.22}$$

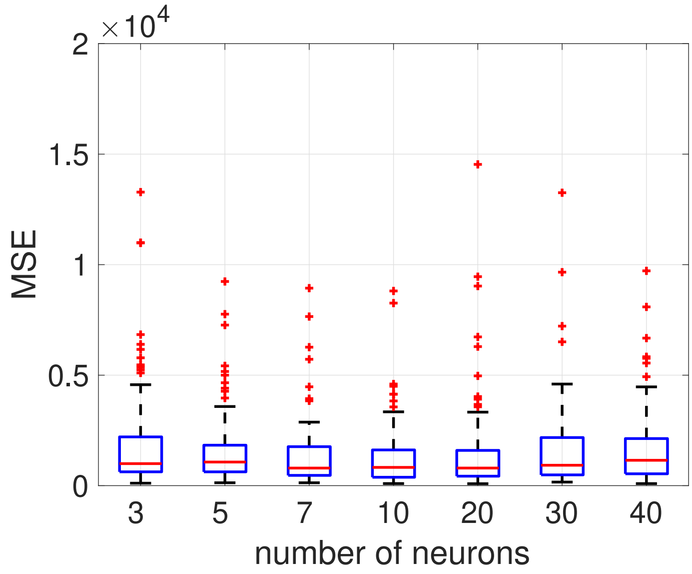
$$u_{\text{co2},n} = \alpha_{\text{co2},1} + \alpha_{\text{co2},2} w_n, \tag{3.23}$$

$$u_{\text{par},n} = \alpha_{\text{par},1} + \alpha_{\text{par},2} w_n, \tag{3.24}$$

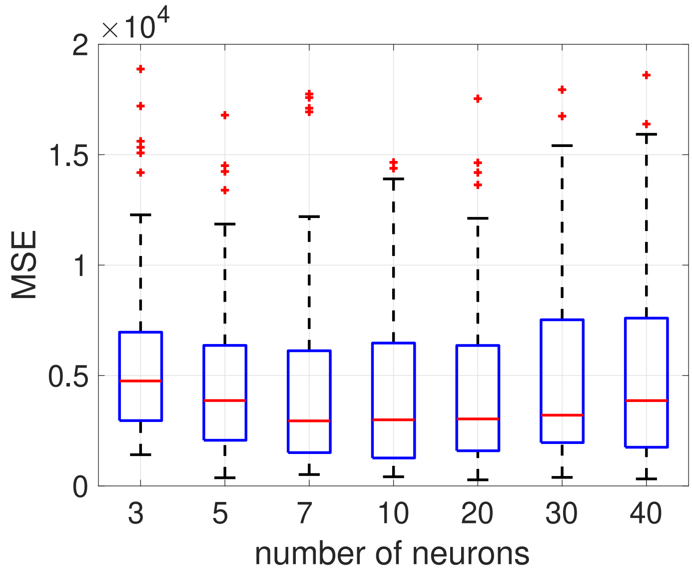
for $n = 0, 1, \ldots, N-1$, where $\alpha_{t,1}$, $\alpha_{\text{co2},1}$, and $\alpha_{\text{par},1}$ are the average values of the input signals u_n, $u_{\text{co2},n}$, and $u_{\text{par},n}$, respectively, while $\alpha_{t,2}$, $\alpha_{\text{co2},2}$, and $\alpha_{\text{par},2}$ are the amplitude of the variations around the averages. The numerical values of such parameters are reported in Table 3.1 along with the other parameters included in (3.12). Clearly, also the values of the system inputs were discretized: overall, we generated $L = G \cdot N = 10^5$ discrete input values $U_n^{(l)}$, $l = 1, \ldots, L$.
Then, we solved G times the discretized model (3.12) with the aforementioned random initial conditions and system inputs, and collected all the state values in the vector $\underline{X}_n^{(l)}$ for $n = 0, 1, \ldots, N$ and $l = 1, \ldots, L$ in order to construct the data pairs $\left(\underline{\tilde{X}}_n^{(l)}, \underline{X}_{n+1}^{(l)}\right)$ for the training. The data set made up of $L = 100000$ pairs was then split into a training set including 70000 elements, a validation set of 15000 elements, and a test set composed of 15000 pairs, i.e., 700, 150, and 150 different initial conditions, respectively. All the simulations were carried out using Matlab on an Intel Core i9 CPU with clock frequency equal to 3.5 GHz and 64 GB of RAM. The training was performed using the Matlab *trainlm* function, which provides an efficient implementation of the Levenberg-Marquardt algorithm [78]. To have a comprehensive evaluation, we considered one-hidden-layer feedforward neural networks with different numbers ν of neurons varying in the set $\{3, 5, 7, 10, 20, 30, 40\}$ to obtain different levels of approximation.
Two rounds of simulations were performed, corresponding to different values of the noise w_n used for the generation of the input vector $\underline{U}_n$ at the various time steps. In the first round, denoted as “Case A” in the following, we chose w_n as a white noise uniformly distributed in the range $[-0.5, +0.5]$, while in the second round, referred to as “Case B”, we focused on a white noise with uniform distribution in the range $[-1, +1]$.
The accuracy of the black-box model was measured by means of a mean square error (MSE) indicator accounting for the difference between the structural and non-structural dry weights obtained by (3.12) and (3.16) over the test set. More specifically, we define the following performance indicator:

$$\text{MSE} = \frac{1}{H} \frac{1}{N} \sum_{h=1}^{N} \sum_{n=1}^{N} \left\| \underline{X}_n^{(h),\text{model}} - \underline{X}_n^{(h),\text{black box}} \right\|^2, \tag{3.25}$$

where $H = 150$ is the number of different initial conditions within the test set, $\|\cdot\|$ is the Euclidean norm, and $\underline{X}_n^{(h),\text{model}}$ and $\underline{X}_n^{(h),\text{black box}}$, $h = 1, \ldots, H$, $n = 0, 1, \ldots, N-1$, are the state vectors of the model (3.12) and of the black box one (3.16), respectively, for the h-th initial condition on the test set at the time step n.



(Case A)



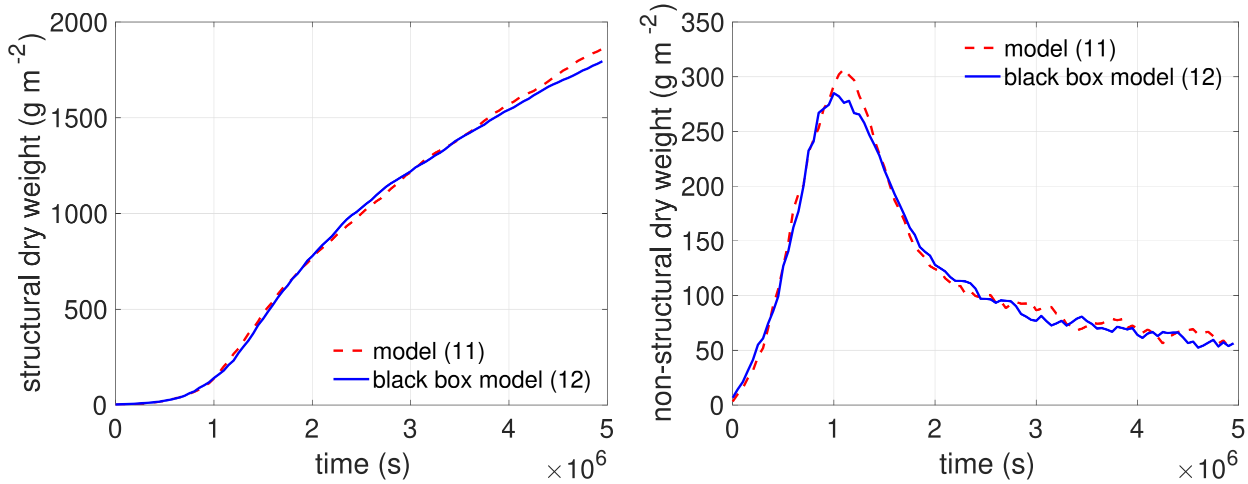
(Case B)

Figure 3.7 – Boxplots of the MSE performance indicator obtained by one-hidden-layer feedforward neural networks with different computational units in Case A (top) and Case B (bottom).

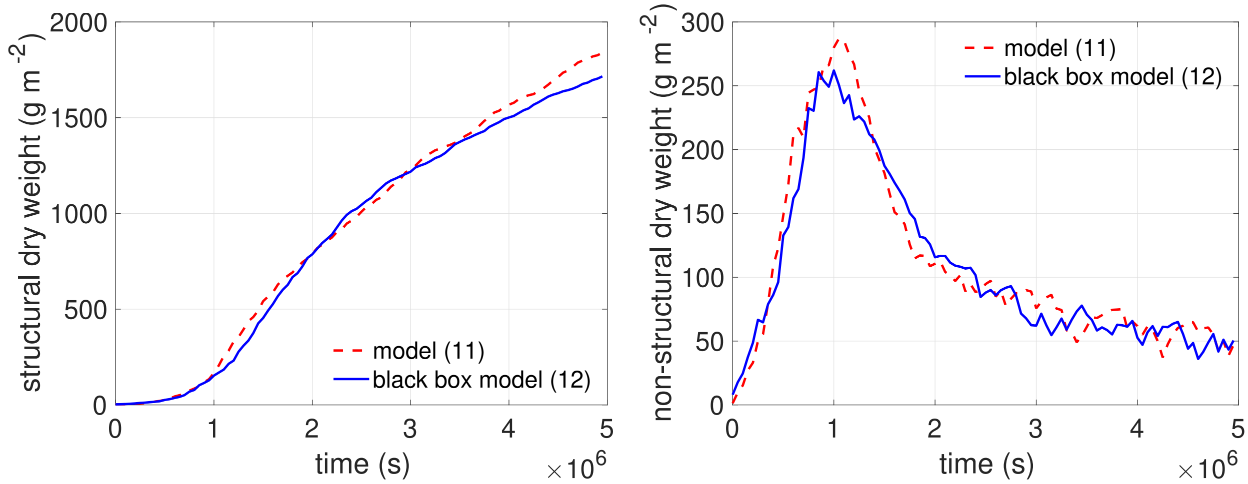
Figure 3.7 contains the boxplots of the MSE (3.25) for the considered numbers of neurons in both Case A and Case B, while Table 3.2 reports its average values and standard deviations. It turns out that in Case A the average error between the output of the model (3.12) and the output of the black box one (3.16), as well as its dispersion, is reduced as compared to Case B. Both in Case A and Case B, the best performances are obtained for $\nu = 7$ and $\nu = 10$, while lower and larger values of ν are characterized by a small decay of accuracy. The larger values of the MSE for smaller numbers of computational units are due to the poor approximating capability of the neural networks due to too small size of hidden units, while the decay for larger number of neurons is due to overfitting. As expected, the lower the values of the noise w_n, $n = 0, 1, \ldots, N$, used to generate the system inputs, the better the approximation, as neural networks have to learn less-varying input signals. Figure 3.8 reports the result of a randomly-extracted simulation from the test set, obtained with $\nu = 10$ neurons. In more detail, both the structural and non-structural dry

Table 3.2 – Average values and standard deviations of the MSE performance indicator in the considered case studies.

Num. of	Case A		Case B	
neurons	avg	std dev	avg	std dev
$\nu=3$	$2.12 \cdot 10^3$	$3.04 \cdot 10^3$	$5.87 \cdot 10^3$	$4.16 \cdot 10^3$
$\nu=5$	$1.59 \cdot 10^3$	$1.64 \cdot 10^3$	$5.40 \cdot 10^3$	$5.45 \cdot 10^3$
$\nu=7$	$1.38 \cdot 10^3$	$1.51 \cdot 10^3$	$4.58 \cdot 10^3$	$4.71 \cdot 10^3$
$\nu=10$	$1.34 \cdot 10^3$	$1.49 \cdot 10^3$	$4.62 \cdot 10^3$	$5.14 \cdot 10^3$
$\nu=20$	$1.48 \cdot 10^3$	$2.13 \cdot 10^3$	$5.45 \cdot 10^3$	$7.51 \cdot 10^3$
$\nu=30$	$1.69 \cdot 10^3$	$1.97 \cdot 10^3$	$5.74 \cdot 10^3$	$6.51 \cdot 10^3$
$\nu=40$	$1.66 \cdot 10^3$	$1.77 \cdot 10^3$	$5.96 \cdot 10^3$	$6.40 \cdot 10^3$



(Case A)



(Case B)

Figure 3.8 – Plot of a trajectory of the structural dry weight (left) and the non-structural dry weight (right), randomly-extracted from the test set, obtained by using the model (3.12) and the black-box model (3.16) with $\nu = 10$ computational units in Case A (top) and Case B (bottom).

weights obtained by the model (3.12) and the black box one (3.16) are reported in Case A and Case B. First of all, we point out that obtained curves are consistent with the biological behavior of the structural and non-structural dry weights according to [83]. Then, we observe the good approximation capability of the black-box model, which is able to reproduce the behavior of the model (3.12) quite accurately. Once again, we note the better approximation capability of the black-box model for a limited variation of the system inputs.

Table 3.3 – Average computational time (seconds) for each simulation in the test set for the considered case studies.

Num. of	Case A		Case B	
neurons	model (3.12)	black-box model (3.16)	model (3.12)	black-box model (3.16)
$\nu = 3$		$8.59 \cdot 10^{-4}$		$8.80 \cdot 10^{-4}$
$\nu = 5$		$8.66 \cdot 10^{-4}$		$8.78 \cdot 10^{-4}$
$\nu = 7$		$8.69 \cdot 10^{-4}$		$8.94 \cdot 10^{-4}$
$\nu = 10$	$6.11 \cdot 10^{-3}$	$8.76 \cdot 10^{-4}$	$5.95 \cdot 10^{-3}$	$9.00 \cdot 10^{-4}$
$\nu = 20$		$8.95 \cdot 10^{-4}$		$9.27 \cdot 10^{-4}$
$\nu = 30$		$9.15 \cdot 10^{-4}$		$9.40 \cdot 10^{-4}$
$\nu = 40$		$9.38 \cdot 10^{-4}$		$9.67 \cdot 10^{-4}$

Lastly, Table 3.3 contains the average time needed to complete a simulation in the two case studies, i.e., the time needed to compute the structural and non-structural dry weights from the discrete time step 0 to N. We observe that the black-box model (3.16) is characterized by a computational time that is one order of magnitude less than the one of the model (3.12). This is due to the fact that, as pointed out in Section 3.4, no numerical solution of an ODE is required for the black-box model, as the output of the neural network is directly equal to the state of the system at the next discretization instant, without the need of performing a numerical integration. We also observe that the computational requirements of the black-box model are almost constant with respect to the number ν of neurons.

3.5.2.2 Results with a Dataset of Real-World Measurements

After validating the capability of the black-box model (3.16) to approximate crop growth dynamics using synthetic data, we used the dataset [4] of real measurements of the dry weight again for a complete and comprehensive assessment of performances. The type and number of data, along with interpolation and smoothing, were the same considered for the parameter identification described in Section 3.5.1. We considered again one-hidden-layer feedforward neural networks with different numbers ν of neurons varying in the set $\{3, 5, 7, 10, 20, 30, 40\}$ to obtain different levels of approximation. The accuracy of the black-box model was reassessed by means of a MSE indicator quantifying the discrepancy between the total dry weight of the real data belonging to the test set and the one provided by the black-box model (3.16).
An example trajectory for the total dry weight (dashed, red line) and the corresponding estimate from the black-box model (blue line) are shown in Fig. 3.9. Similarly to the case of the white-box identification with real data, we observe that the two trajectories satisfactorily overlap, confirming the effectiveness of the black-box model to approximate the crop growth, also when considering experimental data. This effectiveness is once again confirmed by the boxplots of the MSE over the available test data (see the second plot in Fig.. 3.9). In more detail, we observe that the best results are obtained for $\nu = 10$ and $\nu = 40$ computational units. Compared with the boxplots obtained from the training using synthetic data (see Fig. 3.7), the MSE, in this case, presents a greater variability. This behavior may be ascribed to the limited amount of available data used

to train the black-box model. In fact, the dataset from [4] contains only a small number of data points from just six different experimental setups, which may not fully capture the variability of crop growth under different environmental conditions. Consequently, neural networks may not have learned all possible growth patterns. However, achieving satisfactory approximations of the crop growth despite the limited amount of data is encouraging. We expect that much better results could be obtained by using a larger dataset, for instance coming from measurement performed on a prototype of the AVF concept. Nevertheless, the best median value of the MSE indicator is given by 314.48, and it is obtained with $\nu = 7$ neurons. If we compare such result with the MSE of the white-box model with real data, we observe that they are very close to each other, thus confirming the effectiveness of the black-box model.

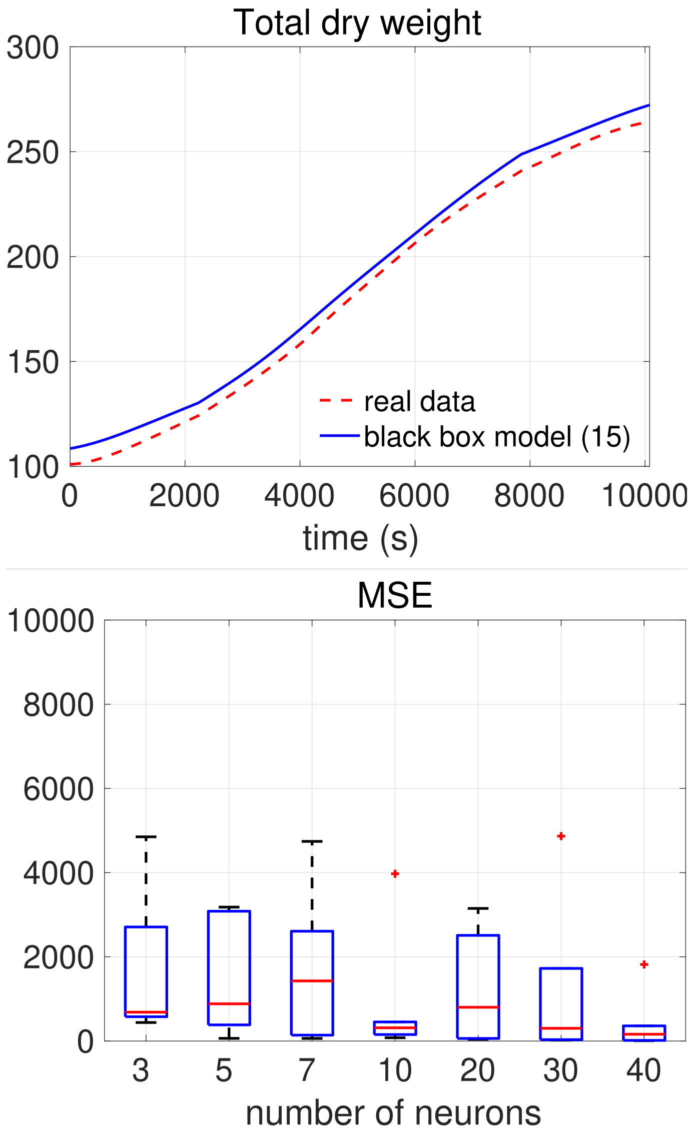


Figure 3.9 – Example of behavior of the total dry weight obtained by the black-box model (3.16) with $\nu = 10$ computational units compared to a reference trajectory taken from the dataset [4] and boxplot of the MSE over the available trajectories.

3.6 Conclusion

In this chapter, we have investigated lettuce growth dynamics in an Adaptive Vertical Farm (AVF) using both a two-state model for structural and non-structural dry mass, requiring precise parameter tuning, and a data-driven, black-box model based on a feedforward neural network. The key parameters of the two-state model were estimated by minimizing the least-squares difference between measured and predicted states over time. Numerical results using synthetic and real-world dataset showed that while some parameters could not be accurately identified from available measurements, others could be, and the estimated parameters effectively reconstructed the crop behavior, as confirmed by real-world data. These findings emphasize the importance of a careful analysis of the model's theoretical properties and a sensitivity analysis to ensure full parameter estimation from field measurements. In addition, we presented a data-driven model of crop growth based on a one-hidden-layer feedforward neural network, using data from a continuous-time dynamic growth model for lettuce cultivation and the same real-world dataset as dor the white-box model. The data-driven model demonstrated satisfactory accuracy and significantly reduced computational time, making it highly suitable for integration into optimization algorithms. These algorithms aim to determine the optimal scheduling of seedings in the multiple shelves of the greenhouse, ultimately maximizing production yield. Thus, the combination of both model types offers a robust framework for efficient crop growth prediction and effective AVF management.

