

# Adaptive direct data-driven predictive LIDAR-assisted wind turbine control

N. van Wering

Master of Science Thesis



# **Adaptive direct data-driven predictive LIDAR-assisted wind turbine control**

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft  
University of Technology

N. van Wering

June 10, 2025

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of  
Technology



Copyright © Delft Center for Systems and Control (DCSC)  
All rights reserved.



DELFT UNIVERSITY OF TECHNOLOGY  
DEPARTMENT OF  
DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)

The undersigned hereby certify that they have read and recommend to the Faculty of  
Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis  
entitled

ADAPTIVE DIRECT DATA-DRIVEN PREDICTIVE LIDAR-ASSISTED WIND TURBINE  
CONTROL

by

N. VAN WERING

in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE SYSTEMS AND CONTROL

Dated: June 10, 2025

Supervisor(s):

\_\_\_\_\_  
prof.dr.ir. J.W. van Wingerden

\_\_\_\_\_  
ir. R.T.O. Dinkla

Reader(s):

\_\_\_\_\_  
dr.ir. A.K. Pamososuryo

\_\_\_\_\_  
dr.ir. M. Guo



---

# Abstract

To achieve net-zero greenhouse gas emissions by 2050, the Global Wind Energy Council (GWEC) emphasizes the need for a significant expansion in global wind power capacity. A key factor of this growth is the upscaling of wind turbines, which increases the swept area and exposes the blades to higher wind speeds at higher altitudes, thereby increasing energy yield. However, this upscaling introduces significant control challenges. Larger wind turbines increase aeroelastic complexity, with stronger nonlinear dynamics arising from increased blade flexibility and more significant wind shear across the expanded rotor diameter. Additionally, the increased rotor inertia delays the system response, making rotor speed regulation more difficult under varying wind conditions.

This thesis proposes an adaptive closed-loop Subspace Predictive Control (SPC) framework designed in an attempt to handle the complexity of larger, nonlinear wind turbines. Closed-loop SPC is a direct Data-Driven Predictive Control (DDPC) method that does not rely on explicit state-space modeling, but instead uses measured input-output data to predict future outputs and compute optimal control action. For the optimal control action, it sets up a receding horizon optimization that regulates above-rated rotor speed. This thesis focuses on the above-rated region, where aeroelastic complexity becomes more pronounced due to higher wind speeds and greater wind speed variations, posing significant control challenges.

To capture time-varying and nonlinear behavior more effectively, the closed-loop SPC incorporates Recursive Least Squares (RLS). The controller adapts to time-varying conditions through online parameter estimation using RLS, which updates a locally linear model in real time. Three RLS variants are examined: standard RLS without forgetting, RLS with exponential forgetting, and RLS with directional forgetting. Standard RLS weighs all past data equally, which may be effective when the system dynamics remain stationary but limits adaptability to changing conditions. Exponential forgetting addresses this by placing more weight on recent data, improving adaptiveness, but at the potential cost of losing parameter estimation accuracy in less-excited directions. Directional forgetting refines this further by applying forgetting selectively along the directions of incoming data, preserving excitation in recently unexcited directions and enhancing estimation robustness.

To reduce the phase lag introduced by increased rotor inertia, wind preview information is incorporated into the closed-loop SPC as a feedforward signal. This wind preview is included

in the receding horizon optimization problem, enabling the controller to anticipate and proactively respond to upcoming wind changes. Additionally, the wind preview is demonstrated using more realistic measurements obtained through a LIDAR simulator.