CHAPTER 4

Unmanned Aerial Vehicles (UAVs) in the Adaptive Vertical Farm

Contents

4.1 Introduction

With increasingly sophisticated technology and onboard sensors, UAVs are becoming a key tool for many companies and consultants around the world, from the transportation sector to security and surveillance of industrial facilities, maintenance and inspection of industrial sites and infrastructure, agriculture, etc. Their flexible and highly maneuverable nature allows them to dynamically survey areas and automatically acquire data, thanks to the wide range of sensors that can be loaded on board. One of their major uses besides video surveillance concerns precision agriculture, mainly in the open field, but indoor applications in greenhouses have begun to appear in recent years. Using drones for precision agriculture offers a number of benefits. For one, it allows farmers to monitor crops in real-time and identify problems before they become widespread. Regarding the fast-growing Vertical Farming technology in agriculture, UAVs can also cover large areas quickly and efficiently, reducing the amount of time and resources needed for manual inspections. In addition, the data collected by drones can be used to optimize crop yields, improve soil health, and reduce environmental impact [136]. To this end, as mentionned in Chapter 1, quadcopters tend to be used for their straightforward mechanical design, quick mobility, and higher cargo capacity compared to other drones. However, such UAVs have a nonlinear coupled dynamic behavior being an underactuated system. In this respect, nonlinear state observer play a crucial role in quadrotor control when it may not be possible to directly measure some state variables due to the high cost of data acquisition systems, for instance, based on available measurements and dynamics of the system [97, 145, 160]. Furthermore, the investigation of a state estimation scheme is also motivated by several indoor operation challenges for the UAV in the AVF [75, 143]. First, indoor environments typically lack a clear line of sight to GPS satellites, making GPS-based position estimation unreliable. An observer is essential for accurately estimating position and altitude, with altitude being a crucial parameter for navigation and monitoring, especially in environments with a strong vertical orientation. Additionally, operating near surfaces like walls and crops can introduce disturbances and sensor noise, which an observer can help filter out to provide more reliable state estimates. The limited space in indoor environments further emphasizes the importance of accurate state estimation to avoid collisions with obstacles and navigate in confined spaces. Changes in lighting conditions can affect vision-based sensors, but an observer can combine data from multiple sensors to improve robustness. Noise and vibrations from machinery or equipment, which impact accelerometers and gyros, can also be mitigated by an observer to maintain accurate state estimates. Moreover, temperature and humidity variations in indoor settings can affect sensor performance, but an observer can adapt to these environmental changes to ensure accurate estimation. Finally, the dense crop structure in vertical farms poses challenges for vision-based systems, but an observer can help navigate through complex environments by fusing information from various sensors. We propose in this chapter, a nonlinear observer design for the UAV operationg in the AVF, being a confined space with potential disturbances such as the variable artificial lighting conditions, close proximity to surfaces or crop density, as mentionned previously. Hence, an accurate and reliable state estimation is crucial for ensuring the UAV's effective and safe operation in the AVF. In this work, we consider the ETH Zurich OS4 quadrotor model proposed by [38, 39] being a versatile and efficient tool that aligns well with the operational needs of an AVF, mainly for the experiment phase of improving the existant prototype. The ETH Zurich OS4 has a compact size with control capabilities, and potential for embedding other technologies make it a fitting choice for tasks such as crop monitoring, environmental sensing, and data collection.

The chapter 4 is organized as follow, Section 4.2 introduces the LMI $\mathcal{H}\infty$-based observer design, which leverages Linear Matrix Inequalities (LMI) and the $\mathcal{H}\infty$ criterion to ensure accurate state estimation under challenging conditions. This design guarantees robust performance by minimizing the estimation error and providing a solid foundation for the observer's reliability in the presence of uncertainties. Building upon this, Section 4.3 proposes a novel observer design approach using an output-based dynamic extension. The section is divided into two subsections: Section 4.3.1 explores the observer design in the continuous-time domain, outlining the steps and equations necessary for implementing the design. Section 4.3.2 focuses on the discrete-time case, where the observer design process is adapted to enhance computational efficiency and simplify implementation, making it more suitable for real-time applications. Section 4.4 is devoted to the simulation results highlighting the effectiveness of the proposed observer designs. It is divided into three parts: Section 4.4.1 provides the mathematical model of the UAV used in the simulations, offering a clear representation of the system dynamics. Section 4.4.2 presents the results of the LMI $\mathcal{H}_\infty$-based observer design, demonstrating its robustness and performance in state estimation. Lastly, Section 4.4.3 discusses the results of the extended dynamic output observer in the discrete-time case, showcasing its advantages and accuracy in handling the UAV's dynamics.

4.2 LMI $\mathcal{H}_\infty$ - Based Observer Design for the UAV

In order to enhance the robustness of state estimation for UAVs operating in dynamic environments, we propose an $\mathcal{H}_\infty$ observer design via Linear Matrix Inequalities (LMI). This design ensures accurate state estimation in the presence of disturbances and noise, minimizing the worst-case estimation error (the maximum possible error in the state estimation under the most challenging conditions). By applying a new, less conservative LMI condition, we solve the $\mathcal{H}_\infty$ circle criterion design, improving observer performance and robustness under uncertain conditions.
We consider the following class of systems, defined in 2.57 :

$$\begin{cases} \dot{x} = \Gamma(x) \\ y = Cx \end{cases} \tag{4.1}$$

The LPV approach allows treatment of the nonlinearity with the best precision and exploits all the interesting properties of the nonlinearity of the investigated system. According to [7], before introducing the observer structure, it would be better to write $\Gamma(.)$ in a detailed form as:

$$\Gamma(x) \triangleq A + \Pi \begin{pmatrix} \Gamma_1(\Lambda_1 x) \\ \vdots \\ \Gamma_q(\Lambda_q x) \end{pmatrix} = A + \Pi \sum_{i=1}^{q} e_q(i)\Gamma_i(\Lambda_i x) \tag{4.2}$$

where $A \in \mathbb{R}^{n\times n}, \Pi \in \mathbb{R}^{n\times q}$ and $\Lambda_i \in \mathbb{R}^{n_i\times n}$ are constant matrices, and q is the number of the nonlinear components of the system. Therefore, we consider the following generalized observer [22], [175], defined by the equations:

$$\dot{\hat{x}} = A + \sum_{i=1}^{q} \Pi_q(i)\Gamma_i(\vartheta_i) + L(y - C\hat{x}) \tag{4.3a}$$

$$\vartheta_i^j = \Lambda_i^j \hat{x} + K_i^j (y - C\hat{x}) \tag{4.3b}$$

where ϑ_i^j is the j^{th} component of $\vartheta_i, j = 1, \ldots, n_i$. The matrices $L \in \mathbb{R}^{n\times p}$ and $K_i^j \in \mathbb{R}^{1\times p}$ are the observer parameters. The dynamics equation of the estimation error $e = x - \hat{x}$ is then given by

$$\dot{e} = A + \left(\sum_{i=1}^{q} \sum_{j=1}^{jn_i} \left[\gamma_{ij} \Pi e_q(i) \left(\Lambda_i^j - K_i^j C \right) \right] - LC \right) e \tag{4.4}$$

We need to define the matrix $\Psi := (\Psi_{ij})_{ij}$, where

$$\Psi_{ij} = \begin{cases} \gamma_{ij}, i = 1, \ldots, q, j \leq n_i \\ 0, i = 1, \ldots, q, j > n_i \end{cases} \tag{4.5}$$

and the affine matrix function:

$$\mathcal{A}(\Psi) = A + \sum_{i=1}^{q} \sum_{j=1}^{n_i} \Psi_{ij} \Pi e_q(i) \left(\Lambda_i^j - K_i^j C \right) \tag{4.6}$$

Hence, the estimation error can be reduced to

$$\dot{e} = (\mathcal{A}(\Psi) - LC)e \tag{4.7}$$

Proceeding as in the previous section, from condition (2.65), it follows that the parameter Ψ belongs to a bounded convex set $\mathcal{H}_{q,\bar{n}}$ for which the set of vertices is defined by

$$\mathcal{V}_{\mathcal{H}_{q,\bar{n}}} = \left\{ \Phi \in \mathbb{R}^{q\times \bar{n}} : \begin{cases} \Phi_{ij} \in \left\{ \underline{\gamma}_{\gamma_{ij}}, \bar{\gamma}_{\gamma_{ij}} \right\}, j \leq n_i \\ \Phi_{ij} = 0, j > n_i \end{cases} \right\} \tag{4.8}$$

where $\bar{n} = \max_i(n_i)$. By applying the result of Theorem 2.3.13, more general LMI conditions summarized in the following theorem are obtained.

Theorem 4.2.1 ([7]). *Let define:*

$$\mathfrak{P} \triangleq \{(i,j) : i = 1, \dots, q; j = 1, \dots, n_i\}.$$

and:

$$\eta \triangleq \text{card}(\mathfrak{P}) = \sum_{i=1}^{i=q} n_i$$

We define also the following matrix :

$$\mathbb{A}_{ij}^{\mu} \triangleq A + \eta \left[\mu \Pi_{e_q(i)} \left(\Lambda_i^j - K_i^j C\right)\right] \quad \forall \mu \in \{\gamma_{\gamma_{ij}}, \bar{\gamma}_{\gamma_{ij}}\}, \forall (i,j) \in \mathfrak{P}. \tag{4.9}$$

The observer (4.3) is asymptotically convergent if the two following conditions are satisfied:

- *If $\mathfrak{P} = \varnothing$, there exists a positive definite matrix $\mathcal{X}$ and a matrix $\mathcal{Y}$ of appropriate dimensions so that the following LMI holds:*

$$A^T \mathcal{X} + \mathcal{X} A - C^T \mathcal{Y} - \mathcal{Y}^T C < 0. \tag{4.10}$$

- *If $\mathfrak{P} \neq \varnothing$, there exist a positive definite matrix $\mathcal{P}$, some matrices $\mathcal{R}_{ij}, K_i^j, (i,j) \in \mathfrak{P}$ of appropriate dimensions so that the following matrix inequalities hold:*

$$\left(\mathbb{A}_{ij}^{\mu}\right)^T \mathcal{P} + \mathcal{P} \mathbb{A}_{ij}^{\mu} - C^T \mathcal{R}_{ij} - \mathcal{R}_{ij}^T C < 0; \quad \forall \mu \in \left\{\underline{\gamma}_{\gamma_{ij}}, \bar{\gamma}_{\gamma_{ij}}\right\}, \quad \forall (i,j) \in \mathfrak{P} \tag{4.11}$$

Then, the observer gains K_i^j are free solutions of (4.11), and the gain L is given by

$$L = \begin{cases} \mathcal{X}^{-1} \mathcal{Y}^T, \ \text{if } \mathfrak{P} = \varnothing \\ \frac{1}{\eta} \sum_{(i,j) \in \mathfrak{P}} \mathcal{P}^{-1} \mathcal{R}_{ij}^T, \ \text{if } \mathfrak{P} \neq \varnothing \end{cases} \tag{4.12}$$

Remark 4.2.2. *Notice that in (4.11) there are two inequalities to be solved for each $(i,j) \in \mathfrak{P}$. We refer the reader to [7] for more details.*

The main result is formulated in the following theorem applied to the system (4.1).

Theorem 4.2.3 ([43]). *If there exist some symmetric positive definite matrices $\mathcal{P} \in \mathbb{R}^{n \times n}$ and R_{ij}, K_i^j, $(i,j) \in \mathfrak{P}$ of appropriate dimensions so that the following convex optimization is solvable* $\min(\varrho)$ *subject to*

$$\begin{bmatrix} (\mathbb{A}_{ij}^{\mu})^T \mathcal{P} + \mathcal{P} \mathbb{A}_{ij}^{\mu} - C^T \mathcal{R}_{ij} - \mathcal{R}_{ij}^T C + \mathbb{I}_n & \mathcal{P} E \\ E^T \mathcal{P} & -\varrho \mathbb{I}_m \end{bmatrix} < 0$$
$$\forall \mu \in \left\{\underline{\gamma}_{\gamma_{ij}}, \bar{\gamma}_{\gamma_{ij}}\right\} \tag{4.13}$$

then, the $\mathcal{H}_\infty$ criterion

$$\|e\|_{\ell_2^n} \leq \sqrt{\varrho \|w\|_{\ell_2^q}^2 + \nu \left\|e_0\right\|^2}$$

is satisfied, where $e = x - \hat{x}$ denotes the estimation error, $\varrho > 0$ the disturbance attenuation level representing the disturbance gain from w to e, and $\nu > 0$ such that $\nu = \lambda_{\max} P$. Therefore, the observer gain L is given by

$$L = \frac{1}{\eta} \sum_{(i,j) \in \mathfrak{P}} \mathcal{P}^{-1} \mathcal{R}_{ij}^T. \tag{4.14}$$

Proof. The proof is omitted as it is technical and partially inspired from [7].

□

Clearly, the maximization of ϱ is well-suited to reduce the gain L. To achieve this, it is necessary to maximize the variables $\mathcal{R}_{ij}, (i,j) \in \mathfrak{P}$ by the following procedure. We choose a fixed α such that

$$L^T L \leq \alpha I, \tag{4.15}$$

so that

$$\mathcal{R}_{ij}\mathcal{P}^{-1}\mathcal{P}^{-1}\mathcal{R}_{ij}^T \leq \alpha I$$

We suppose that

$$\mathcal{P} > \mathbb{I}_n, \quad -\alpha I + \mathcal{R}_{ij}\mathcal{R}_{ij}^T \leq 0. \tag{4.16}$$

Then, the optimization is based on imposing the LMIs

$$\mathcal{P} > \mathbb{I}_n$$

$$\begin{bmatrix} -\alpha\mathbb{I}_p & R_{ij} \\ R'_{ij} & -\mathbb{I}_n \end{bmatrix} < 0$$

4.3 Observer Design by Using an Output-Based Dynamic Extension

This section presents the observer design methodology based on a novel output-based dynamic extension. The design is approached in two distinct time domains: continuous-time and discrete-time. In the first Subsection 4.3.1, we explore the observer design for the continuous-time case, where theoretical foundations and robustness criteria are derived using a continuous-time system model. This part provides a comprehensive understanding of the dynamics and performance guarantees for state estimation. In the second Subsection 4.3.2, the focus shifts to the discrete-time case, where the observer design is adapted to suit practical, real-time implementation in systems such as UAVs operating in AVFs. This section highlights how the continuous-time observer principles are translated into discrete-time, ensuring that the design aligns with the system's sampling nature and computational constraints.

4.3.1 Observer design for the Continuous-Time Case

We consider nonlinear systems in the form

$$\begin{cases} \dot{x} = \psi(x, w) \\ y = \phi(x, v) \end{cases} \tag{4.17}$$

where $x \in \mathbb{R}^n$ is the system state and $y \in \mathbb{R}^p$ is the vector of output measurements of the system. $w \in \mathbb{R}^s$ and $v \in \mathbb{R}^q$ are unknown external disturbances. The functions $\psi(\cdot,\cdot)$ and $\phi(\cdot,\cdot)$ are assumed to be Lipschitz-continuous, with respect to x, on $\Omega \subseteq \mathbb{R}^n$, where Ω is a positively invariant compact and convex set of the system (4.17). Then, the functions $\psi(\pi_\Omega(\cdot), w)$ and $\phi(\pi_\Omega(\cdot), v)$ are globally Lipschitz in $\mathbb{R}^n$, where $\pi_\Omega(x)$ is the *Hilbert* projection of x on the convex set Ω.
Usually in the literature, we propose the following state observer:

$$\dot{\hat{x}} = \psi(\hat{x}, 0) + \mathbb{L}\Big(y - \phi(\hat{x}, 0)\Big), \tag{4.18}$$

where $\hat{x} \in \mathbb{R}^n$ is the estimate of the system state x and $\mathbb{L} \in \mathbb{R}^{n\times p}$ is the constant observer gain to be determined such that the estimation error $e = x - \hat{x}$ converges exponentially towards zero. However, with this kind of classical and natural observer, the observer gain is multiplied by the output nonlinearity and then by the disturbance in the measurement. This coupling between the observer gain and the output measurement can amplify the effect of the measurement noise. In this section, the main objective consists in proposing a different state observer for the system (4.17). We will propose a filtered-like output based observer, which allows both avoiding the output nonlinearity and disturbances to be explicitly multiplied by the observer gain.

4.3.1.1 Output-based dynamic extension for continuous-time system

The initial system is given by the equations (4.17). We perform a change of output variable by introducing the following system:

$$\begin{cases} \dot{\eta}(t) = \alpha y(t) \\ \eta(0) = \eta_0 \text{ (known)} \end{cases} \tag{4.19}$$

Since η_0, α, and $y(t)$ are known, then the state $\eta(t)$ may be considered as the new output measurement generated from the measurement $y(t)$. Therefore, we consider $\boldsymbol{y}(t) = \eta(t)$ as the new output, expressed in terms of the extended state vector ξ as follows:

$$\boldsymbol{y}(t) = \eta(t) = \overbrace{\begin{bmatrix} \mathbb{I}_p & 0 \end{bmatrix}}^{C} \xi, \text{ with } \xi(t) \triangleq \begin{bmatrix} \eta(t) \\ x(t) \end{bmatrix} \tag{4.20}$$

where $\mathbb{I}_p$ is the identity matrix of dimension p. Let us define also the vector $\boldsymbol{\omega}$ which combines the w and v as follows:

$$\boldsymbol{\omega}(t) \triangleq \begin{bmatrix} w(t) \\ v(t) \end{bmatrix}.$$

This allows us to transform the equations of the original system into a new system with the state variable ξ:

$$\begin{cases} \dot{\xi} = f(\xi, \boldsymbol{\omega}) \triangleq \begin{bmatrix} \alpha\phi(x, v) \\ \psi(x, w) \end{bmatrix} \\ \boldsymbol{y} = \eta = C\xi \end{cases} \tag{4.21}$$

This transformed system incorporates the relationships between the output vector $\boldsymbol{y}$ and the system states represented by ξ. Note that the dynamics of the states are now given by $f(\xi, \boldsymbol{\omega})$, which is a function of the generalized state ξ and extended disturbance vector $\boldsymbol{\omega} \in \mathbb{R}^{n_w}$, where $n_w = s + q$. The state observer we will consider in this section corresponds to the system (4.21) with linear outputs. Then, the observer synthesis and the Lyapunov analysis become simpler. The new output $\boldsymbol{y}(t)$ plays the role of a filter which makes it possible to filter the disturbances or the measurement noise.