The adaptive closed-loop SPC is validated on the DTU 10 MW reference turbine using QBlade, a high-fidelity wind turbine simulator. Various wind scenarios, including gusts, ramps, and turbulent inflow, are evaluated with and without wind preview feedforward. Results demonstrate that the inclusion of wind preview significantly improves rotor speed tracking performance and reduces pitch activity. This improvement is also observed when more realistic LIDAR wind measurements are used in simulations with a turbulent wind field. In the conducted wind cases, among the RLS-based adaptive closed-loop SPC strategies, exponential forgetting combined with wind preview consistently outperformed the other RLS approaches across all scenarios evaluated in this thesis. These findings demonstrate that introducing adaptiveness through forgetting, together with feedforward wind information, can enhance closed-loop Subspace Predictive Control (SPC) performance in rated rotor speed tracking.

---

# Table of Contents

<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1-1 Wind turbine control and upscaling challenges . . . . .	2
1-1-1 Basic wind turbine control . . . . .	2
1-1-2 Increased wind turbine model complexity due to upscaling . . . . .	4
1-2 Literature review . . . . .	4
1-2-1 Control solutions to facilitate wind turbine upscaling . . . . .	5
1-2-2 DDPC for Nonlinear Systems: State-of-the-art . . . . .	8
1-3 Research question . . . . .	9
1-4 Outline . . . . .	11
<b>2 Adaptive closed-loop SPC with disturbance feedforward</b>	<b>13</b>
2-1 Setup and assumptions . . . . .	13
2-1-1 Model structure . . . . .	14
2-1-2 Notation . . . . .	14
2-1-3 Assumptions . . . . .	16
2-2 Disturbance feedforward SPC . . . . .	16
2-2-1 Obtaining the data equation . . . . .	17
2-2-2 Constructing an output predictor . . . . .	19
2-3 Recursive solution of the parameter estimation problem . . . . .	21
2-3-1 Recursive Least Squares (RLS) . . . . .	21
2-3-2 Exponential forgetting RLS . . . . .	23
2-3-3 Directional forgetting RLS . . . . .	24

<b>3</b>	<b>Simulation setup and results</b>	<b>27</b>
3-1	Simulation setup . . . . .	27
3-1-1	Wind turbine simulator and wind turbine model . . . . .	27
3-1-2	LIDAR simulator . . . . .	29
3-1-3	Simulation overview . . . . .	30
3-1-4	Open loop data collection . . . . .	31
3-2	Adaptive closed-loop SPC controller design . . . . .	33
3-2-1	Optimal control problem formulation . . . . .	33
3-2-2	Effect of parameter variations . . . . .	34
3-3	Wind cases . . . . .	37
3-3-1	Wind case I: Mexican hat wind gust . . . . .	37
3-3-2	Wind case II: Wind ramp . . . . .	38
3-3-3	Wind case III: Turbulent wind . . . . .	38
3-4	Simulation results . . . . .	39
3-4-1	Sensitivity analysis . . . . .	39
3-4-2	Wind case I without controller, excitation and wind preview . . . . .	43
3-4-3	Wind case II without wind preview . . . . .	45
3-4-4	Wind case II with wind preview . . . . .	47
3-4-5	Wind case II with and without wind preview . . . . .	49
3-4-6	Wind case III with LIDAR preview . . . . .	51
3-5	Concluding remarks on the simulations . . . . .	53
<b>4</b>	<b>Conclusion</b>	<b>55</b>
4-1	General conclusion . . . . .	55
4-2	Future work and recommendations . . . . .	58
	<b>Bibliography</b>	<b>61</b>
	<b>Glossary</b>	<b>67</b>
	List of Acronyms . . . . .	67

---

# Acknowledgements

I would like to thank my mother for her support and pride, which I deeply felt throughout this recent period, giving me the motivation I needed. I would also like to thank my sister Ruby, who I am very proud of and who is always there when needed. I am also grateful to my girlfriend Sirinya, who makes me a lucky and happy man.

Furthermore, I would like to thank Jan-Willem and Rogier for their guidance and valuable feedback.

Delft, University of Technology  
June 10, 2025

N. van Wering



---

# Chapter 1

---

## Introduction

The world's energy needs have grown significantly over the last few decades, with global energy consumption increasing by 54% from 1990 to 2020 [1]. This increase has been driven by urban expansion, industrial development, and the spread of digital technologies [2]. To this date, most of the consumed energy comes from fossil fuels like oil, coal, and natural gas. However, these sources are exhaustible and release greenhouse gases that contribute to global warming and consequently drive climate change. The combination of the increase in energy demand and the dominant role of fossil fuels in the supply of energy is an issue. This problem can be solved by using cleaner, renewable energy sources. Wind energy is one such renewable source. Although greenhouse gases are emitted during the manufacturing of wind turbines, their operation is free of such emissions. Furthermore, wind energy holds substantial promise due to its widespread geographic availability and the ongoing advancements in turbine efficiency and technology [3].

The Hornsea Two Wind Farm in the United Kingdom (UK) is a good example of the potential of wind energy. This wind farm has a capacity of 1.2 GW and can supply electricity for around 1.4 million households [4]. While large projects as these are important steps forward, wind energy needs to grow much faster worldwide to meet long-term climate goals. The Global Wind Energy Council (GWEC) calculated that to meet net-zero greenhouse gas emissions, a cumulative global installation of wind turbines of 3200 GW is necessary by 2050. Currently, 906 GW of wind turbines are in operation globally, therefore, to reach the goal for net-zero, another 2294 GW of wind turbines must be installed [5]. Based on the added wind capacity from previous years, the trend shows that at this rate, only 68% of the goal for net-zero will be achieved [5]. Research on the optimization of wind turbine design can help achieve this ambitious goal of 3200 GW capacity by 2050.

One method of capturing more wind speed is done by wind turbine upscaling, which involves increasing rotor size and hub height. A larger rotor diameter significantly increases the swept area, allowing the turbine to capture more energy from the wind. Additionally, higher hub heights enable turbines to access generally higher wind speeds available at greater altitudes, further enhancing energy output. A survey of experts reveals that wind turbine rotor sizes are expected to grow significantly, as illustrated by Figure 1-1 [6].

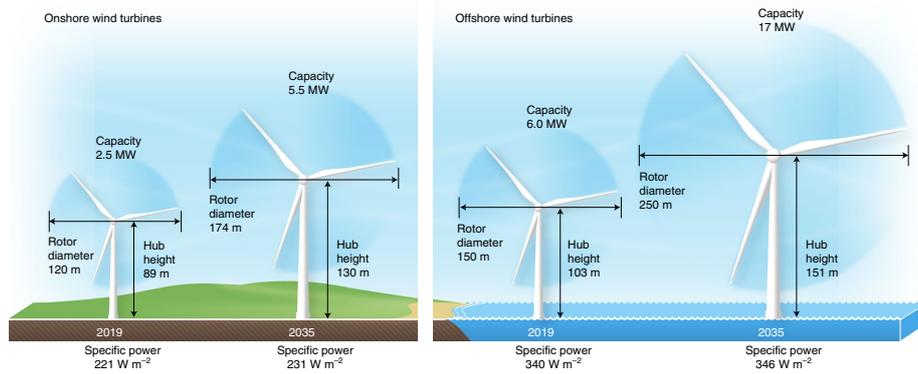


Figure 1-1: Expected growth of wind turbine size and power [6].

Furthermore, this higher power production with upscaled wind turbines results in a lower Levelized Cost of Energy (LCOE). The LCOE is a key metric in the energy sector used to measure the cost-effectiveness of electricity generation. It is typically expressed in US Dollars per MWh and accounts for the total costs of building and operating a power source, such as a wind turbine, over its expected lifetime. In [7], upscaled wind turbines and their LCOE are analyzed, finding that the use of 20 MW instead of 6 MW wind turbines for a 1 GW wind farm results in a 20% lower LCOE. Between 2014 and 2019, the increase in wind turbine power capacity already reduced the overall LCOE by more than 40% [7].