Before tackling the observer design problem, we recall the following lemma:

Lemma 4.3.1 ([178]). *Consider a function $\varphi : \mathbb{R}^n \longrightarrow \mathbb{R}^p$. Assume that φ is γ_φ−Lipschitz on a nonempty subset $\Omega \subseteq \mathbb{R}^n$, with γ_{φ_i} the Lipschitz constant of each component φ_i of φ. Then, there exist*

- *a Lipschitz extension $\breve{\varphi} \triangleq \varphi \circ \pi_\Omega : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ of φ, with $\breve{\varphi}(x) = \varphi(x), \forall x \in \Omega$;*
- *functions $\varphi_{ij} : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$*
- *constants $\underline{\gamma}_{\varphi_{ij}}$ and $\bar{\gamma}_{\varphi_{ij}}$*

such that for all $X, Y \in \mathbb{R}^n, X \neq Y$, we have

$$\breve{\varphi}(X) - \breve{\varphi}(Y) = \sum_{i=1}^{p} \sum_{j=1}^{n} \varphi_{ij}(t) \mathcal{H}_{ij}^{p,n} \Big(X - Y \Big) \tag{4.22}$$

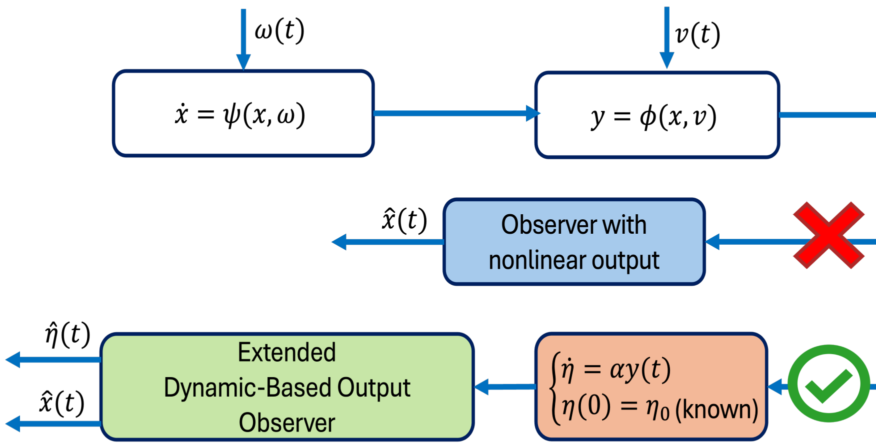


Figure 4.1 – Output-Based Dynamic Extension Technique for Continuous-time Systems

and

$$-\gamma_{\varphi_{ij}} \leq \underline{\gamma}_{\varphi_{ij}} \leq \varphi_{ij}(t) \leq \bar{\gamma}_{\varphi_{ij}} \leq \gamma_{\varphi_{ij}} \tag{4.23}$$

where

$$\varphi_{ij}(t) \triangleq \varphi_{ij}\left(X^{Y_{j-1}}, X^{Y_j}\right), \ \mathcal{H}_{ij}^{p,n} \triangleq \mathrm{e}_p(i)\mathrm{e}_n^\top(j).$$

4.3.1.2 Observer Design

This section is devoted to the observer design method proposed in this work. In addition to the new observer structure and the output-based extended dynamics technique, we propose novel LMI-based synthesis procedure guaranteeing the Input-to-State (ISS) property of the estimation error. Instead of the standard state observer form (4.18), we propose a new observer structure by exploiting the extended dynamics (4.21). The observer is driven by the created output measurement $\boldsymbol{y}(t)$. This new output $\boldsymbol{y}(t)$ plays the role of a filter which makes it possible to filter the disturbances or the measurement noise. The estimation strategy is depicted in Figure 4.1.
The state observer form is described by the following equations:

$$\begin{cases} \dot{\hat{\xi}} = \check{f}(\hat{\xi}, 0) + \mathcal{L}\left(\boldsymbol{y} - C\hat{\xi}\right) \\ \hat{x} = \begin{bmatrix} 0 & \mathbb{I}_n \end{bmatrix} \hat{\xi} \end{cases} \tag{4.24}$$

where

$$\check{f}(\hat{\xi}, \boldsymbol{\omega}) = f\left(\pi_{\Omega_\xi}(\hat{\xi}), \boldsymbol{\omega}\right), \ \forall \boldsymbol{\omega} \in \Omega_{\boldsymbol{\omega}} \ni 0, \ \Omega_\xi \triangleq \mathbb{R}^p \times \Omega,$$

and $\mathcal{L} \in \mathbb{R}^{(n+p)\times p}$ is the constant observer gain to be computed such that the estimation error $\epsilon(t) \triangleq x(t) - \hat{x}(t)$ satisfies an exponential ISS bound to be determined later. To develop a synthesis method based on Lyapunov analysis, let us define the extended error vector $\epsilon_\xi(t) \triangleq \xi(t) - \hat{\xi}(t)$, which satisfies the following dynamics:

$$\dot{\epsilon}_\xi = \left[f(\xi, \boldsymbol{\omega}) - \check{f}(\hat{\xi}, 0)\right] - \mathcal{L}C\epsilon_\xi. \tag{4.25}$$

We decompose the nonlinear term as follows:

$$f(\xi, \boldsymbol{\omega}) - \check{f}(\hat{\xi}, 0) = \left(f(\xi, \boldsymbol{\omega}) - \check{f}(\hat{\xi}, \boldsymbol{\omega})\right) + \left(\check{f}(\hat{\xi}, \boldsymbol{\omega}) - \check{f}(\hat{\xi}, 0)\right). \tag{4.26}$$

Since $\hat{\xi} \in \Omega_\xi$, then we have $f(\xi, \boldsymbol{\omega}) = \check{f}(\xi, \boldsymbol{\omega})$. It follows that from Lemma 4.3.3, there exist function $f_{ij}^{\xi} : \mathbb{R}^{n_p} \times \mathbb{R}^{n_p} \longrightarrow \mathbb{R}$ where $n_p = n + p$ and constants $\underline{\gamma}_{f_{ij}^{\xi}}$ and $\bar{\gamma}_{f_{ij}^{\xi}}$ such that

$$f(\xi, \boldsymbol{\omega}) - \check{f}(\hat{\xi}, \boldsymbol{\omega}) = \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} f_{ij}^{\xi}(t) \mathcal{H}_{ij}^{n_p} \left(\xi - \hat{\xi} \right) \tag{4.27}$$

and

$$-\gamma_{f_{ij}^{\xi}} \leq \underline{\gamma}_{f_{ij}^{\xi}} \leq f_{ij}^{\xi}(t) \leq \bar{\gamma}_{f_{ij}^{\xi}} \leq \gamma_{f_{ij}^{\xi}}. \tag{4.28}$$

Similarly, there exist functions $f_{ij}^{\boldsymbol{\omega}} : \mathbb{R}^{n_w} \times \mathbb{R}^{n_w} \longrightarrow \mathbb{R}$ and constants $\underline{\gamma}_{f_{ij}^{\boldsymbol{\omega}}}$ and $\bar{\gamma}_{f_{ij}^{\boldsymbol{\omega}}}$ such that

$$\check{f}(\hat{\xi}, \boldsymbol{\omega}) - \check{f}(\hat{\xi}, 0) = \sum_{i=1}^{n_p} \sum_{j=1}^{n_w} f_{ij}^{\boldsymbol{\omega}}(t) \mathcal{H}_{ij}^{n_p, n_w} \boldsymbol{\omega}(t) \tag{4.29}$$

and

$$-\gamma_{f_{ij}^{\boldsymbol{\omega}}} \leq \underline{\gamma}_{f_{ij}^{\boldsymbol{\omega}}} \leq f_{ij}^{\boldsymbol{\omega}}(t) \leq \bar{\gamma}_{f_{ij}^{\boldsymbol{\omega}}} \leq \gamma_{f_{ij}^{\boldsymbol{\omega}}}. \tag{4.30}$$

Before tackling the Lyapunov analysis on the estimation error dynamics (4.25), we introduce some transformations and notations to write the dynamics equations in a simpler and convenient form. Then, we introduce the following notations:

$$\sum_{i=1}^{n_p} \sum_{j=1}^{n_p} f_{ij}^{\xi}(t) \mathcal{H}_{ij}^{n_p} \left(\xi - \hat{\xi} \right) = \mathcal{A}_{n_p} \times \Pi_\xi(\epsilon_\xi) \tag{4.31}$$

$$\sum_{i=1}^{n_p} \sum_{j=1}^{n_w} f_{ij}^{\boldsymbol{\omega}}(t) \mathcal{H}_{ij}^{n_p, n_w} \left(\xi - \hat{\xi} \right) = \mathcal{A}_{n_p, n_w} \times \Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}) \tag{4.32}$$

where

$$\mathcal{A}_{\ell_1, \ell_2} \triangleq \text{blkdiag}\Big(\overbrace{\mathbf{e}_{\ell_1, \ell_2}, \ldots, \mathbf{e}_{\ell_1, \ell_2}}^{\ell_2 \text{ times}} \Big),$$

and

$$\mathbf{e}_{\ell_1, \ell_2} \triangleq \overbrace{\left[\mathrm{e}_{\ell_1}^\top(1) \quad \ldots \quad \mathrm{e}_{\ell_1}^\top(\ell_2) \right]}^{\ell_2 \text{ times}}$$

by setting $\mathcal{A}_{\ell_1, \ell_1} = \mathcal{A}_{\ell_1}$. In (4.31) and (4.32), we have $\ell_1 = n_p$ and $\ell_2 = n_w$. The terms $\Pi_\xi(\epsilon_\xi)$ and $\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega})$ are defined in (4.33) and (4.34), respectively.
Hence, the estimation error equation (4.25) is rewritten as follows:

$$\dot{\epsilon}_\xi = \mathcal{A}_{n_p} \Pi_\xi(\epsilon_\xi) - \mathcal{L} C \epsilon_\xi + \mathcal{A}_{n_p, n_w} \Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}). \tag{4.35}$$

4.3.1.2.a New LMI-based ISS bound

This section is devoted to the main theorem, which provides new sufficient LMI conditions ensuring the ISS property of the estimation error $\epsilon(t)$. Before stating the main theorem, we introduce the following matrices defined by the bounds of the functions f_{ij}^{ξ} and $f_{ij}^{\boldsymbol{\omega}}$:

$$\underline{\Lambda}_i^{\xi} \triangleq \text{blkdiag}\Big(\underline{\gamma}_{f_{ij}^{\xi}} \mathbb{I}_{n_p}, j = 1, \ldots, n_p \Big)$$

$$\bar{\Lambda}_i^{\xi} \triangleq \text{blkdiag}\Big(\bar{\gamma}_{f_{ij}^{\xi}} \mathbb{I}_{n_p}, j = 1, \ldots, n_p \Big)$$

$$\underline{\Lambda}_{\xi} \triangleq \mathrm{blkdiag}\Big(\underline{\Lambda}_i^{f^{\xi}}, i = 1, \ldots, n_p\Big)$$

$$\bar{\Lambda}_{\xi} \triangleq \mathrm{blkdiag}\Big(\bar{\Lambda}_i^{f^{\xi}}, i = 1, \ldots, n\Big)$$

$$\underline{\Lambda}_i^{\boldsymbol{\omega}} \triangleq \mathrm{blkdiag}\Big(\underline{\gamma}_{f_{ij}^{\boldsymbol{\omega}}}\mathbb{I}_{n_w}, j = 1, \ldots, n_w\Big)$$

$$\bar{\Lambda}_i^{\boldsymbol{\omega}} \triangleq \mathrm{blkdiag}\Big(\bar{\gamma}_{f_{ij}^{\boldsymbol{\omega}}}\mathbb{I}_{n_w}, j = 1, \ldots, n_w\Big)$$

$$\underline{\Lambda}_{\boldsymbol{\omega}} \triangleq \mathrm{blkdiag}\Big(\underline{\Lambda}_i^{\boldsymbol{\omega}}, i = 1, \ldots, n_p\Big)$$

$$\bar{\Lambda}_{\boldsymbol{\omega}} \triangleq \mathrm{blkdiag}\Big(\bar{\Lambda}_i^{\boldsymbol{\omega}}, i = 1, \ldots, n_p\Big).$$

We also introduce the following symmetric and positive definite matrices S_{ξ} and $S_{\boldsymbol{\omega}}$ as follows:

$$S_{\xi} \triangleq \mathrm{blkdiag}\Big(S_1^{\xi}, \ldots, S_{n_p}^{\xi}\Big), S_i^{\xi} \triangleq \mathrm{blkdiag}\Big(S_{i1}^{\xi}, \ldots, S_{in_p}^{\xi}\Big) \tag{4.36}$$

$$S_{\boldsymbol{\omega}} \triangleq \mathrm{blkdiag}\Big(S_1^{\boldsymbol{\omega}}, \ldots, S_{n_p}^{\boldsymbol{\omega}}\Big), S_i^{\boldsymbol{\omega}} \triangleq \mathrm{blkdiag}\Big(S_{i1}^{\boldsymbol{\omega}}, \ldots, S_{in_w}^{\boldsymbol{\omega}}\Big). \tag{4.37}$$

where S_{ij}^{ξ} and $S_{ij}^{\boldsymbol{\omega}}$ are symmetric and positive definite matrices of appropriate dimensions.
The main result of this work is formulated in the following theorem.

$$\Pi_{\xi}(\epsilon_{\xi}) \triangleq \mathrm{blkdiag}\Big(f_{ij}^{\xi}(t)\mathbb{I}_{n_p}, j = 1, \ldots, n_p, i = 1, \ldots, n_p\Big)\mathbb{H}\epsilon_{\xi}$$
$$= \begin{bmatrix} \begin{pmatrix} f_{11}^{\xi}\mathbb{I}_{n_p} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_{1n_p}^{\xi}\mathbb{I}_{n_p} \end{pmatrix} & 0 & \cdots \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \begin{pmatrix} f_{n_p}^{\xi}\mathbb{I}_{n_p} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_{n_p}^{\xi}\mathbb{I}_{n_p} \end{pmatrix} \end{bmatrix} \overbrace{\begin{bmatrix} \left.\begin{pmatrix} \mathbb{I}_{n_p} \\ \vdots \\ \mathbb{I}_{n_p} \end{pmatrix}\right\} n_p \text{ times} \\ \vdots \\ \left.\begin{pmatrix} \mathbb{I}_{n_p} \\ \vdots \\ \mathbb{I}_{n_p} \end{pmatrix}\right\} n_p \text{ times} \end{bmatrix}}^{\mathbb{H}} \epsilon_{\xi} \tag{4.33}$$

$$\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}) \triangleq \mathrm{blkdiag}\Big(f_{ij}^{\boldsymbol{\omega}}(t)\mathbb{I}_{n_w}, j = 1, \ldots, n_w, i = 1, \ldots, n_p\Big)\mathbb{J}\boldsymbol{\omega}$$
$$= \begin{bmatrix} \begin{pmatrix} f_{11}^{\boldsymbol{\omega}}\mathbb{I}_{n_w} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_{1n_w}^{\boldsymbol{\omega}}\mathbb{I}_{n_w} \end{pmatrix} & 0 & \cdots \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \begin{pmatrix} f_{n_p1}^{\boldsymbol{\omega}}\mathbb{I}_{n_w} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_{n_pn_w}^{\boldsymbol{\omega}}\mathbb{I}_{n_w} \end{pmatrix} \end{bmatrix} \overbrace{\begin{bmatrix} \left.\begin{pmatrix} \mathbb{I}_{n_w} \\ \vdots \\ \mathbb{I}_{n_w} \end{pmatrix}\right\} n_w \text{ times} \\ \vdots \\ \left.\begin{pmatrix} \mathbb{I}_{n_w} \\ \vdots \\ \mathbb{I}_{n_w} \end{pmatrix}\right\} n_w \text{ times} \end{bmatrix}}^{\mathbb{J}} \boldsymbol{\omega} \tag{4.34}$$

Theorem 4.3.2. *([44]) Assume that there exist a symmetric positive definite matrix $\mathcal{P}$, two symmetric positive definite matrices S_ξ and S_ω under the form (4.36)-(4.37), and a matrix $\mathcal{X}$ of adequate dimensions such that the LMI (4.39) is satisfied. Then the estimation error $\epsilon(t)$ with the observer gain $\mathcal{L} = \mathcal{P}^{-1}\mathcal{X}$ satisfies the following exponential ISS bound:*

$$\|\epsilon(t)\| \leq \max\left(\sqrt{\frac{4\lambda_{\max}(\mathcal{P})}{\lambda_{\min}(\mathcal{P})}}\|\epsilon_0\|\mathrm{e}^{-\frac{\beta}{2}t},\ \sqrt{\frac{4\lambda}{\beta\lambda_{\min}(\mathcal{P})}}\sup_{s\in[0,\ t]}\|\boldsymbol{\omega}(s)\|\right). \tag{4.38}$$

$$\Theta \triangleq \begin{bmatrix} -\mathbb{H}^\top\mathbb{A}_\xi(S_\xi)\mathbb{H} + \mathcal{C}(\mathcal{X}) + \beta\mathcal{P} & \mathcal{P}\mathcal{A}_{n_p} + \mathbb{H}^\top\left(\bar{\Lambda}_\xi + \underline{\Lambda}_\xi\right)^\top S_\xi & \mathcal{P}\mathcal{A}_{n_p,n_w} & 0 \\ (\star) & -2S_\xi & 0 & 0 \\ (\star) & (\star) & -2S_\omega & S_\omega\left(\bar{\Lambda}_\omega + \underline{\Lambda}_\omega\right)\mathbb{J} \\ (\star) & (\star) & (\star) & -\lambda\mathbb{I}_{n_w} - \mathbb{J}^\top\mathbb{A}_\omega(S_\omega)\mathbb{J} \end{bmatrix} < 0 \tag{4.39}$$

$$\mathcal{C}(\mathcal{X}) \triangleq -\mathcal{X}C - C^\top\mathcal{X}^\top,\ \mathbb{A}_\xi(S_\xi) \triangleq \underline{\Lambda}_\xi^\top S_\xi\bar{\Lambda}_\xi + \bar{\Lambda}_\xi^\top S_\xi\underline{\Lambda}_\xi,\ \mathbb{A}_\omega(S_\omega) \triangleq \underline{\Lambda}_\omega^\top S_\omega\bar{\Lambda}_\omega + \bar{\Lambda}_\omega^\top S_\omega\underline{\Lambda}_\omega$$

Proof. Let $\vartheta(\epsilon_\xi) = \epsilon_\xi^\top\mathcal{P}\epsilon_\xi$ be the Lyapunov function candidate, where $\mathcal{P} = \mathcal{P}^\top > 0$. β the decay rate of the state estimation error, and λ the $\mathcal{L}_2$ gain from the generalized disturbances to the state estimation error. We will show that under the condition (4.39), we have

$$\theta(\epsilon_\xi, \boldsymbol{\omega}) \triangleq \dot{\vartheta}(\epsilon_\xi) + \beta\vartheta(\epsilon_\xi) - \lambda\boldsymbol{\omega}^\top\boldsymbol{\omega} < 0. \tag{4.40}$$

By developing the derivative of $\vartheta(\epsilon_\xi)$ along the trajectories of (4.35), and by changing the variable $\mathcal{X} = \mathcal{P}\mathcal{L}$, we obtain

$$\theta(\epsilon_\xi, \boldsymbol{\omega}) = \begin{bmatrix} \epsilon_\xi \\ \Pi_\xi(\epsilon_\xi) \\ \Pi_\omega(\boldsymbol{\omega}) \\ \boldsymbol{\omega} \end{bmatrix}^\top \Omega_{continuous} \begin{bmatrix} \epsilon_\xi \\ \Pi_\xi(\epsilon_\xi) \\ \Pi_\omega(\boldsymbol{\omega}) \\ \boldsymbol{\omega} \end{bmatrix} \tag{4.41}$$

where

$$\Omega_{continuous} \triangleq \begin{bmatrix} -\mathcal{X} - \mathcal{X}^\top + \beta\mathcal{P} & \mathcal{P}\mathcal{A}_{n_p} & \mathcal{P}\mathcal{A}_{n_p,n_w} & 0 \\ \mathcal{A}_{n_p}^\top\mathcal{P} & 0 & 0 & 0 \\ \mathcal{A}_{n_p,n_w}^\top\mathcal{P} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\lambda\mathbb{I}_{n_w} \end{bmatrix}. \tag{4.42}$$

On other hand, from (4.28) and (4.30), we have

$$\begin{aligned} \vartheta_\xi \triangleq{}& \left(\Pi_\xi(\epsilon_\xi) - \bar{\Lambda}_\xi\mathbb{H}\epsilon_\xi\right)^\top S_\xi\left(\Pi_\xi(\epsilon_\xi) - \underline{\Lambda}_\xi\mathbb{H}\epsilon_\xi\right) \\ &+ \left(\Pi_\xi(\epsilon_\xi) - \underline{\Lambda}_\xi\mathbb{H}\epsilon_\xi\right)^\top S_\xi\left(\Pi_\xi(\epsilon_\xi) - \bar{\Lambda}_\xi\mathbb{H}\epsilon_\xi\right) \leq 0 \end{aligned} \tag{4.43}$$

$$\begin{aligned} \vartheta_\omega \triangleq{}& \left(\Pi_\omega(\boldsymbol{\omega}) - \bar{\Lambda}_\omega\mathbb{J}\boldsymbol{\omega}\right)^\top S_\omega\left(\Pi_\omega(\boldsymbol{\omega}) - \underline{\Lambda}_\omega\mathbb{J}\boldsymbol{\omega}\right) \\ &+ \left(\Pi_\omega(\boldsymbol{\omega}) - \underline{\Lambda}_\omega\mathbb{J}\boldsymbol{\omega}\right)^\top S_\omega\left(\Pi_\omega(\boldsymbol{\omega}) - \bar{\Lambda}_\omega\mathbb{J}\boldsymbol{\omega}\right) \leq 0 \end{aligned} \tag{4.44}$$

It is obvious that $\theta(\epsilon_\xi, \boldsymbol{\omega}) \leq \theta(\epsilon_\xi, \boldsymbol{\omega}) - (\vartheta_\xi + \vartheta_{\boldsymbol{\omega}})$. After expanding the terms ϑ_ξ and $\vartheta_{\boldsymbol{\omega}}$, we obtain

$$\theta(\epsilon_\xi, \boldsymbol{\omega}) - (\vartheta_\xi + \vartheta_{\boldsymbol{\omega}}) = \begin{bmatrix} \epsilon_\xi \\ \Pi_\xi(\epsilon_\xi) \\ \Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}) \\ \boldsymbol{\omega} \end{bmatrix}^\top \Theta \begin{bmatrix} \epsilon_\xi \\ \Pi_\xi(\epsilon_\xi) \\ \Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}) \\ \boldsymbol{\omega} \end{bmatrix} \tag{4.45}$$

where Θ is defined in (4.39). This means that under the condition (4.39), we have $\theta(\epsilon_\xi, \boldsymbol{\omega}) < 0$, which implies that

$$\dot{\vartheta}(\epsilon_\xi) < -\beta\vartheta(\epsilon_\xi) + \lambda\|\boldsymbol{\omega}(t)\|^2. \tag{4.46}$$

Hence, from the well-known comparison theorem, we deduce the following inequality:

$$\vartheta(\epsilon_\xi(t)) \leq \vartheta(\epsilon_\xi(0))\mathrm{e}^{-\beta t} + \lambda\mathrm{e}^{-\beta t}\int_0^t \mathrm{e}^{\beta s}\|\boldsymbol{\omega}(s)\|^2 \mathrm{d}s.$$

Since

$$\lambda_{\min}(\mathcal{P})\|\epsilon_\xi\|^2 \leq \vartheta(\epsilon_\xi(t)) \leq \lambda_{\max}(\mathcal{P})\|\epsilon_\xi\|^2$$

then by developing the integral term, we get

$$\|\epsilon_\xi(t)\|^2 \leq \frac{\lambda_{\max}(\mathcal{P})}{\lambda_{\min}(\mathcal{P})}\|\epsilon_\xi(0)\|^2\mathrm{e}^{-\beta t} + \frac{\lambda}{\beta\lambda_{\min}(\mathcal{P})}\sup_{s\in[0,\ t]}\|\boldsymbol{\omega}(s)\|^2. \tag{4.47}$$

By construction of $\eta(t)$ in (4.19), η_0 is known, then we can take $\hat{\xi}_1(0) = \eta_0$, which means that $\|\epsilon_\xi(0)\| = \|\epsilon(0)\|$. Finally, since $\|\epsilon(t)\| \leq \|\epsilon_\xi(t)\|, \forall t \geq 0$, then the bound (4.38) is inferred. This ends the proof of Theorem 4.3.6.