## 1-1 Wind turbine control and upscaling challenges

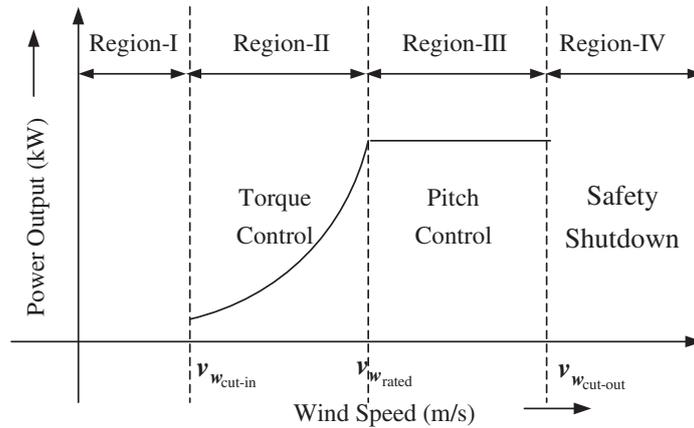
In this subsection, basic wind turbine controls are explored. Here, the basics of power curve tracking with pitch and torque control are presented. Furthermore, challenges in wind turbine control that arise from upscaling turbine size are discussed.

### 1-1-1 Basic wind turbine control

The general goal of wind turbine control is to maximize power production while minimizing the cost of wind energy, where the latter can be achieved via load minimization. Load minimization is critical in reducing turbine fatigue as it reduces the amplitude and frequency of mechanical stresses experienced by the structure. This decreases the structural degradation, particularly for the blade, tower, and foundation, thereby increasing the wind turbine's lifespan. Power maximization involves absorbing the maximum possible amount of wind energy by the wind turbine. However, this objective is not pursued once the rated wind speed is reached, as the wind turbine will have already achieved its maximum power capacity by then.

Wind turbines generate aerodynamic power through the conversion of kinetic energy in wind, described by the following equation:

$$P_w = C_p(\beta, \lambda) \frac{1}{2} \rho \pi R^2 v^3,$$



**Figure 1-2:** The operating regions of a variable wind speed wind turbine [9]

where  $P_w$  is the aerodynamic power,  $\rho$  is air density,  $R$  the rotor radius,  $v$  wind speed, and  $C_p$  the aerodynamic power coefficient, which depends on pitch angle  $\beta$  and tip speed ratio  $\lambda$ . The tip speed ratio  $\lambda$  is defined as:

$$\lambda = \frac{\omega R}{v}, \quad (1-1)$$

where  $\omega$  is the rotor angular velocity. The aerodynamic power captured by the rotor is converted to electric power. During partial load conditions, when the wind speed is below the level required to reach the maximum power output, the control objective is to maximize power capture. As wind speed increases beyond this point, the turbine enters the full load region, where the electric power output is regulated and maintained at the rated power. The rated power is defined as the maximum continuous power that the turbine is designed to deliver [8]. The variation of the aerodynamic power output as a function of wind speed is shown in Figure 1-2.

As shown in Figure 1-2, wind turbine operation is divided into four regions. In region I, below the cut-in wind speed, the turbine remains idle without generating power. Region II, between the cut-in and rated wind speeds, focuses on optimizing aerodynamic efficiency. Here, generator torque control continuously adjusts the rotor speed to maintain an optimal, constant tip speed ratio  $\lambda$ , while the pitch angle remains fixed at the position that maximizes the power coefficient. This strategy ensures maximum aerodynamic power extraction as the wind speed varies.

Region III spans from the rated wind speed to the cut-out wind speed. In this region, controlling rotor speed above rated conditions is vital to prevent mechanical and electrical overloads due to excessive aerodynamic forces. Because these forces grow significantly as wind speed rises, generator torque alone becomes insufficient to keep the rotor speed and output power within safe limits. Typically, the rated generator torque is maintained constant in Region III. Thus, pitch control is employed as the main actuator, adjusting the blade pitch angle  $\beta$  to regulate power to its rated value by controlling the aerodynamic torque on the blades. This allows the wind turbine to maintain its rated power output despite increasing wind speeds.