□

4.3.2 Observer Design for the Discrete-Time Case

Motivated by real-world applications, developing discrete-time observers offers advantages in simplicity of implementation, robustness to sampling, and computational efficiency compared to continuous-time observers, especially in digital control systems and applications with non-smooth nonlinearities. In the context of drones in AVF, although the continuous-time observer plays a crucial role in the theoretical development of system dynamics and provides the foundation for design principles and robustness criteria, UAVs in AVFs are implemented in real-time with sensors and processors that inherently deal with discrete-time data. Therefore, the discrete observer design is more directly applicable to these systems, aligning with their sampling nature and computational constraints. Thus, we dedicate this section to investigate the discrete-time case of the dynamic extended output nonlinear observer. We consider the following discrete-time system :

$$\begin{cases} x_{k+1} = \psi(x_k, w_k) \\ y_k = \phi(x_k, v_k) \end{cases} \tag{4.48}$$

where k, the discrete-time index, is non-negative integer such as $k \in \{0, 1, \dots, N\}$, $x_k \in \mathbb{R}^n$ is the system state and $y_k \in \mathbb{R}^p$ is the vector of output measurements of the system. $w_k \in \mathbb{R}^s$ and $v_k \in \mathbb{R}^q$ are unknown external disturbances. The nonlinear functions $\psi(\cdot,\cdot)$ and $\phi(\cdot,\cdot)$ are assumed to be Lipschitz with respect to x_k, on $\Omega \subseteq \mathbb{R}^n$, where Ω_k is a positively invariant compact and convex set of the system (4.48). Then, the functions $\psi(\pi_\Omega(\cdot), w_k)$ and $\phi(\pi_\Omega(\cdot), v_k)$ are globally Lipschitz in $\mathbb{R}^n$, where $\pi_\Omega(x_k)$ is the *Hilbert* projection of x_k on the convex set Ω.

Commonly in the literature, we propose the following state observer:

$$\hat{x}_{k+1} = \psi(\hat{x}_k, 0) + \mathbb{L}\Big(y - \phi(\hat{x}_k, 0)\Big), \tag{4.49}$$

where $\hat{x}_k \in \mathbb{R}^n$ is the estimate of the system state x_k and $\mathbb{L} \in \mathbb{R}^{n\times p}$ is the constant observer gain to be determined such that the estimation error $e_k = x_k - \hat{x}_k$ converges exponentially towards zero. Nevertheless, with this kind of classical and natural observer, the observer gain is multiplied by the output nonlinearity and then by the disturbance in the measurement. Thus, the effect of the measurement noise can be amplified by this coupling between the observer gain and the output measurement. In this paper, the main objective consists of suggesting a different state observer for the discrete-time system (4.48). We will suggest a filtered-like output-based observer, that makes feasible the avoidance of the output nonlinearity and disturbances explicit multiplication by the observer gain. For the observer design problem, we introduce the following lemma in its discrete-time version.

Lemma 4.3.3 ([178]). *Consider a function $\varphi : \mathbb{R}^n \longrightarrow \mathbb{R}^p$. Assume that φ is γ_φ−Lipschitz on a nonempty subset $\Omega \subseteq \mathbb{R}^n$, with γ_{φ_i} the Lipschitz constant of each component φ_i of φ. Then, there exist*

- *a Lipschitz extension $\breve{\varphi} \triangleq \varphi \circ \pi_\Omega : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ of φ, with $\breve{\varphi}(x) = \varphi(x), \forall x \in \Omega$;*
- *functions $\varphi_{ij} : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$*
- *constants $\underline{\gamma}_{\varphi_{ij}}$ and $\bar{\gamma}_{\varphi_{ij}}$*

such that for all $X, Y \in \mathbb{R}^n, X \neq Y$, we have

$$\breve{\varphi}(X) - \breve{\varphi}(Y) = \sum_{i=1}^{p}\sum_{j=1}^{n} \varphi_{ij}(k)\mathcal{H}_{ij}^{p,n}\Big(X - Y\Big) \tag{4.50}$$

and

$$-\gamma_{\varphi_{ij}} \leq \underline{\gamma}_{\varphi_{ij}} \leq \varphi_{ij}(k) \leq \bar{\gamma}_{\varphi_{ij}} \leq \gamma_{\varphi_{ij}} \tag{4.51}$$

where

$$\varphi_{ij}(k) \triangleq \varphi_{ij}\Big(X^{Y_{j-1}}, X^{Y_j}\Big),\ \mathcal{H}_{ij}^{p,n} \triangleq \mathrm{e}_p(i)\mathrm{e}_n^\top(j).$$

Proof. The detail of the proof is given in [178]. It is based on the use of the properties of the *Hilbert* projection function, π_Ω, on the convex set Ω.

□

4.3.2.1 Output-Based Dynamic Extension Approach for Discrete-time systems

The initial system is given by the set of equations (4.48). We introduce the following system to perform an output variable change:

$$\begin{cases} \eta_{k+1} = \kappa(\eta_k, y_k) \\ \eta_0 \text{ (known)} \end{cases} \tag{4.52}$$

where $\kappa : \mathbb{R}^{n_\eta} \times \mathbb{R}^p \longrightarrow \mathbb{R}^{n_\eta}$ is a global Lipschitz function with respect to its arguments. The function κ is known and chosen by the user, depending on the system at hand. Since η_0, $\kappa(.)$, and y_k are known, then the state η_k may be considered as the new output measurement driven from the measurement y_k. Thus, we consider the new output $\boldsymbol{y}_k = \eta_k$, represented in terms of the extended state vector ξ_k as follows:

$$\boldsymbol{y}_k = \eta_k = \overbrace{\begin{bmatrix} \mathbb{I}_p & 0 \end{bmatrix}}^{C} \xi_k, \text{ with } \xi_k \triangleq \begin{bmatrix} \eta_k \\ x_k \end{bmatrix} \tag{4.53}$$

where $\mathbb{I}_p$ is the identity matrix of dimension p.
Let us define also the vector $\boldsymbol{\omega}$ combining the w and v as follows:

$$\boldsymbol{\omega}_k \triangleq \begin{bmatrix} w_k \\ v_k \end{bmatrix}.$$

Hence, we can transform the equations of the original system into a new system with the state variable ξ:

$$\begin{cases} \xi_{k+1} = f(\xi_k, \boldsymbol{\omega}_k) \triangleq \begin{bmatrix} \kappa\left(\eta_k, \phi(x_k, v_k)\right) \\ \psi(x_k, w_k) \end{bmatrix} \\ \boldsymbol{y}_k = \eta_k = C\xi_k \end{cases} \tag{4.54}$$

This new system (4.54) underscores, as in the continous-time systems case, the relationships between the output vector $\boldsymbol{y}_k$ and the system states represented by ξ_k. It is worth to note that the dynamics of the states are now given by $f(\xi_k, \boldsymbol{\omega}_k)$, which is a function of the generalized state ξ_k and extended disturbance vector $\boldsymbol{\omega}_k \in \mathbb{R}^{n_w}$, where $n_w = s + q$. The state observer we will consider in this work corresponds to the transformed system (4.54) with linear outputs. Then, the observer synthesis and the Lyapunov analysis become simpler.

Remark 4.3.4. *Beyond simplifying the convergence analysis, the new output $\boldsymbol{y}_k$ serves as a filtered version of the true measurement y_k, which can enhance the performance and quality of estimation, particularly in scenarios where output measurements are affected by disturbances. Given that this paper focuses on an LMI-based approach, where the design relies on solving a set of LMI conditions, adopting a linear filter can further streamline the design process and prevent an exponential increase in the size of LMIs to be solved. Thus, we generally propose considering a linear filter as follows:*

$$\kappa(\eta_k, y_k) \triangleq A_\eta \eta_k + B_\eta y_k, \tag{4.55}$$

where A_η and B_η are known constant matrices of appropriate dimensions. However, in the proposed theoretical developments, we consider the general form (4.52), which provides a more comprehensive design method.

Remark 4.3.5. *About the choice of the matrices A_η and B_η :*
Since the aim of the output-based dynamic extension approach is reducing the effect of measurement noise on the state observer :

- *A_η should introduce some dynamics that slow or filter the effect of the noise :*

$$A_\eta = \alpha_A \mathbb{I}_p, \quad 0 < \alpha_A < 1$$

such as α_A a tuning parameter.

- *B_η should weigh how much of the noisy measurement y_k impacts the extended output η_k . In many cases, this matrix will not be identity. Instead, it would be tuned to balance between responsiveness and noise reduction :*

$$B_\eta = \beta_B \mathbb{I}_p, \quad 0 < \beta_B < 1$$

such as β_B a tuning parameter.

Hence, adjusting α_A and β_B ensures a trade-off between the smoothness of the new output and the system responsiveness to the noisy measurements.

4.3.2.2 Discrete-Time Observer Design

This section focuses on the observer design approach proposed in this paper. By applying the extended dynamics (4.54), we suggest a discrete-time observer structure driven by the new output $\boldsymbol{y}_k$, instead of using the standard state observer form (4.49) driven by the original nonlinear output y_k. Actually, the new output $\boldsymbol{y}_k$, acting as a filter, makes feasible the filtering of disturbances or measurement noise. Figure 4.2 shows the estimation strategy. The discrete-time state observer

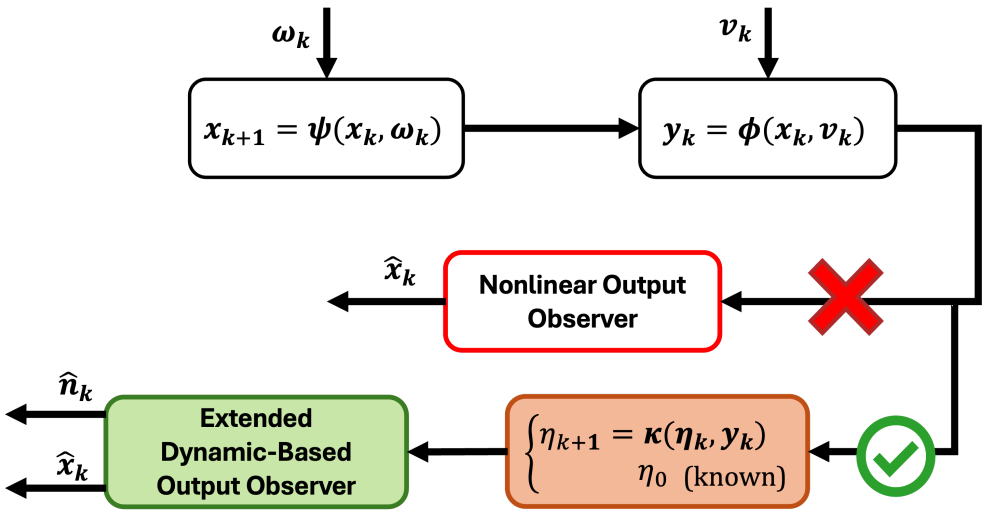


Figure 4.2 – Output-based Dynamic Extension Observer for Discrete-time systems

form is described by the following equations:

$$\begin{cases} \hat{\xi}_{k+1} = \check{f}(\hat{\xi}_k, 0) + \mathcal{L}\Big(\boldsymbol{y}_k - C\xi_k\Big) \\ \hat{x}_k = \begin{bmatrix} 0 & \mathbb{I}_n \end{bmatrix} \hat{\xi}_k \end{cases} \tag{4.58}$$

where

$$\check{f}(\hat{\xi}_k, \boldsymbol{\omega_k}) = f\left(\pi_{\Omega_\xi}(\hat{\xi}_k), \boldsymbol{\omega_k}\right), \forall \boldsymbol{\omega_k} \in \Omega_{\boldsymbol{\omega}} \ni 0, \ \Omega_\xi \triangleq \mathbb{R}^p \times \Omega, \tag{4.59}$$

and $\mathcal{L} \in \mathbb{R}^{(n+p)\times p}$ is the constant observer gain to be computed such that the estimation error $\epsilon_k \triangleq x_k - \hat{x}_k$ satisfies an exponential ISS bound to be determined later. To develop a synthesis method based on Lyapunov analysis, let us define the extended error vector $\tilde{\xi}_k \triangleq \xi_k - \hat{\xi}_k$, which satisfies the following dynamics:

$$\tilde{\xi}_{k+1} = \Big[f(\xi_k, \boldsymbol{\omega}_k) - \check{f}(\hat{\xi}_k, 0)\Big] - \mathcal{L}C\tilde{\xi}_k. \tag{4.60}$$

We decompose the nonlinear term as follows:

$$f(\xi_k, \boldsymbol{\omega}_k) - \check{f}(\hat{\xi}_k, 0) = \Big(f(\xi_k, \boldsymbol{\omega}_k) - \check{f}(\hat{\xi}_k, \boldsymbol{\omega}_k)\Big) + \Big(\check{f}(\hat{\xi}_k, \boldsymbol{\omega}_k) - \check{f}(\hat{\xi}_k, 0)\Big). \tag{4.61}$$

Since $\hat{\xi}_k \in \Omega_\xi$, then we have $f(\xi_k, \boldsymbol{\omega}_k) = \check{f}(\xi_k, \boldsymbol{\omega}_k)$. It follows that from Lemma 4.3.3, there exist function $f^\xi_{ij} : \mathbb{R}^{n_p} \times \mathbb{R}^{n_p} \longrightarrow \mathbb{R}$ and constants $\underline{\gamma}_{f^\xi_{ij}}$ and $\bar{\gamma}_{f^\xi_{ij}}$ such that

$$f(\xi_k, \boldsymbol{\omega}_k) - \check{f}(\hat{\xi}_k, \boldsymbol{\omega}_k) = \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} f^\xi_{ij}(k) \mathcal{H}^{n_p}_{ij} \tilde{\xi}_k$$

$$\Pi_\xi(\tilde{\xi}_k) \triangleq \text{blkdiag}\Big(f_{ij}^{\xi}(k)\mathbb{I}_{n_p}, j=1,\dots,n_p, i=1,\dots,n_p\Big)\mathbb{H}\tilde{\xi}_k$$

$$= \begin{bmatrix} \begin{pmatrix} f_{11}^{\xi}\mathbb{I}_{n_p} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & f_{1n_p}^{\xi}\mathbb{I}_{n_p} \end{pmatrix} & 0 & \dots \\ \vdots & \ddots & \vdots \\ 0 & \dots & \begin{pmatrix} f_{n_p}^{\xi}\mathbb{I}_{n_p} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & f_{n_p}^{\xi}\mathbb{I}_{n_p} \end{pmatrix} \end{bmatrix} \overbrace{\begin{bmatrix} \left.\begin{pmatrix} \mathbb{I}_{n_p} \\ \vdots \\ \mathbb{I}_{n_p} \end{pmatrix}\right\} n_p \text{ times} \\ \vdots \\ \left.\begin{pmatrix} \mathbb{I}_{n_p} \\ \vdots \\ \mathbb{I}_{n_p} \end{pmatrix}\right\} n_p \text{ times} \end{bmatrix}}^{\mathbb{H}} \tilde{\xi}_k \tag{4.56}$$

$$\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k) \triangleq \text{blkdiag}\Big(f_{ij}^{\boldsymbol{\omega}}(k)\mathbb{I}_{n_w}, j=1,\dots,n_w, i=1,\dots,n_p\Big)\mathbb{J}\boldsymbol{\omega}_k$$

$$= \begin{bmatrix} \begin{pmatrix} f_{11}^{\boldsymbol{\omega}}\mathbb{I}_{n_w} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & f_{1n_w}^{\boldsymbol{\omega}}\mathbb{I}_{n_w} \end{pmatrix} & 0 & \dots \\ \vdots & \ddots & \vdots \\ 0 & \dots & \begin{pmatrix} f_{n_p1}^{\boldsymbol{\omega}}\mathbb{I}_{n_w} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & f_{n_pn_w}^{\boldsymbol{\omega}}\mathbb{I}_{n_w} \end{pmatrix} \end{bmatrix} \overbrace{\begin{bmatrix} \left.\begin{pmatrix} \mathbb{I}_{n_w} \\ \vdots \\ \mathbb{I}_{n_w} \end{pmatrix}\right\} n_w \text{ times} \\ \vdots \\ \left.\begin{pmatrix} \mathbb{I}_{n_w} \\ \vdots \\ \mathbb{I}_{n_w} \end{pmatrix}\right\} n_w \text{ times} \end{bmatrix}}^{\mathbb{J}} \boldsymbol{\omega}_k \tag{4.57}$$

and

$$-\gamma_{f_{ij}^{\xi}} \leq \underline{\gamma}_{f_{ij}^{\xi}} \leq f_{ij}^{\xi}(k) \leq \bar{\gamma}_{f_{ij}^{\xi}} \leq \gamma_{f_{ij}^{\xi}}. \tag{4.62}$$

Similarly, there exist functions $f_{ij}^{\boldsymbol{\omega}}: \mathbb{R}^{n_w} \times \mathbb{R}^{n_w} \longrightarrow \mathbb{R}$ and constants $\underline{\gamma}_{f_{ij}^{\boldsymbol{\omega}}}$ and $\bar{\gamma}_{f_{ij}^{\boldsymbol{\omega}}}$ such that

$$\check{f}(\hat{\xi}_k, \boldsymbol{\omega}_k) - \check{f}(\hat{\xi}_k, 0) = \sum_{i=1}^{n_p}\sum_{j=1}^{n_w} f_{ij}^{\boldsymbol{\omega}}(k)\mathcal{H}_{ij}^{n_p,n_w}\boldsymbol{\omega}_k \tag{4.63}$$

and

$$-\gamma_{f_{ij}^{\boldsymbol{\omega}}} \leq \underline{\gamma}_{f_{ij}^{\boldsymbol{\omega}}} \leq f_{ij}^{\boldsymbol{\omega}}(k) \leq \bar{\gamma}_{f_{ij}^{\boldsymbol{\omega}}} \leq \gamma_{f_{ij}^{\boldsymbol{\omega}}}. \tag{4.64}$$

Before tackling the Lyapunov analysis on the estimation error dynamics (4.60), we introduce some transformations and notations to simplify writing the dynamics equations in the convenient form. Then, we introduce the following notations:

$$\sum_{i=1}^{n_p}\sum_{j=1}^{n_p} f_{ij}^{\xi}(k)\mathcal{H}_{ij}^{n_p}\Big(\xi_k - \hat{\xi}_k\Big) = \mathcal{A}_{n_p} \times \Pi_\xi(\tilde{\xi}_k) \tag{4.65}$$

$$\sum_{i=1}^{n_p}\sum_{j=1}^{n_w} f_{ij}^{\boldsymbol{\omega}}(k)\mathcal{H}_{ij}^{n_p,n_w}\boldsymbol{\omega}_k = \mathcal{A}_{n_p,n_w} \times \Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k) \tag{4.66}$$

where

$$\mathcal{A}_{\ell_1,\ell_2} \triangleq \text{blkdiag}\Big(\overbrace{\mathbf{e}_{\ell_2},\ldots,\mathbf{e}_{\ell_2}}^{\ell_1 \text{ times}}\Big),$$

and

$$\mathbf{e}_{\ell_2} \triangleq \overbrace{\begin{bmatrix}\mathrm{e}_{\ell_2}^\top(1) & \ldots & \mathrm{e}_{\ell_2}^\top(\ell_2)\end{bmatrix}}^{\ell_2 \text{ times}}$$

by setting $\mathcal{A}_{\ell_1,\ell_1} = \mathcal{A}_{\ell_1}$. In (4.65) and (4.66), we have $\ell_1 = n_p$ and $\ell_2 = n_w$. The terms $\Pi_\xi(\tilde{\xi}_k)$ and $\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k)$ are defined in (4.56) and (4.57), respectively.
Hence, the estimation error equation (4.60) is rewritten as follows:

$$\tilde{\xi}_{k+1} = \mathcal{A}_{n_p}\Pi_\xi(\tilde{\xi}_k) - \mathcal{L}C\tilde{\xi}_k + \mathcal{A}_{n_p,n_w}\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k). \tag{4.67}$$

4.3.2.2.a Exponential ISS Stability Analysis

This section presents the main theorem, which provides new sufficient LMI conditions ensuring the ISS property of the estimation error ϵ_k. To state the main theorem for the observer design in the discrete-time domain, we introduce, similiraly to the approach in Section 4.3.1 the following matrices defined by the bounds of the functions f_{ij}^ξ and $f_{ij}^{\boldsymbol{\omega}}$:

$$\underline{\Lambda}_i^\xi \triangleq \text{blkdiag}\Big(\underline{\gamma}_{f_{ij}^\xi}\mathbb{I}_{n_p}, j=1,\ldots,n_p\Big)$$

$$\bar{\Lambda}_i^\xi \triangleq \text{blkdiag}\Big(\bar{\gamma}_{f_{ij}^\xi}\mathbb{I}_{n_p}, j=1,\ldots,n_p\Big)$$

$$\underline{\Lambda}_\xi \triangleq \text{blkdiag}\Big(\underline{\Lambda}_i^{f^\xi}, i=1,\ldots,n_p\Big)$$

$$\bar{\Lambda}_\xi \triangleq \text{blkdiag}\Big(\bar{\Lambda}_i^{f^\xi}, i=1,\ldots,n\Big)$$

$$\underline{\Lambda}_i^{\boldsymbol{\omega}} \triangleq \text{blkdiag}\Big(\underline{\gamma}_{f_{ij}^{\boldsymbol{\omega}}}\mathbb{I}_{n_w}, j=1,\ldots,n_w\Big)$$

$$\bar{\Lambda}_i^{\boldsymbol{\omega}} \triangleq \text{blkdiag}\Big(\bar{\gamma}_{f_{ij}^{\boldsymbol{\omega}}}\mathbb{I}_{n_w}, j=1,\ldots,n_w\Big)$$

$$\underline{\Lambda}_{\boldsymbol{\omega}} \triangleq \text{blkdiag}\Big(\underline{\Lambda}_i^{\boldsymbol{\omega}}, i=1,\ldots,n_p\Big)$$

$$\bar{\Lambda}_{\boldsymbol{\omega}} \triangleq \text{blkdiag}\Big(\bar{\Lambda}_i^{\boldsymbol{\omega}}, i=1,\ldots,n_p\Big).$$

We also introduce the following symmetric and positive definite matrices S_ξ and $S_{\boldsymbol{\omega}}$ as follows:

$$S_\xi \triangleq \text{blkdiag}\Big(S_1^\xi,\ldots,S_{n_p}^\xi\Big), S_i^\xi \triangleq \text{blkdiag}\Big(S_{i1}^\xi,\ldots,S_{in_p}^\xi\Big) \tag{4.68}$$

$$S_{\boldsymbol{\omega}} \triangleq \text{blkdiag}\Big(S_1^{\boldsymbol{\omega}},\ldots,S_{n_p}^{\boldsymbol{\omega}}\Big), S_i^{\boldsymbol{\omega}} \triangleq \text{blkdiag}\Big(S_{i1}^{\boldsymbol{\omega}},\ldots,S_{in_w}^{\boldsymbol{\omega}}\Big). \tag{4.69}$$

where S_{ij}^ξ and $S_{ij}^{\boldsymbol{\omega}}$ are symmetric and positive definite matrices of appropriate dimensions.
The main result is summarized in the following theorem applied to the system 4.54 :

$$\mathbb{LMI} \triangleq \begin{bmatrix} -\beta\mathcal{P} - \mathbb{H}^\top \mathbb{A}_\xi(S_\xi)\mathbb{H} & \mathbb{H}^\top \left(\bar{\Lambda}_\xi + \underline{\Lambda}_\xi\right)^\top S_\xi & 0 & 0 & \mathcal{C}^\top \mathcal{X}^\top \\ (\star) & -2S_\xi & 0 & 0 & \mathcal{A}_{n_p}^\top \mathcal{P} \\ (\star) & (\star) & -2S_\omega & S_\omega\left(\bar{\Lambda}_\omega + \underline{\Lambda}_\omega\right)\mathbb{J} & \mathcal{A}_{n_p,n_w}^\top \mathcal{P} \\ (\star) & (\star) & (\star) & -\mathbb{Q} - \mathbb{J}^\top \mathbb{A}_\omega(S_\omega)\mathbb{J} & 0 \\ (\star) & (\star) & (\star) & (\star) & -\mathcal{P} \end{bmatrix} < 0 \tag{4.70}$$

$$\mathbb{A}_\xi(S_\xi) \triangleq \underline{\Lambda}_\xi^\top S_\xi \bar{\Lambda}_\xi + \bar{\Lambda}_\xi^\top S_\xi \underline{\Lambda}_\xi, \ \mathbb{A}_\omega(S_\omega) \triangleq \underline{\Lambda}_\omega^\top S_\omega \bar{\Lambda}_\omega + \bar{\Lambda}_\omega^\top S_\omega \underline{\Lambda}_\omega$$

$$\Omega_{discrete} \triangleq \begin{bmatrix} C^\top \mathcal{L}^\top \mathcal{P}\mathcal{L}C - \beta\mathcal{P} & -C^\top \mathcal{L}^\top \mathcal{P}\mathcal{A}_{n_p} & -C^\top \mathcal{L}^\top \mathcal{P}\mathcal{A}_{n_p,n_w} & 0 \\ (\star) & \mathcal{A}_{n_p}^\top \mathcal{P}\mathcal{A}_{n_p} & \mathcal{A}_{n_p}^\top \mathcal{P}\mathcal{A}_{n_p,n_w} & 0 \\ (\star) & (\star) & \mathcal{A}_{n_p,n_w}^\top \mathcal{P}\mathcal{A}_{n_p,n_w} & 0 \\ (\star) & (\star) & (\star) & -\mathbb{Q} \end{bmatrix} \tag{4.73}$$

Theorem 4.3.6. *Assume that there exist a symmetric positive definite matrix $\mathcal{P}$, two symmetric positive definite matrices S_ξ and S_ω under the form (4.68)-(4.69), and a matrix $\mathcal{X}$ of adequate dimensions such that the LMI (4.70) is satisfied. Then, with $\hat{\eta}_0 = \eta_0$, the estimation error ϵ_k with the observer gain $\mathcal{L} = \mathcal{P}^{-1}\mathcal{X}$ satisfies the following exponential ISS bound with the exponential discount parameter $0 < \beta < 1$:*

$$\|\epsilon_k\| \le \sqrt{\frac{\lambda_{\max}(\mathcal{P})}{\lambda_{\min}(\mathcal{P})}} \|\epsilon_0\| \, \beta^k + \sqrt{\frac{\lambda_{\max}(\mathbb{Q})}{\beta\lambda_{\min}(\mathcal{P})}} \sum_{j=0}^{k-1} \sqrt{\beta}^{k-j} \, \|\boldsymbol{\omega}_j\| \, . \tag{4.71}$$

Proof. Let $\vartheta(\tilde{\xi}_k) = \tilde{\xi}_k^\top \, \mathcal{P} \, \tilde{\xi}_k$ be the Lyapunov function candidate, where $\mathcal{P} = \mathcal{P}^\top > 0$ and

$$\Delta\vartheta(\tilde{\xi}_k) \triangleq \vartheta(\tilde{\xi}_{k+1}) - \beta\vartheta(\tilde{\xi}_k).$$

We will show that under the condition (4.70), we have

$$\theta(\tilde{\xi}_k, \boldsymbol{\omega}_k) \triangleq \Delta\vartheta(\tilde{\xi}_k) - \boldsymbol{\omega}_k^\top \mathbb{Q}\boldsymbol{\omega}_k < 0, \tag{4.72}$$

where $\mathbb{Q} = \mathbb{Q}^\top > 0$. By developing $\Delta\vartheta(\tilde{\xi}_k)$ and by changing the variable $\mathcal{X} = \mathcal{P}\mathcal{L}$, we obtain

$$\theta(\tilde{\xi}_k, \boldsymbol{\omega}_k) = \begin{bmatrix} \tilde{\xi}_k \\ \Pi_\xi(\tilde{\xi}_k) \\ \Pi_\omega(\boldsymbol{\omega}_k) \\ \boldsymbol{\omega}_k \end{bmatrix}^\top \Omega_{discrete} \begin{bmatrix} \tilde{\xi}_k \\ \Pi_\xi(\tilde{\xi}_k) \\ \Pi_\omega(\boldsymbol{\omega}_k) \\ \boldsymbol{\omega}_k \end{bmatrix}$$

where Ω is given by 4.73. On other hand, from (4.28) and (4.30), we have

$$\vartheta_\xi \triangleq \left(\Pi_\xi(\tilde{\xi}_k) - \bar{\Lambda}_\xi \mathbb{H}\tilde{\xi}_k\right)^\top S_\xi \left(\Pi_\xi(\tilde{\xi}_k) - \underline{\Lambda}_\xi \mathbb{H}\tilde{\xi}_k\right) + \left(\Pi_\xi(\tilde{\xi}_k) - \underline{\Lambda}_\xi \mathbb{H}\tilde{\xi}_k\right)^\top S_\xi \left(\Pi_\xi(\tilde{\xi}_k) - \bar{\Lambda}_\xi \mathbb{H}\tilde{\xi}_k\right) \leq 0$$

$$\vartheta_{\boldsymbol{\omega}} \triangleq \left(\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k) - \bar{\Lambda}_{\boldsymbol{\omega}} \mathbb{J}\boldsymbol{\omega}_k\right)^\top S_{\boldsymbol{\omega}} \left(\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k) - \underline{\Lambda}_{\boldsymbol{\omega}} \mathbb{J}\boldsymbol{\omega}_k\right) + \left(\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k) - \underline{\Lambda}_{\boldsymbol{\omega}} \mathbb{J}\boldsymbol{\omega}_k\right)^\top S_{\boldsymbol{\omega}} \left(\Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k) - \bar{\Lambda}_{\boldsymbol{\omega}} \mathbb{J}\boldsymbol{\omega}_k\right) \leq 0$$

It is obvious that $\theta(\tilde{\xi}_k, \boldsymbol{\omega}_k) \leq \theta(\tilde{\xi}_k, \boldsymbol{\omega}_k) - (\vartheta_\xi + \vartheta_{\boldsymbol{\omega}})$. After expanding the terms ϑ_ξ and $\vartheta_{\boldsymbol{\omega}}$, we obtain

$$\theta(\tilde{\xi}_k, \boldsymbol{\omega}_k) - (\vartheta_\xi + \vartheta_{\boldsymbol{\omega}}) = \begin{bmatrix} \tilde{\xi}_k \\ \Pi_\xi(\epsilon_{\xi_k}) \\ \Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k) \\ \boldsymbol{\omega}_k \end{bmatrix}^\top \Theta \begin{bmatrix} \tilde{\xi}_k \\ \Pi_\xi(\epsilon_{\xi_k}) \\ \Pi_{\boldsymbol{\omega}}(\boldsymbol{\omega}_k) \\ \boldsymbol{\omega}_k \end{bmatrix}$$

where Θ is given in (4.74).

$$\Theta \triangleq \begin{bmatrix} \Theta_{1,1} & -(\mathcal{L}\mathcal{C})^\top \mathcal{P}\mathcal{A}_{n_p} + \mathbb{H}^\top \left(\bar{\Lambda}_\xi + \underline{\Lambda}_\xi\right)^\top S_\xi & -(\mathcal{L}\mathcal{C})^\top \mathcal{P}\mathcal{A}_{n_p,n_w} & 0 \\ (\star) & \mathcal{A}_{n_p}^\top \mathcal{P}\mathcal{A}_{n_p} - 2S_\xi & \mathcal{A}_{n_p}^\top \mathcal{P}\mathcal{A}_{n_p,n_w} & 0 \\ (\star) & (\star) & \mathcal{A}_{n_p,n_w}^\top \mathcal{P}\mathcal{A}_{n_p,n_w} - 2S_{\boldsymbol{\omega}} & S_{\boldsymbol{\omega}}\left(\bar{\Lambda}_{\boldsymbol{\omega}} + \underline{\Lambda}_{\boldsymbol{\omega}}\right)\mathbb{J} \\ (\star) & (\star) & (\star) & -\lambda \mathbb{I}_{n_w} - \mathbb{J}^\top \mathbb{A}_{\boldsymbol{\omega}}(S_{\boldsymbol{\omega}})\mathbb{J} \end{bmatrix} \tag{4.74}$$

$$\Theta_{1,1} = (\mathcal{L}\mathcal{C})^\top P\mathcal{L}C - \beta\mathcal{P} - \mathbb{H}^\top \mathbb{A}_\xi(S_\xi)\mathbb{H},\ \mathbb{A}_\xi(S_\xi) \triangleq \underline{\Lambda}_\xi^\top S_\xi \bar{\Lambda}_\xi + \bar{\Lambda}_\xi^\top S_\xi \underline{\Lambda}_\xi,\ \mathbb{A}_{\boldsymbol{\omega}}(S_{\boldsymbol{\omega}}) \triangleq \underline{\Lambda}_{\boldsymbol{\omega}}^\top S_{\boldsymbol{\omega}} \bar{\Lambda}_{\boldsymbol{\omega}} + \bar{\Lambda}_{\boldsymbol{\omega}}^\top S_{\boldsymbol{\omega}} \underline{\Lambda}_{\boldsymbol{\omega}}$$

From Schur Lemma, we have $\Theta < 0$ is equivalent to (4.39). Now, we prove the boundedness of the state estimation error ϵ_k in terms of the initial error ϵ_0 and the norm of the noise, $\|\boldsymbol{\omega}_i\|$ weighted by an exponential discount parameter. We have

$$\theta(\tilde{\xi}_k, \boldsymbol{\omega}_k) = \Delta\vartheta(\tilde{\xi}_k) - \boldsymbol{\omega}_k^\top \mathbb{Q}\boldsymbol{\omega}_k < 0,$$

which leads to

$$\vartheta(\tilde{\xi}_k) < \beta\vartheta(\tilde{\xi}_{k-1}) + \boldsymbol{\omega}_{k-1}^\top \mathbb{Q}\boldsymbol{\omega}_{k-1}.$$

By using backward substitution and geometric sequence theory, we derive an upper bound for $\vartheta(\tilde{\xi}_k)$ as

$$\begin{aligned} \vartheta(\tilde{\xi}_k) &< \beta^k \vartheta(\tilde{\xi}_0) + \sum_{i=1}^{k} \beta^{i-1} \boldsymbol{\omega}_{k-i}^\top \mathbb{Q}\boldsymbol{\omega}_{k-i} \\ &\leq \beta^k \vartheta(\tilde{\xi}_0) + \frac{\lambda_{\max}(\mathbb{Q})}{\beta} \sum_{j=0}^{k-1} \beta^{k-j} \|\boldsymbol{\omega}_j\|^2 . \end{aligned} \tag{4.75}$$

We establish the relationship between $\vartheta(\tilde{\xi}_k)$ and the norm of the extended estimation error $\|\tilde{\xi}_k\|$ by using the inequality

$$\lambda_{\min}(\mathcal{P}) \left\|\tilde{\xi}_k\right\|^2 \leq \vartheta_k(\tilde{\xi}_k) \leq \lambda_{\max}(\mathcal{P}) \left\|\tilde{\xi}_k\right\|^2. \tag{4.76}$$

For $\hat{\eta}_0 = \eta_0$, which is always possible since η_0 is known, we have $\|\tilde{\xi}_0\| = \|\epsilon_0\|$. Hence, by combining (4.76) with the previous equality, the ISS bound (4.71) on the estimation error ϵ_k is straightforwardly inferred. That ends the proof.

□

4.4 Simulation Results

In this section, we showcase the effectiveness of the proposed continuous-time observer design in 4.2 and the discrete-time observer design 4.3.2 applied to the OS4 indoor mini quadrotor model developed by researchers in ETH Zurich [38, 39]. Although the physical OS4 quadrotor may not be commercially available, its well-established mathematical model offers an excellent approximation to simulate UAV dynamics and test various control strategies. This model provides the necessary framework to explore state estimation techniques in the indoor dynamic environment of AVFs, where factors like limited space and disturbances pose challenges to drone performance. As represented in Figure 4.4, we assume that the drone is moving between cultivation shelves in the AVF, to perform autonomous actions (e.g. inspecting crop, watering, trimming..).
We consider that :

- given the space constraints the roll, pitch, and yaw must be kept within safe limits to avoid collisions with walls, shelves, or plants.
- the drone must perform precision movements, with accurate roll, pitch, and yaw rates.
- the UAV stability can be affected by the faults of air circulation caused by the drone propellers and/or the motor noise.

4.4.1 Mathematical Model of the UAV

The quadcopter is a four-rotor aircraft composed of a rigid body and four rotors generating independent thrusts represented in Figure 4.3 by F_1, F_2, F_3 and F_4.

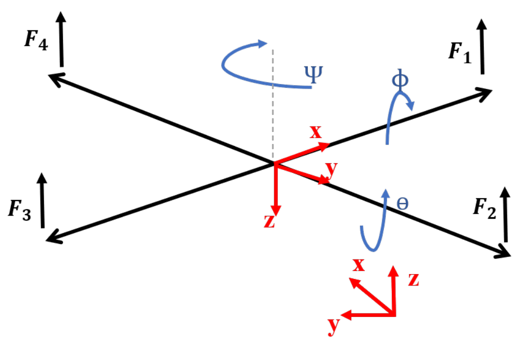


Figure 4.3 – Inertial and body frame of the quadrotor. The circular arrows indicate the direction of rotation of each propeller.

The dynamics of the rotation and the translation motion are generated by both Euler-Newton and Euler-Lagrange as detailed in [39]- [145]. The subsequent model is given by the following equations, [39]:

$$\begin{cases} \ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J_r}{I_x}\dot{\theta}\Omega + \frac{l}{I_x}U_2 \\ \ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{J_r}{I_y}\dot{\phi}\Omega + \frac{l}{I_y}U_3 \\ \ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}U_4 \\ \ddot{z} = -g + (\cos\phi\cos\theta)\frac{1}{m}U_1 \\ \ddot{x} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{1}{m}U_1 \\ \ddot{y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\frac{1}{m}U_1 \end{cases} \tag{4.77}$$

The input signals of the system $U = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix}^\top$ and the angular velocities $\Omega = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \end{bmatrix}^\top$ satisfy the following relation:

$$\begin{cases} U_1 = b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ U_2 = b\left(\Omega_4^2 - \Omega_2^2\right) \\ U_3 = b\left(\Omega_3^2 - \Omega_1^2\right) \\ U_4 = d\left(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2\right) \\ \Omega = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3 \end{cases} \tag{4.78}$$

Table 4.1 – OS4 Parameters from [39]

Parameter	**Definition**	**Value**
ϕ	Roll angle	-
θ	Pitch angle	-
ψ	Yaw angle	-
Ω	Rotor speed	-
$I_{x,y,z}$	Body inertia	7.5e-3 kg.m^2
J_r	Rotor inertia	6e-5 kg.m^2
b	Thrust factor	-
d	Drag factor	-
l	Lever	0.23 m
m	Mass	0.65 kg
g	Gravitational acceleration	9.81 kg.m^2

where the parameters and their values are reported in the Table 4.1. For more details on the conceptual design of the drone, we refer the reader to [38].
In this study, we focus on rotational dynamics and vertical motion. Actually, given the dynamic structure AVF, the quadcopter often operates in a vertically structured space through densely packed crops. Thus, maintaining stable altitude and orientation is critical for avoiding collision with crops or obstacles that may be located at different vertical levels. Having accurate control over altitude and orientation is crucial for stable maneuvering in order to perform its primary tasks in the AVF, often involving monitoring and navigating through the vertical space. While horizontal control is still crucial and worth investigating later, the altitude and the attitude variables for the observer design are the most relevant for the UAV's stability and maneuvering in the AVF. All tests were carried out on a computer equipped with Apple M1 CPU, 16 GB of RAM, and a Matlab R2022b version.

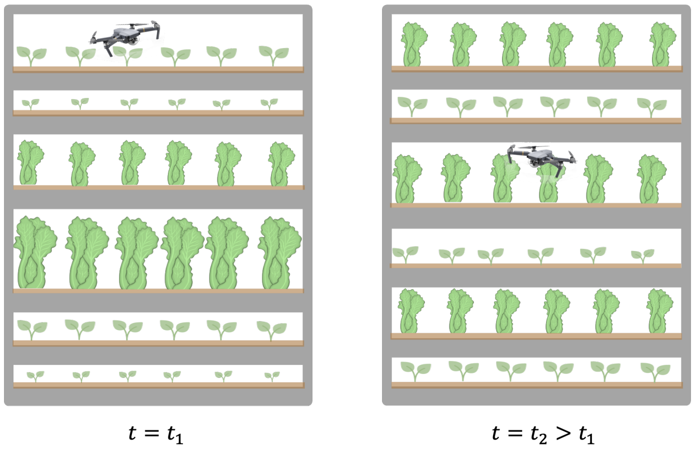


Figure 4.4 – An indoor micro-quadcopter operating within the AVF

4.4.2 Results of the LMI H_∞-Based Observer Design

In this section, we present the results of the simulation performed to evaluate the performance of the proposed LMI $\mathcal{H}_\infty$ -based observer for the UAV. We consider the following state vector:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{bmatrix}^\top$$

with

$$\begin{cases} x_1 = \phi \\ x_2 = \dot{x}_1 = \dot{\phi} \\ x_3 = \theta \\ x_4 = \dot{x}_3 = \dot{\theta} \\ x_5 = \psi \\ x_6 = \dot{x}_5 = \dot{\psi} \\ x_7 = z \\ x_8 = \dot{x}_7 = \dot{z} \end{cases} \tag{4.79}$$

From (4.77)-(4.85), the system can be described as:

$$\begin{cases} \dot{x} = \Gamma(x) + Ew(x) \\ y = Cx \end{cases} \tag{4.80}$$

where $x \in \mathbb{R}^n$ is the system state, $y_t \in \mathbb{R}^n$ the output vector, $E \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{m\times n}$ two matrices, $w \in \mathbb{R}^m$ an external disturbance, and $\Gamma : \mathbb{R}^n \to \mathbb{R}^n$ a nonlinear function defined as follows:

$$\Gamma(x) = \begin{pmatrix} x_2 \\ x_4 x_6 a_1 \\ x_4 \\ x_2 x_6 a_3 \\ x_6 \\ x_4 x_2 a_5 \\ x_8 \\ -g \end{pmatrix}, \qquad E = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.81}$$

$$w(x) = \begin{pmatrix} x_4 a_2 \Omega + b_1 U_2 \\ x_2 a_4 \Omega + b_2 U_3 \\ b_3 U_4 \\ (\cos x_1 \cos x_3) \frac{1}{m} U_1 \end{pmatrix}, \tag{4.82}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{4.83}$$

The constants parameters $a_i,\ i \in \{1,2,3,4,5\}$ and $b_i,\ i \in \{1,2,3\}$ are given by :

$$\begin{array}{ll} a_1 = \frac{I_y - I_z}{I_x} & b_1 = \frac{l}{I_x} \\ a_2 = \frac{-J_r}{I_x} & b_2 = \frac{l}{I_y} \\ a_3 = \frac{I_z - I_x}{I_y} & b_3 = \frac{1}{I_z} \\ a_4 = \frac{J_r}{I_y} & \\ a_5 = \frac{I_x - I_y}{I_z} & \end{array}$$

such as the values of $l,\ m,\ J_r,\ I_x,\ I_y,\ I_z$ used in the simulations assumed from [38] for an OS4 quadcopter, as reported in Table 4.1. Actually, in the system (4.80), the disturbances are assumed to be originated from the rotor dynamics represented by Ω, which accounts for rotor imperfections and environmental factors in the confined space of a vertical farm. The disturbance term $w(x)$ captures the physical interactions between the rotor-induced forces and the UAV's dynamics, as well as the control inputs (U_1, U_2, U_3, U_4). Specifically, terms like $x_4 a_2 \Omega$ and $x_2 a_4 \Omega$ describe gyroscopic coupling and cross-axis interactions caused by the rotation of the UAV's propellers, leading to effects on pitch and roll rates. Additionally, the terms $b_1 U_2$, $b_2 U_3$, and $b_3 U_4$ represent the thrust and moment contributions from the rotors, which can be influenced by actuator imperfections and aerodynamic forces such as drag and localized turbulence.

We consider the initial state and the estimated state defined as $x_0 = \begin{bmatrix} \frac{\pi}{4} & 0 & \frac{\pi}{4} & 0 & \frac{\pi}{4} & 0 & 2\pi & 0 \end{bmatrix}^\top$ and $\hat{x}_0 = \begin{bmatrix} 4 & 0.7 & 9 & 7 & 10 & 10 & 0 & 0 \end{bmatrix}^\top$, respectively. The initial state represents the true configuration of the quadcopter, where it is not at rest at the origin, but at some arbitrary initial position and orientation while the initial estimate reflects the observer's starting guess. The discrepancies between the initial states represent the initial error that the observer needs to overcome.

To assess the effectiveness of the observer, system and measurement noises were introduced, modeled as zero-mean Gaussian distributions with a covariance of 0.01 . The simulation was conducted using the Matlab LMI toolbox with the Yalmip interface, and the optimal solution for the observer design was obtained, resulting in $\mu = 767.0387$.

Figures 4.5 present the comparison between the true system states and the estimated states over time. As observed, the estimates converge to the actual system states, demonstrating the observer's ability to effectively track the system's dynamics. Specifically, the trajectory of the estimated states (represented by the red dashed lines) follows the true system state (blue lines), even in the presence of noise and disturbances. This convergence indicates that the observer is able to provide accurate state estimation despite the uncertainties and disturbances inherent in the system, further highlighting its robustness and reliability in real-world scenarios involving UAVs in dynamic environments such as the AVF.

4.4.3 Results of the Extended Dynamic Output Observer: Discret-time case

In this part, we assess the effectiveness of the discrete-time nonlinear observer with a dynamic extended output presented in Section 4.3.2. While the continuous-time observer design provides valuable theoretical insight, the discrete-time observer adapts the core principles derived from continuous-time systems for practical, real-time implementation as mentionned in Section 4.3.2. For this reason, the simulation was performed in the discrete-time domain, which is the most

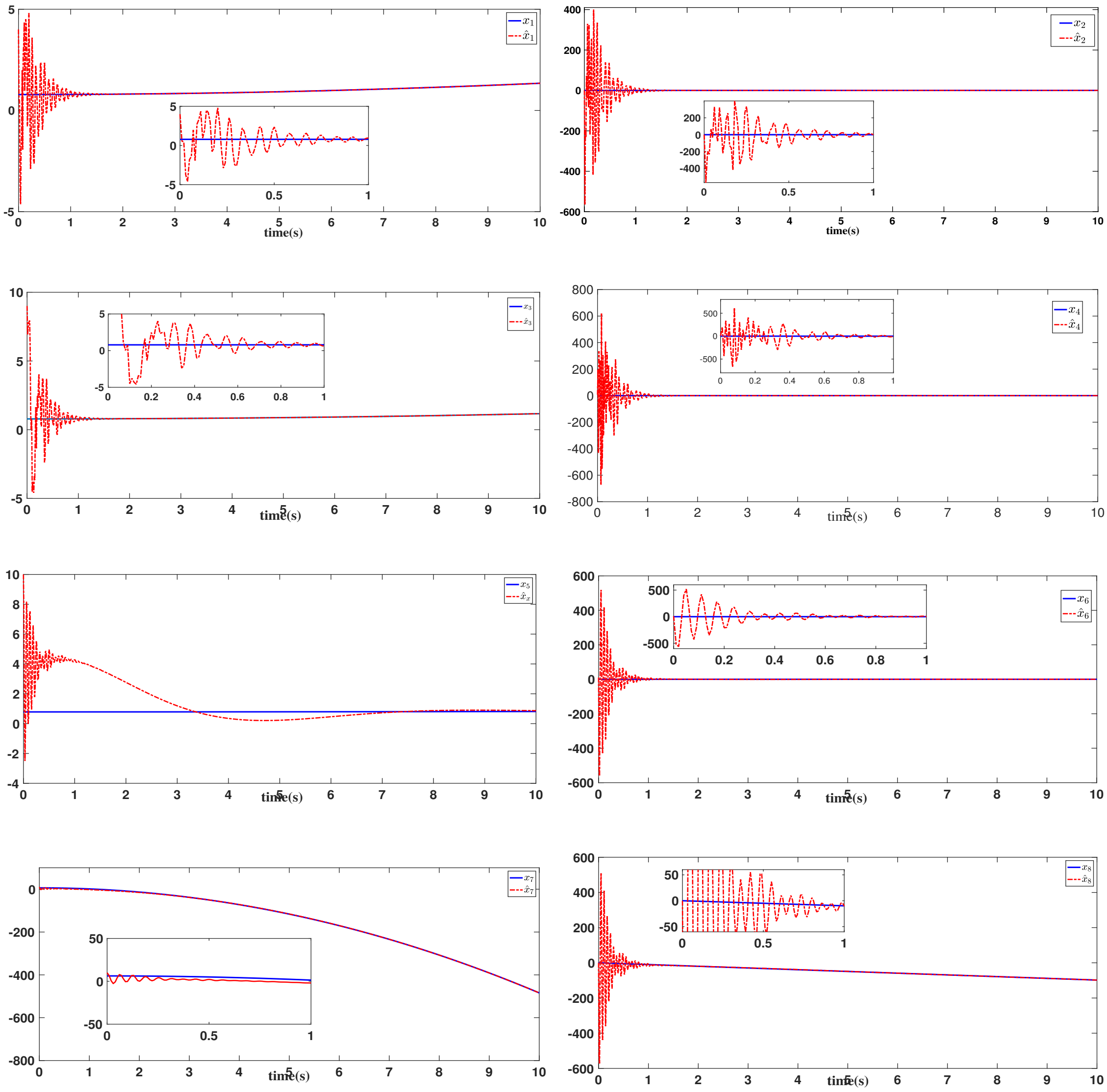


Figure 4.5 – States and their estimates obtained in the simulations with the proposed H_∞-based observer

relevant for the specific application of UAVs in AVFs, ensuring that the observer design aligns with the operational characteristics of the system.
We begin by discretizing the dynamic model (4.77) using Euler's method with a step size of 0.01. The resulting system is given by:

$$\begin{cases} x_{k+1} = \psi(x_k, \omega_k) \\ y_k = \phi(x_k, v_k) \end{cases} \tag{4.84}$$

where $\psi : \mathbb{R}^8 \times \mathbb{R}^8 \to \mathbb{R}^8$, $\phi : \mathbb{R}^8 \times \mathbb{R}^4 \to \mathbb{R}^4$ are nonlinear functions depending on the states x_k, the disturbance $\omega_k = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 & w_8 \end{bmatrix}^\top$ and the measurement noise $v_k = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}^\top$ representing, respectively the airflow faults disturbances and the sensor noise. The state and output vectors are as follows:

$$x_k = \begin{bmatrix} \phi_k & \dot{\phi}_k & \theta_k & \dot{\theta}_k & \psi_k & \dot{\psi}_k & z_k & \dot{z}_k \end{bmatrix}^\top \tag{4.85}$$

$$y = \begin{bmatrix} \phi_k & \theta_k & \psi_k & z_k \end{bmatrix}^\top \tag{4.86}$$

such as ϕ is the roll angle, θ is the pitch angle, ψ is the yaw angle and z is the vertical altitude. $\dot{\phi}$, $\dot{\theta}$, $\dot{\phi}$ are the angular rates and $\dot{z}$ is the vertical acceleration.
We perform the system (4.84) transformation according to the approach detailed in Section 4.3.2. The resulting extended state vector is given by:

$$\begin{cases} \xi_{k+1} = \begin{bmatrix} A_\eta \eta_k + B_\eta \begin{bmatrix} \phi_k + v_1 \\ \theta_k + v_2 \\ \psi_k + v_3 \\ z_k + v_4 \end{bmatrix} \\ \dot{\phi}_k + w_1 \\ \dot{\theta}_k \dot{\psi}_k a_1 + \dot{\theta}_k a_2 \Omega + b_1 u_2 + w_2 \\ \dot{\theta}_k + w_3 \\ \dot{\phi}_k \dot{\psi}_k a_3 + \dot{\phi}_k a_4 \Omega + b_2 u_3 + w_4 \\ \dot{\psi}_k + w_5 \\ \dot{\phi}_k \dot{\theta}_k a_5 + b_3 u_4 + w_6 \\ \dot{z}_k + w_7 \\ -g + \cos(\phi_k)\cos(\theta_k)\frac{u_1}{m} + w_8 \end{bmatrix} \\ \boldsymbol{y}_k = C\xi_k, \quad C = \begin{bmatrix} \mathbb{I}_4 & 0 \end{bmatrix}. \end{cases} \tag{4.87}$$

such as $\mathbb{I}_4$ is an identity matrix of dimension 4.
Simulations are performed on the transformed model (4.87) and made by the mean of Matlab LMI toolbox with Yalmip interface. The introduced disturbances and measurement noise are modeled by zero-mean Gaussian distribution with covariance equal to 0.01 and 0.02, respectively, reflecting the indoor environment. As show the results in Figure 4.6, the convergence of the estimated states highlights the observer's capacity to handle sensor noise and disturbances over $5s$ of simulation time, which is a common temporal window allowing to capture important transient responses for UAVs of fast dynamics especially for small drones like the OS4,.

4.5 Conclusion

In this chapter, to account for the disturbances that the operating UAV may encounter while functioning within the AVF, we propose three distinct nonlinear observer designs. Two of these designs are applied to a continuous-time model of the UAV, while the third is based on a discrete-time model to simplify implementation and improve computational efficiency. The results obtained demonstrate the effectiveness of the proposed approaches

Figure 4.6 – States and their estimates obtained in the simulations with the proposed discrete-time extended dynamic output observer

Conclusions and Future Work

This thesis is part of a larger effort to address global challenges in fresh food production in general and agriculture in particular, mainly in the context of vertical farming and precision agriculture. The Adaptive Vertical Farm (AVF) concept, investigated in the framework of this PhD project, is a cutting-edge technology aiming to optimize the vertical space among cultivation shelves to adapt to crop growth during the production cycle. Thus, the concept is believed to align with global trends toward resource-efficient farming methods to support the global transition toward more sustainable food production. The vegetable food production technology transferred by Space V S.r.l. for space exploration holds significant promise for addressing critical challenges in food production on Earth. Initially designed to sustain astronauts with locally grown fresh food during long-duration missions aboard the International Space Station (ISS), the potential of the AVF is not only confined to space farming applications. The scalability and adaptability of the adaptive multilayer greenhouse make it also a promising solution for overcoming the challenges associated with conventional agriculture, particularly in regions where space, water, and energy are limited. In urban environments, vertical farming systems supported by the adaptive principle can contribute to local food security along with reducing the carbon footprint of food transportation. AVFs can be integrated into existing infrastructure, reducing the need for new land and providing a sustainable means of growing a variety of crops. During three years of this PhD project, we have explored two key development areas to optimize the performance of the innovative multilayer greenhouse: growing crops within the AVF and the possibility of integrating UAVs to enable the crops' continuous monitoring, detecting eventual diseases, assessing irrigation and nutrient needs, allowing for more informed decision-making during the production cycle.

Chapter 1 provides a comprehensive overview of existing vertical farming solutions, covering both the employed paradigms and key technological trends, such as the integration of sensors, actuation systems, aerial robotics platforms for crop monitoring, and artificial intelligence-based techniques for optimizing farm automation. The chapter introduces the motivations and core concepts behind the AVF with respect to the rapid growth of the vertical farming market. The conceptual design and its main subsystems (i.e., subsystems of air conditioning, lighting, fertirrigation, shelf handling, microclimate for each shelf, etc.) are showcased, highlighting the advantages in terms of sustainability and efficiency compared to traditional VFs. In this scope, the chapter also explores the potential integration of UAV platforms into the AVF as a dynamic monitoring subsystem to address current technical challenges related to the dynamic architecture of the AVF. In order to investigate in depth the objectives, a theoretical background is presented in **Chapter 2** for understanding the key concepts applied in the continuation of this work, mainly data-driven models and nonlinear LMI-based observers, essential tools for the results presented in **Chapter 3 and Chapter 4**.

Chapter 3 is devoted to the crop cultivated within the AVF, addressing mainly the crop growth modeling regarding the dynamic nature of the greenhouse. The lettuce growth model is considered, as commonly used for traditional vertical farms due to its rapid growth cycle, making it an ideal candidate for controlled environment systems. This is particulary beneficial in testing and simulation scenarios where clear results are needed in a short time. As a commonly grown crop in vertical farming systems, lettuce shares its growth characteristics with leafy greens, herbs, and microgreens, which allows the model to generalize across a range of similar crops. We have presented two different approaches to model the growth of crops within an AVF. The first one is based on a dynamic model described by a two-ODE system, which depends on several parameters to be properly tuned via an identification procedure based on a least-squares cost to optimize performances. The second approach relies on a data-driven, black-box model having the form of one-hidden-layer feedforward neural networks, which must be trained to obtain an approximation of the crop growth dynamics. Our results indicate that, on the one hand, the ODE-based model provides reliable predictions, but it requires precise parameter tuning. On the other hand, the black-box model offers greater flexibility and adaptability to different crop types (mainly leafy greens, herbs) and environmental conditions. Once trained, it significantly reduces computational effort, making it more suitable for real-time applications in AVFs. However, while the lettuce growth model is effective for certain crops, it has limitations when generalized to other plants. Different species possess unique physiological characteristics and growth requirements. For instance, fruiting plants such as tomatoes, cucumbers, or peppers have a more complex growth cycle that includes fruiting stages, which the lettuce model does not account for. Similarly, root vegetables like carrots, beets, or radishes have growth patterns focused on root development, which would not be fully captured by the lettuce-based model. These differences in growth dynamics mean that the same parameters used for lettuce may not be sufficient to accurately model these plants, requiring species-specific adjustments to the model's parameters.

In order to ensure a bird's-eye view on crops and the growing environment through the monitoring UAVs, an accurate state estimation for the drone is essential. In fact, as AVF environments present unique challenges such sensor noise, disturbances from machinery, varying environmental conditions, and confined spaces, accurate state estimation becomes crucial for UAV to perform the demanded tasks. The key contributions of **Chapter 4** include the development of a observer design based on Linear Matrix Inequality (LMI)-based H_∞ state estimation that relies on improved LMI conditions and an observer design by using a new output-based dynamic extension approach, which was applied in both continuous and discrete cases. The efficiency of the proposed estimated scheme was assessed and showed an ability to maintain accurate estimates in challenging conditions. Within the scope of this thesis, the use of UAVs remained a concept. While this provides valuable insights into the potential applications of UAVs in precision agriculture, it is important to acknowledge that real-world applications introduce several challenges that were not fully captured such as the coordination with the shelves' movements, the trajectory of the drone, etc. It is also worth noting that, the work does not delve into the specific sensors and actuators required for the adaptive aspects of the AVF, apart from the fondamental sensors and actuators of traditional VFs. For example, the potential for real-time adjustments of layer distances to optimize UAV operation or other farming processes is not explored in detail. These limitations in sensor and actuator integration in the context of the AVF are areas for future improvement. It is important to note that during the time of this thesis, the prototype's sensorization and actuation systems were still under development, which limited the ability to include detailed discussions on these components.

To this end, future work will delve into deepening the current works and exploring new pillars of the AVF developement. Among others, we cite :

- Refining the parameter estimation process for the white-box model to enhance its generalizability and robustness considering different crop types (i.e. leafy greens, fruitig plants, and root vegetables).
- Exploring the use of more advanced neural network architectures, such as recurrent or multi-layer networks, to improve the accuracy of the black-box model.
- Coupling the black-box approach with the scheduling algorithms proposed for the AVF (e.g. [28, 29, 31] in order to have an integrated framework for the optimization of the AVF performance.
- Investigating advanced nonlinear estimation techniques, including AI-based estimation methods, and comparing the results to identify the most effective observer design. This could be coupled with a control strategy for both shelf movements and task-specific UAV trajectories.

The AVF is a revolutionary, cutting-edge technology for both terrestrial and space-based agriculture. Researchers from Space V, the University of Genoa, and collaborating institutions, alongside myself, will continue working on its development, with the aim of making a complete functionnal system in very near future.

Bibliography

Bibliography

[1] Global vertical farming market projection 2022-2032. https://www.statista.com/statistics/487666/projection-vertical-farming-market-worldwide/. [Online], accessed 2024.

[2] M. Gaggero L. Mantelli V. Santamaria A. Traverso A. Alessandri, P. Bagnerini. Black-box modeling and optimal control of a two-phase flow using level set methods. *IEEE Trans. Control Systems Technology*, 30(2):520–534, 2022.

[3] M. Sanguineti A. Alessandri, M. Gaggero. Data-driven performance metrics for neural network learning. *International Journal of Adaptive Control and Signal Processing*, 2024, in press.

[4] S. Hemming H.F. de Zwart A. Elings A. Petropoulou, B. van Marrewijk and M. Bijlaard. 3rd autonomous greenhouse challenge: Time-series data on realized climate with annotated crop lettuce-images. https://doi.org/10.4121/21960932.v1. [Online], accessed 2023.

[5] S. Hemming H.F. de Zwart A. Elings A. Petropoulou, B. van Marrewijk and M. Bijlaard. Lettuce production in intelligent greenhouses: 3D imaging and computer vision for plant spacing decisions. *Sensors*, 23:2929, 2023.

[6] N.M. Salma S. Afif R.E. Masithoh L. Sutiarso A. Rizkiana, A.P. Nugroho and T. Okayasu. Plant growth prediction model for lettuce (lactuca sativa.) in plant factories using artificial neural network. In *IOP Conference Series: Earth and Environmental Science*, volume 733, page 012027. IOP Publishing, 2021.

[7] B. Boulkroune H. Rafaralahy A. Zemouche, R. Rajamani and M. Zasadzinski. Convex optimization based dual gain observer design for lipschitz nonlinear systems. In *2016 American Control Conference (ACC)*, pages 125–130, 2016.

[8] F. Mazenc A. Zemouche, F. Zhang and R. Rajamani. High-gain nonlinear observer with lower tuning parameter. *IEEE Transactions on Automatic Control*, 64(8):3194–3209, 2018.

[9] F. Mazenc A. Zemouche, F. Zhang and R. Rajamani. High-gain nonlinear observer with lower tuning parameter. *IEEE Transactions on Automatic Control*, 64(8):3194–3209, 2019.

[10] B. Acikmese and M. Corless. Observers for systems with nonlinearities satisfying incremental quadratic constraints. *Automatica*, 47(7):1339–1348, 2011.

[11] A. Adil, A. Hamaz, I. N'doye, A. Zemouche, T. M. Laleg-Kirati, and F. Bedouhene. On high-gain observer design for nonlinear systems with delayed output measurements. *Automatica*, 141:110281, 2022.

[12] Aerofarms. https://www.aerofarms.com, 2023. [Online], accessed 2024.

[13] K. Al-Kodmany. The vertical farm: a review of developments and implications for the vertical city. Buildings 8 (2): 24, 2018.

[14] A. Alessandri, M. Baglietto, and G. Battistelli. Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica*, 44(7):1753–1765, 2008.

[15] A. Alessandri, M. Baglietto, G. Battistelli, and M. Gaggero. Moving-horizon state estimation for nonlinear systems using neural networks. *IEEE Transactions on Neural Networks*, 22(5):768–780, 2011.

[16] A. Alessandri, M. Baglietto, G. Battistelli, and V. Zavala. Advances in moving horizon estimation for nonlinear systems. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5681–5688. IEEE, 2010.

[17] A. Alessandri, M. Cuneo, S. Pagnan, and M. Sanguineti. A recursive algorithm for nonlinear least-squares problems. *Computational Optimization and Applications*, 38(2):195–216, 2007.

[18] A. Alessandri and A. Rossi. Time-varying increasing-gain observers for nonlinear systems. *Automatica*, 49(9):2845–2852, 2013.

[19] A. Alessandri, M. Zasadzinski, and A. Zemouche. Optimistic vs pessimistic moving-horizon estimation for quasi–lpv discrete-time systems. *IFAC-PapersOnLine*, 53(2):5004–5009, 2020.

[20] N. Ampazis and S.J. Perantonis. Two highly efficient second-order algorithms for training feedforward networks. *IEEE Transactions on Neural Networks*, 13(5):1064–1074, 2002.

[21] V. Andrieu, L. Praly, and A. Astolfi. High gain observers with updated gain and homogeneous correction terms. *Automatica*, 45(2):422–428, 2009.

[22] M. Arcak and P. Kokotovic. Observer-based control of systems with slope-restricted nonlinearities. *IEEE Transactions on Automatic Control*, 46(7):1146–1150, 2001.

[23] H. Arezki, A. Alessandri, and A. Zemouche. Robust stability analysis of moving-horizon estimator for LPV discrete-time systems. In *22nd IFAC World Congress*, 2023.

[24] D. Astolfi and L. Marconi. A high-gain nonlinear observer with limited gain power. *IEEE Transactions on Automatic Control*, 60(11):3059–3064, 2015.

[25] P. Giuliani T. Herman L. Barnes B. Franchetti, V. Ntouskos and F. Pirri. Vision based modeling of plants phenotyping in vertical farming under artificial lighting. *Sensors*, 19(20):4378, 2019.

[26] I. Lebedev B. Gubanov, V. Lebedeva and M. Astapova. Algorithms and software for evaluation of plant height in vertical farm using UAVs. In *Agriculture Digitalization and Organic Production: Proceedings of the Second International Conference, ADOP 2022, St. Petersburg, Russia, June 06–08, 2022*, pages 351–362. Springer, 2022.

[27] P. Bagnerini, E. Chnib, M. Gaggero, and A. Zemouche. Adaptive vertical farm for space farming. In *Applied Automation Technology in Teaching and Development, AALE 2023*, 2023.

[28] P. Bagnerini, M. Gaggero, and M. Ghio. Mixed-integer linear programming for the scheduling of seedings in an industrial adaptive vertical farm. In *Proc. IEEE Conf. on Automation Science and Engineering*, pages 1–6, 2023.

[29] P. Bagnerini, M. Gaggero, and M. Ghio. Model predictive control for the scheduling of seedings in an adaptive vertical farm. In *Proc. IEEE Conf. on Decision and Control*, pages 4303–4308, 2023.

[30] P. Bagnerini, M. Gaggero, M. Ghio, and M.A. Malerba F. Malerba. Adaptive vertical farm for fresh food production in orbital stations and future lunar settlements. In *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 402–407. IEEE, 2022.

[31] P. Bagnerini, M. Gaggero, M. Ghio, and F. Malerba. The adaptive vertical farm as an efficient tool for the cultivation of multiple crops in space. In *Proc. Int. Workshop on Metrology for Aerospace*, pages 231–236, 2023.

[32] E. W. Bai and S. Sastry. Observability of nonlinear systems: A geometric approach. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1249–1254. IEEE, 2005.

[33] H.B. Barlow. Unsupervised learning. *Neural Computation*, 1(3):295–311, 1989.

[34] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory*, 39(3):930–945, 1993.

[35] G. Besançon. Remarks on nonlinear adaptive observer design. *Systems & control letters*, 41(4):271–280, 2000.

[36] F.H. Besthorn. Vertical farming: Social work and sustainable urban agriculture in an age of global food crises. *Australian Social Work*, 66(2):187–203, 2013.

[37] J. Birkby. Vertical farming. *ATTRA Sustainable Agriculture*, 2(1):1–12, 2016.

[38] S. Bouabdallah. Design and control of quadrotors with application to autonomous flying. *Project Report, Ecole Polytechnique*, 2007.

[39] S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2247–2252, 2005.

[40] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[41] C.Edwards and S. Spurgeon. *Sliding mode control: theory and applications*. Crc Press, 1998.

[42] L.H. Chiang, E.L. Russell, and R.D. Braatz. Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 50(2):243–252, 2000.

[43] E. Chnib, P. Bagnerini, M. Gaggero, and A. Zemouche. LMI based $\mathcal{H}_\infty$ observer design for a quadcopter model operating in an adaptive vertical farm. *IFAC-PapersOnLine*, 56(2):10837–10842, 2023.

[44] E. Chnib, P. Bagnerini, M. Gaggero, and A. Zemouche. Observer design by using a new output-based dynamic extension technique. In *2024 European Control Conference (ECC)*, pages 1736–1740. IEEE, 2024.

[45] E. Chnib, P. Bagnerini, M. Gaggero, and A. Zemouche. Parameter estimation of a dynamic growth model for lettuce in an adaptive vertical farm. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 1169–1174. IEEE, 2024.

[46] M. Cıceklı and N.T. Barlas. Transformation of today greenhouses into high-technology vertical farming systems for metropolitan regions. *Journal of Environmental Protection and Ecology*, 15(3):1066–1073, 2014.

[47] Space V S.r.l. company. https://www.spacev.bio. [Online], accessed 2024.

[48] P. Cunningham, M. Cord, and S.J. Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008.

[49] D. Despommier. *The vertical farm: feeding the world in the 21st century*. Macmillan, 2010.

[50] D. Despommier. Farming up the city: The rise of urban vertical farms. In *Trends in Biotechnology*, volume 31, pages 388–389. Elsevier, 2013.

[51] S.X. Ding. *Data-driven design of fault diagnosis and fault-tolerant control systems*. Springer, 2014.

[52] Corvus Drones. https://www.corvusdrones.com, 2023. [Online], accessed 2024.

[53] S. Heo Y.S. Chung E.B.M. Karunathilake, A.T. Le and S. Mansoor. The path to smart farming: Innovations and opportunities in precision agriculture. *Agriculture*, 13(8):1593, 2023.

[54] C. Eigenbrod and N. Gruda. Urban vegetable for food security in cities. a review. *Agronomy for Sustainable Development*, 35:483–498, 2015.

[55] A. Mahmoudi Lahijani F. Kalantari, O. Mohd Tahir and S. Kalantari. A review of vertical farming technology: a guide for implementation of building integrated agriculture in cities. In *Advanced Engineering Forum*, volume 24, pages 76–91. Trans Tech Publ, 2017.

[56] R.A. Joni F. Kalantari, O.M. Tahir and E. Fatemi. Opportunities and challenges in sustainability of vertical farming: A review. *Journal of Landscape Ecology*, 11(1):35–60, 2018.

[57] R. Fletcher. Practical methods of optimization. *John Wiley & Sons*, 2000.

[58] M.J. Florin et al. *Towards precision agriculture for whole farms using a combination of simulation modelling and spatially dense soil and crop information*. University of Sydney, 2008.

[59] Food and Agriculture Organization of the United Nations. *The future of food and agriculture: Trends and challenges*. Fao, 2017.

[60] J. P. Gauthier, H. Hammouri, and S. Othman. A simple observer for nonlinear systems, applications to bioreactors. *IEEE Transactions on Automatic Control*, 37(6):875–880, 1992.

[61] J. P. Gauthier, H. Hammouri, and S. Othman. A simple observer for nonlinear systems applications to bioreactors. *IEEE Transactions on automatic control*, 37(6):875–880, 1992.

[62] J. P. Gauthier and I.A.K. Kupka. Observability and observers for nonlinear systems. *SIAM J. on Control and Optimization*, 32(4):975–994, 1994.

[63] J.P. Gauthier and I. Kupka. *Deterministic Observation Theory and Applications*. Cambridge University Press, Cambridge, UK, 2001.

[64] J.P. Gauthier and I. A. K. Kupka. Observability and observers for nonlinear systems. *SIAM journal on control and optimization*, 32(4):975–994, 1994.

[65] S. von Caemmerer G.D. Farquhar and J.A. Berry. A biochemical model of photosynthetic co2 assimilation in leaves of c3 species. *Planta*, 149:78–90, 1980.

[66] J.A. Carver M.E. Hummerick R.M. Wheeler R.C. Morrow G.D. Massa, N.F. Dufour and T.M. Smith. Veg-01: Veggie hardware validation testing on the international space station. *Open Agriculture*, 2(1):33–41, 2017.

[67] Z. Ge and Z. Song. *Multivariate statistical process control: Process monitoring methods and applications*. Springer Science & Business Media, 2012.

[68] Z. Ge, Z. Song, and F. Gao. Review of recent research on data-based process monitoring. *Industrial & Engineering Chemistry Research*, 52(10):3543–3562, 2013.

[69] R. Gebbers and V.I. Adamchuk. Precision agriculture and food security. *Science*, 327(5967):828–831, 2010.

[70] M. Gentry. Local heat, local food: Integrating vertical hydroponic farming with district heating in sweden. *Energy*, 174:191–197, 2019.

[71] M. Ghio. System for the vegetable garden and nursery cultivation of plants, October 22 2024. US Patent 12,120,995.

[72] J. Goudriaan, H.H. Van Laar, H. Van Keulen, and W. Louwerse. Photosynthesis, CO2 and plant production. pages 107–122, 1985.

[73] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley, 2001.

[74] M.W. Naouar H. Berriri and I. Slama-Belkhodja. Easy and fast sensor fault detection and isolation algorithm for electrical drives. *IEEE Transactions on Power Electronics*, 27(2):490–499, 2011.

[75] X. Zhang H. Hua, Y. Fang and B. Lu. A novel robust observer-based nonlinear trajectory tracking control strategy for quadrotors. *IEEE Transactions on Control Systems Technology*, 29(5):1952–1963, 2020.

[76] A. Zemouche H. Kheloufi, F. Bedouhene and A. Alessandri. Observer-based stabilisation of linear systems with parameter uncertainties by using enhanced LMI conditions. *International Journal of Control*, 88(6):1189–1200, 2015.

[77] T. Poggio H. Mhaskar, Q.L. Liao. Learning real and boolean functions: When is deep better than shallow. Technical report, Center for Brains, Minds and Machines, Memo No. 045, USA, 2016.

[78] M. Hagan and M. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Trans. Neural Networks*, 5:989–993, 1994.

[79] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, 1998.

[80] Simon Haykin. *Communication Systems*. Wiley, Hoboken, NJ, 5th edition, 2008.

[81] R.G Healy and J.S Rosenberg. *Land use and the states*. RFF Press, 2013.

[82] E.J. Van Henten. Validation of a dynamic lettuce growth model for greenhouse climate control. In *Agricultural Systems*, volume 45, pages 55–72, 1994.

[83] E.J. Van Henten and G. Van Straten. Sensitivity analysis of a dynamic growth model of lettuce. In *Journal of Agricultural Engineering Research*, volume 59, pages 19–31, 1994.

[84] R. Hermann and J. P. Gauthier. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, 1977.

[85] John H. Hubbard and Beverly H. West. *Differential Equations: A Dynamical Systems Approach*. Springer-Verlag, New York, 1995.

[86] iFarm. https://ifarm.fi/, 2023. [Online], accessed 2024.

[87] E. Fitz-Rodríguez R. Salazar-Moreno I.L. López-Cruz, A. Ruiz-García and A. Rojano-Aguilar. A comparison of bayesian and classical methods for parameter estimation in greenhouse crop models. In *V International Symposium on Models for Plant Growth, Environment Control and Farming Management in Protected Cultivation*, volume 1182, pages 241–248, 2016.

[88] R. Ilin, R. Kozma, and P.J. Werbos. Beyond feedforward models trained by backpropagation: A practical training tool for a more efficient universal approximator. *IEEE Transactions on Neural Networks*, 19(6):929–937, 2008.

[89] XROOTS: Growing Plants in Space. https://science.nasa.gov/biological-physical/investigations/xroots/. [Online], accessed December 16, 2024.

[90] T.D. Do T. Dragicevic I.S. Mohamed, S. Rovetta and A.A. Zaki Diab. A neural-network-based model predictive control of three-phase inverter with an output lc filter. *IEEE Access*, 7:124737–124749, 2019.

[91] A. Isidori. *Nonlinear Control Systems*. Springer, London, 3rd edition, 1995.

[92] F. Asch J. de Boer J. Schreiber-G. Weber J. Germer, J. Sauerborn and J. Müller. Skyfarming an ecological innovation to enhance global food security. *Journal für Verbraucherschutz und Lebensmittelsicherheit*, 6:237–251, 2011.

[93] J. Wang J. Peng and Y. Wang. Neural network based robust hybrid control for robotic system: An hâàû approach. *Nonlinear Dynamics*, 65:421–431, 2011.

[94] R.A. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1(4):295–307, 1988.

[95] G. James, D. Witten, T. Hastie, R. Tibshirani, et al. *An introduction to statistical learning*, volume 112. Springer, 2013.

[96] J.B. Jones Jr. *Hydroponics: a practical guide for the soilless grower*. CRC Press, 2016.

[97] X. Yu L. Guo K. Guo, J. Jia and L. Xie. Multiple observers based anti-disturbance control for a quadrotor uav against payload and wind disturbances. *Control Engineering Practice*, 102:104560, 2020.

[98] V. Kurková and M. Sanguineti. Comparison of worst-case errors in linear and neural network approximation. *IEEE Trans. Inf. Theory*, 48(1):264–275, 2002.

[99] P. Kadlec, B. Gabrys, and S. Strandt. Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, 33(4):795–814, 2009.

[100] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[101] Thomas Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, New Jersey, 2000.

[102] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

[103] R.E. Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[104] M. Kano and M. Ogawa. The state of the art in chemical process control in japan: Good practice and questionnaire survey. *Journal of Process Control*, 20(9):969–982, 2010.

[105] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2002.

[106] S. Khatibisepehr, B. Huang, and S. Khare. Design of inferential sensors in the process industry: A review of bayesian methods. *Journal of Process Control*, 23(10):1575–1596, 2013.

[107] H. Kheloufi, F Bedouhene, A Zemouche, and A. Alessandri. Observer-based stabilisation of linear systems with parameter uncertainties by using enhanced LMI conditions. *International Journal of Control*, 88(6):1189–1200, 2015.

[108] D. Kurtulus and T. H. Lee. *State Observers: Theory and Applications*. Springer, Berlin, 2004.

[109] Z.G. Hou J. Yu L. Cheng, W. Liu and M. Tan. Neural-network-based nonlinear model predictive control for piezoelectric actuators. *IEEE Transactions on Industrial Electronics*, 62(12):7717–7727, 2015.

[110] A. Van Den Dobbelsteen L. Graamans, M. Tenpierik and C. Stanghellini. Plant factories: Reducing energy demand at high internal heat loads through facade design. *Applied Energy*, 262:114544, 2020.

[111] A. van den Dobbelsteen I. Tsafaras L. Graamans, E. Baeza and C. Stanghellini. Plant factories versus greenhouses: Comparison of resource use efficiency. *Agricultural Systems*, 160:31–43, 2018.

[112] M. Boutayeb L. Hassan, A. Zemouche. H_∞ observers design for a class of nonlinear time-delay systems in descriptor form. *International Journal of Control*, 84(10):1653–1663, 2011.

[113] M. Boutayeb L. Hassan, A. Zemouche. Robust unknown input observers for nonlinear time-delay systems. *SIAM Journal on Control and Optimization*, 51(4):2735–2752, 2013.

[114] J.M. Sousa L.F. Mendonand J.M.G. Sá da Costa. An architecture for fault detection and isolation based on fuzzy methods. *Expert Systems with Applications*, 36(2):1092–1104, 2009.

[115] L. Ljung. Asymptotic behaviour of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Trans. on Automatic Control*, AC-24(1):36–50, February 1979.

[116] T. Ma J.E.V. Johnson L.O. Orimoloye, M-C. Sung. Comparing the effectiveness of deep feedforward neural networks and shallow architectures for predicting stock price indices. *Expert Systems With Applications*, 139:1–18, 2020. art.no. 112828.

[117] D. G. Luenberger. *An Introduction to Observers*, volume 16. IEEE, 1971.

[118] M. Cole M. Eaton, T. Shelford and N. Mattson. Modeling resource consumption and carbon emissions associated with lettuce production in plant factories. *Journal of Cleaner Production*, page 135569, 2022.

[119] O. Celikel M. Siropyan and O. Pinarer. Artificial intelligence driven vertical farming management system. In *Proceedings of the World Congress on Engineering 2022*, 2022.

[120] J. MacGregor and A. Cinar. Monitoring, fault diagnosis, fault-tolerant control and optimization: Data-driven methods. *Computers & Chemical Engineering*, 47:111–120, 2012.

[121] C. C. Malveaux. Vertical take off and landing autonomous/semiautonomous/remote controlled aerial agricultural sensor platform, September 18 2014. US Patent App. 13/845,123.

[122] P. S. Maybeck. *Stochastic Models, Estimation, and Control*. Academic Press, 1979.

[123] T. Mazumdar. Generalized projection theorem and weak noncoercive evolution problems in Hilbert space. *Journal of Mathematical Analysis and Applications*, 46(1):143–168, 1974.

[124] N.M. Hamdan M.H.M. Saad and M.R. Sarker. State of the art of urban smart vertical farming automation system: Advanced topologies, issues and recommendations. *Electronics*, 10(12):1422, 2021.

[125] M. Chowdhury M. Ali S. Samsuzzaman M.R. Ali K.Y. Lee M.S.N. Kabir, M. Reza and S.-O. Chung. Technological trends and engineering issues on vertical farms: a review. *Horticulturae*, 9(11):1229, 2023.

[126] M. Wang N. Zhang and N. Wang. Precision agriculture-a worldwide overview. *Computers and Electronics in Agriculture*, 36(2-3):113–132, 2002.

[127] G. De Nicolao and G. Ferrari-Trecate. Regularization networks: Fast weight calculation via kalman filtering. *IEEE Transactions on Neural Networks*, 12(2):228–235, 2001.

[128] C. C. S. Nicole, F. Charalambous, S. Martinakos, S. Van De Voort, Z. Li, M. Verhoog, and MPCM. Krijn. Lettuce growth and quality optimization in a plant factory. In *VIII International Symposium on Light in Horticulture 1134*, pages 231–238, 2016.

[129] H. Nijmeijer and A. J. Van der Schaft. *Nonlinear Dynamical Control Systems*. Springer, New York, 1990.

[130] K. Ogata. *Modern Control Engineering*. Prentice Hall, New Jersey, 5th edition edition, 2010.

[131] F. Orsini, G. Pennisi, F. Zulfiqar, G. Gianquinto, et al. Sustainable use of resources in plant factories with artificial lighting (PFALs). *European Journal of Horticultural Science*, 85(5):297–309, 2020.

[132] M. Gaggero P. Bagnerini and M. Ghio. Mixed-integer linear programming for the scheduling of seedings in an industrial adaptive vertical farm. In *Proc. IEEE Conf. on Automation Science and Engineering*, pages 1–6, 2023.

[133] M. Gaggero P. Bagnerini and M. Ghio. Model predictive control for the scheduling of seedings in an adaptive vertical farm. In *Proc. IEEE Conf. on Decision and Control*, pages 4303–4308, 2023.

[134] M. Ghio P. Bagnerini, M. Gaggero and F. Malerba. The adaptive vertical farm as an efficient tool for the cultivation of multiple crops in space. In *Proc. Int. Workshop on Metrology for Aerospace*, pages 231–236, 2023.

[135] M.A. Hussain P. Kittisupakorn, P. Thitiyasook and W. Daosud. Neural network based model predictive control for a steel pickling process. *Journal of Process Control*, 19(4):579–590, 2009.

[136] R.K. Mahendran S. Naseer M. Shafiq P. Velusamy, S. Rajendran and J.-G. Choi. Unmanned aerial vehicles (uav) in precision agriculture: Applications and challenges. *Energies*, 15(1):217, 2021.

[137] K. Patan. Neural network-based model predictive control: Fault tolerance and stability. *IEEE Transactions on Control Systems Technology*, 23(3):1147–1155, 2014.

[138] A. Perrusquía and W. Yu. Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview. *Neurocomputing*, 438:145–154, 2021.

[139] C. Peterson. Advanced aeroponics, 2012.

[140] G.H. Buck-Sorlin P.H.B. De Visser and G.W.A.M. van Der Heijden. Optimizing illumination in the greenhouse using a 3d model of tomato and a ray tracer. *Frontiers in Plant Science*, 5:48, 2014.

[141] J.D. Pineda-Jaramillo. A shallow neural network approach for identifying the leading causes associated to pedestrian deaths in medellin. *J. Transp. Health*, 6:1–18, 2020. art.no. 100912.

[142] A. Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–196, 1999.

[143] Y. Yu Q. Wang, J.-W. Wang and C.-Y. Sun. Robust attitude control of an indoor micro quadrotor with input delay. In *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, pages 2363–2368. IEEE, 2014.

[144] S.J. Qin. Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36(2):220–234, 2012.

[145] R.A. Yunmar R. Uswarman, S. Istiqphara and A.Z. Rakhman. Robust control of a quadcopter flying via sliding mode. *Journal of Science and Applicative Technology*, 2(1):135–143, 2019.

[146] T. Parisini R. Zoppoli, M. Sanguineti. Approximating networks and the extended ritz method for the solution of functional optimization problems. *J. of Optim. Theory and Applications*, 112(2):403–439, 2002.

[147] S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.

[148] I. Malounas E. Rodias C. Hellmann Santos S. Fountas, N. Mylonas and E. Pekkeriet. Agricultural robotics for field operations. *Sensors*, 20(9):2672, 2020.

[149] J. J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, NJ, 1991.

[150] Y. Song and J. W. Grizzle. The extended Kalman filter as a local asymptotic obeserver for nonlinear discrete-time systems. In *Proc. 1992 ACC*, pages 3365–3369, Chicago, USA, 1992.

[151] E. D. Sontag. Smooth stabilization implies coprime factorization. *IEEE transactions on automatic control*, 34(4):435–443, 1989.

[152] E. D. Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and optimal control theory*, pages 163–220. Springer, 2008.

[153] E. D. Sontag and Y. Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.

[154] E.D. Sontag and Yuan. W. New characterizations of input-to-state stability. *IEEE Transactions on Automatic Control*, 41(9):1283–1294, 1996.

[155] H. W. Sorenson. *Kalman Filtering Theory and Application*. IEEE Press, 1990.

[156] S. K. Spurgeon. Sliding mode observers: a survey. *International Journal of Systems Science*, 39(8):751–764, 2008.

[157] Germina S.r.l. https://www.germina.bio/. [Online], accessed 2024.

[158] D.G. Sweeney. Modelling the growth of winter lettuce. In *Mathematics and Plant Physiology*, pages 217–229, 1981.

[159] K.H. Low T. Thanaraj and B.F. Ng. Actuator fault detection and isolation on multi-rotor uav using extreme learning neuro-fuzzy systems. *ISA Transactions*, 138:168–185, 2023.

[160] D. Ichalal T.H. Pham and S. Mammar. Lpv and nonlinear-based control of an autonomous quadcopter under variations of mass and moment of inertia. *IFAC-PapersOnLine*, 52(28):176–183, 2019.

[161] J.H.M. Thornley and R.G. Hurd. An analysis of the growth of young tomato plants in water culture at different light integrals and CO2 concentrations: II. A mathematical model. *Annals of Botany*, 38(2):389–400, 1974.

[162] F. A. Valentine. A Lipschitz condition preserving extension for a vector function. *American Journal of Mathematics*, 67(1):83–93, 1945.

[163] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A review of process fault detection and diagnosis: Part iii: Process history based methods. *Computers & Chemical Engineering*, 27(3):327–346, 2003.

[164] A. Ju'N.T. Kaufman V.M. Fernández-Cabanyás, L. Pérez-Urrestarazu and J.A. Gross. Comparative analysis of horizontal and vertical decoupled aquaponic systems for basil production and effect of light supplementation by led. *Agronomy*, 10(9):1414, 2020.

[165] D. Waldron. Evolution of vertical farms and the development of a simulation methodology. *WIT Transactions on Ecology and the Environment*, 217:975–986, 2018.

[166] X.Z. Wang. *Data mining and knowledge discovery for process monitoring and control*. Springer Science & Business Media, 2012.

[167] R. M. Wheeler. Nasa s contributions to vertical farming. In *XXXI International Horticultural Congress (IHC2022): International Symposium on Advances in Vertical Farming 1369*, pages 1–14, 2022.

[168] J. C. Willems. Dissipative dynamical systems part i: General theory. *Archive for Rational Mechanics and Analysis*, 45(5):321–351, 1972.

[169] W. Gao X. Guo W. Zhang L. Zheng X. Hao, J. Jia and M. Wang. MFC-CNN: An automatic grading scheme for light stress levels of lettuce (lactuca sativa l.) leaves. *Computers and Electronics in Agriculture*, 179:105847, 2020.

[170] F. Zhang A. Zemouche G. Chen X. Wang, H. Su. Interval observer design and consensus of multiagent systems with time-varying interval uncertainties. *SIAM Journal on Control and Optimization*, 59(5):3392–3417, 2021.

[171] L. Fridman Y. Shtessel, E. C. Yuri and A. Levant. *Sliding mode control and observation*, volume 10. Springer, 2014.

[172] S. Yin, S.X. Ding, X. Xie, and H. Luo. A review on basic data-driven approaches for industrial process monitoring. *IEEE Transactions on Industrial Electronics*, 61(11):6418–6428, 2014.

[173] F. Gao W. Wang L. Fu Z. Wu, R. Yang and R. Li. Segmentation of abnormal leaves of hydroponic lettuce based on deeplabv3+ for robotic sorting. *Computers and Electronics in Agriculture*, 190:106443, 2021.

[174] A. Zemouche. *Sur l'observation de l'état des systèmes dynamiques non linéaires*. PhD thesis, 2007.

[175] A. Zemouche and M. Boutayeb. A unified adaptive observer synthesis method for a class of systems with both lipschitz and monotone nonlinearities. *Systems & Control Letters*, 58(4):282–288, 2009.

[176] A. Zemouche and M. Boutayeb. On LMI conditions to design observers for Lipschitz nonlinear systems. *Automatica*, 49(2):585–591, 2013.

[177] A. Zemouche and R. Rajamani. LMI-based observer design for non-globally Lipschitz systems using Kirszbraun–Valentine extension theorem. *IEEE Control Systems Letters*, 6:2617–2622, 2022.

[178] A. Zemouche and R. Rajamani. Observer design for non-globally Lipschitz nonlinear systems using Hilbert projection theorem. *IEEE Control Systems Letters*, 6:2581–2586, 2022.

[179] A. Zemouche, R. Rajamani, H. Trinh, and M. Zasadzinski. A new LMI based H_∞ observer design method for lipschitz nonlinear systems. In *2016 European Control Conference (ECC)*, pages 2011–2016, 2016.

[180] D.W. Zingg and T.T. Chisholm. Runge-kutta methods for linear ordinary differential equations. *Applied Numerical Mathematics*, 31(2):227–238, 1999.

Abstract

The world's population is projected to approach 10 billion by 2050, driving an expected rise in food demand due to population growth, economic development, and urbanization. To meet this demand sustainably, greenhouse systems, particularly vertical farming, have emerged as a promising solution, offering high crop yields per unit of cultivation area. The *Adaptive Vertical Farm* (AVF), is an innovative industrial vertical greenhouse that dynamically adjusts the distance between its stacked shelves, optimizing growing conditions as plants progress through their growth stages. This adaptive principle overcomes the traditional conflict between maintaining optimal conditions and minimizing energy consumption.

This thesis presents two main research axes contributing to the development of the AVF. The first axis focuses on developing a data-driven, black-box growth model for crops cultivated in AVFs. Given the dynamic adaptation to plant growth, an accurate crop growth model is essential for optimizing shelf movement and system control. While traditional dynamic growth models often rely on numerous parameters and are specific to certain crop types, we propose a black-box approach using feedforward neural networks to predict plant height at each time step. This model is adaptable to various crop types, computationally efficient after training, and particularly suitable for innovative vertical farming systems like the AVF. The effectiveness of the model is illustrated through synthetic and real-world datasets, showcasing its potential in optimizing crop production.

The second research axis focuses on automating the AVF using Unmanned Aerial Vehicles (UAVs) within the context of Precision Agriculture. UAVs support applications such as crop health monitoring, automatic pollination, spraying, and irrigation, optimizing farm operations across stacked shelves and complementing the existing stationary sensors. To support this automation, we introduce observer designs for nonlinear systems using Linear Matrix Inequalities (LMIs) to provide accurate state estimation for UAVs, ensuring exponential convergence of the observer. Two key contributions are presented: first, a new, less conservative LMI condition applied to solve the $\mathcal{H}_\infty$ circle criterion design, and second, a nonlinear observer design based on output dynamic extension. This method minimizes the impact of measurement noise and guarantees the Input-to-State Stability (ISS) property of the estimation error via a novel LMI condition.

Key words: Adaptive Vertical Farm (AVF), Black-box approach, Crop Growth Model, Data-driven Modeling, Feedforward Neural Networks, Optimization, UAVs, Nonlinear Observer Design, Linear Matrix Inequalities (LMIs), Input-to-State Stability (ISS).

Riassunto

La popolazione mondiale dovrebbe avvicinarsi ai 10 miliardi entro il 2050, con un aumento previsto della domanda alimentare dovuto alla crescita demografica, allo sviluppo economico e all'urbanizzazione. Per soddisfare questa domanda in modo sostenibile, i sistemi di coltivazione in serra, in particolare l'agricoltura verticale, sono emersi come una soluzione promettente, offrendo alti rendimenti per unità di superficie coltivata. La *Serra Verticale Adattiva* (SVA), è una serra verticale industriale innovativa che regola dinamicamente la distanza tra i suoi scaffali impilati, ottimizzando le condizioni di crescita man mano che le piante progrediscono nei loro stadi di sviluppo. Questo principio adattativo supera il conflitto tradizionale tra il mantenimento delle condizioni ottimali e la minimizzazione del consumo energetico.
Questa tesi presenta due principali assi di ricerca che contribuiscono allo sviluppo della SVA. Il primo asse si concentra sullo sviluppo di un modello di crescita delle colture basato su dati e un approccio a scatola nera per le piante coltivate nelle SVA. Data l'adattabilità dinamica alla crescita delle piante, un modello di crescita preciso è essenziale per ottimizzare il movimento degli scaffali e il controllo del sistema. Mentre i modelli di crescita dinamici tradizionali spesso dipendono da numerosi parametri e sono specifici per determinati tipi di colture, proponiamo un approccio a scatola nera usando reti neurali feedforward per prevedere l'altezza delle piante ad ogni passo. Questo modello è adattabile a diversi tipi di colture, computazionalmente efficiente dopo l'addestramento e particolarmente adatto a sistemi di agricoltura verticale innovativi come la SVA. L'efficacia del modello è dimostrata attraverso set di dati sintetici e reali, mostrando il suo potenziale nell'ottimizzazione della produzione agricola.
Il secondo asse di ricerca si concentra sull'automazione della SVA utilizzando veicoli aerei senza pilota (UAV) nel contesto dell'agricoltura di precisione. Gli UAV supportano applicazioni come il monitoraggio della salute delle colture, la pollinizzazione automatica, la nebulizzazione e l'irrigazione, ottimizzando le operazioni agricole attraverso gli scaffali impilati e complementando i sensori esistenti. Per supportare questa automazione, proponiamo la progettazione di osservatori per sistemi non lineari usando Inuguaglianze Matriciali Lineari (LMI) per fornire una stima precisa dello stato degli UAV, garantendo la convergenza esponenziale dell'osservatore. Due contributi chiave sono presentati: prima, una nuova condizione LMI meno conservativa applicata per risolvere il criterio circolare $\mathcal{H}_\infty$, e in secondo luogo, una progettazione di osservatore non lineare basata su estensione dinamica dell'uscita. Questo metodo minimizza l'impatto del rumore di misura e garantisce la proprietà di stabilità entrata-uscita (ISS) dell'errore di stima tramite una nuova condizione LMI.

Parole chiave : Serra Verticale Adattiva, Approccio a scatola nera, Modello di crescita delle colture, Modello basato su dati, Reti neurali feedforward, Ottimizzazione, UAV, Progettazione di osservatori non lineari, Inuguaglianze Matriciali Lineari (LMI), Stabilità entrata-uscita .

Résumé

La population mondiale devrait atteindre près de 10 milliards d'ici 2050, entraînant une hausse attendue de la demande alimentaire due à la croissance démographique, au développement économique et à l'urbanisation. Pour répondre à cette demande de manière durable, les systèmes de cultures en serre, notamment l'agriculture verticale, se sont imposés comme une solution prometteuse, offrant des rendements élevés par unité de surface cultivée. La *Serre Verticale Adaptative* (SVA) est une serre verticale industrielle innovante qui ajuste dynamiquement la distance entre ses étagères empilées, optimisant ainsi les conditions de croissance à mesure que les plantes progressent dans leurs stades de développement. Ce principe adaptatif permet de surmonter le conflit traditionnel entre le maintien de conditions optimales et la minimisation de la consommation d'énergie.

Cette thèse présente deux axes de recherche principaux contribuant au développement de la SVA. Le premier axe consiste à développer un modèle de croissance des cultures basé sur des données et une approche boîte noire pour les plantes cultivées dans les SVA. Étant donnée l'environnement dynamique de la cultivation qui s'adapte à la croissance des plantes tout au long du cycle de la production, un modèle précis de croissance des cultures est essentiel pour optimiser le mouvement des étagères et le contrôle du système. Tandis que les modèles de croissance dynamique traditionnels dépendent souvent de nombreux paramètres et sont spécifiques à certains types de cultures, nous proposons une approche boîte noire utilisant des réseaux de neurones á action directe pour prédire la hauteur des plantes à chaque étape. Ce modèle est adaptable à divers types de cultures, efficace en termes de calcul après l'entraînement et particulièrement adapté aux systèmes d'agriculture verticale innovants tels que la SVA. L'efficacité du modèle est démontrée à travers des ensembles de données synthétiques et réelles, mettant en avant son potentiel dans l'optimisation de la production agricole.

Le second axe de recherche se concentre sur l'automatisation de la SVA grâce à l'intégration de véhicules aériens sans pilote (UAV) dans le cadre de l'agriculture de précision. L'intégration des UAV permet de réaliser des applications telles que le suivi de la santé des cultures en complément des données recueillies par les capteurs stationnaires existants, la pollinisation automatique, la pulvérisation et l'irrigation, etc., optimisant ainsi les opérations agricoles à travers les étagères empilées et complétant les capteurs stationnaires existants. Pour soutenir cette automatisation, nous proposons différentes méthodes de synthése d'observateurs pour des systèmes non linéaires utilisant des inégalités matricielles linéaires (LMI) fournissant ainsi une estimation précise de l'état des UAV et garantissant la convergence exponentielle de l'observateur. Deux contributions majeures sont présentées : premièrement, une nouvelle condition LMI moins conservatrice appliquée pour résoudre le critère circulaire $\mathcal{H}_\infty$, et deuxièmement, une méthode de synthèse d'observateur non linéaire basée sur une extension dynamique de la sortie. Cette méthode minimise l'impact du bruit de mesure et garantit la propriété de stabilité entrée-état (ISS) de l'erreur d'estimation via une nouvelle condition LMI.

Mots clès : Serre Verticale Adaptative (SVA), Modèle de Croissance des Cultures, Modèle Boîte Noire, Réseaux de Neurones Feedforward , UAV, Optimisation, Inégalités Matricielles Linéaires (LMI), Synthèse d'Observteurs Non-Linéaires, Stabilité Entrée-État.

Borsa di dottorato cofinanziata con risorse del
Programma Operativo Nazionale Ricerca e Innovazione 2014-2020 (CCI 2014IT16M2OP005),
risorse Fondo Sociale Europeo REACT-EU, Azione I.1 “Dottorati Innovativi con caratterizzazione industriale”, Azione IV.4 “Dottorati e contratti di ricerca su tematiche dell’innovazione” e Azione IV.5 “Dottorati su tematiche Green”

UNIONE EUROPEA
Fondo Sociale Europeo

Ministero dell'Istruzione, dell'Università e della Ricerca

PON
RICERCA E INNOVAZIONE
2014 - 2020