In Region IV, wind speeds reach or exceed the cut-out limit. For safety purposes, the turbine

undergoes shutdown procedures and ceases power production to prevent structural damage [10].

### 1-1-2 Increased wind turbine model complexity due to upscaling

Larger wind turbines also create more challenges. The complexity of modeling the dynamics can grow because wind speeds can differ significantly across the length of the blades, primarily due to wind shear and the varying altitudes the blades sweep through. As the rotor diameters increase, the vertical distance between the blade tips and the hub height grows, causing the upper portion of the rotor to see substantially higher wind speeds than the lower part. Because aerodynamic forces scale with the square of wind speed, even small changes in wind speed can induce significant different loadings along the blades, amplifying the rotor's nonlinear aerodynamic response. The different aerodynamic forces at different sections of the rotor require more complex dynamic models to accurately predict the turbine's rotor dynamics. This increases the nonlinearity of rotor speed dynamics, making it more challenging to determine the optimal pitch control input [11]. The complexity becomes more pronounced at higher wind speeds, particularly in the rated load region. This is because the differences in aerodynamic loading along the blade length increase significantly due to the squared relationship between wind speed and aerodynamic loading.

The larger wind turbine blades also contribute to a larger rotor inertia. This increased inertia results in slower dynamic responses of the rotor speed to changes in both wind conditions and pitch angle. As a result, the wind turbine system experiences a delay in adapting to the operational state to maximize power generation [12]. This requires advanced control methods that can compensate the phase loss introduced by an increased rotor inertia, ensuring faster adaptation to wind fluctuations. Feedforward control, discussed in the next subsection, plays a crucial role in addressing this challenge.

Moreover, upscaling leads to slimmer, longer rotors due to adverse mass scaling. As the rotor diameter increases, the mass grows faster than the structural strength if proportions are maintained, making thick blades impractically heavy. To limit weight, the chord length gets reduced, resulting in slimmer, more flexible blades. This increased flexibility amplifies the nonlinearity of aeroelastic dynamics [13]. Traditional structural models often represent the blade as a one-dimensional beam and apply linear modal analysis. While this approach simplifies computations, it fails to accurately capture the large deflections observed in modern, large-scale blades [14].

More complex aeroelastic models are thus needed to represent the nonlinear dynamics in large wind turbines, alongside control methods that account for these nonlinear dynamics.

## 1-2 Literature review

The upscaling of wind turbines introduces increased mechanical loads, nonlinear dynamics, and stronger spatial wind variations. To address these challenges, this section reviews advanced control strategies, including Individual Pitch Control (IPC), Model Predictive Control (MPC), direct Data-Driven Predictive Control (DDPC), and the use of Light Detection and Ranging (LIDAR)-based feedforward control. These methods aim to improve control and

adaptability to complex wind turbine behavior. Finally, state-of-the-art DDPC frameworks for nonlinear system applications are discussed, with a focus on their suitability for addressing the control challenges posed by the increased complexity in dynamics of upscaled wind turbines.

### 1-2-1 Control solutions to facilitate wind turbine upscaling

The upscaling of wind turbines introduces complex dynamics, requiring advanced control methods. Several solutions are outlined in this section, including IPC for blade-specific pitch control. This control method with a Multiple Input Multiple Output (MIMO) nature can be accommodated with the discussed MPC. Furthermore, MPC can also utilize previewed information due to employing a receding horizon method. This leads to the examination of LIDAR wind speed information, which can be used in a feedforward control setting with possible implementation in MPC. Finally, direct DDPC gets explored, addressing the challenges in modeling complex wind turbine dynamics.

#### Individual Pitch Control

Active pitch control of wind turbine blades serve the goal of rotor speed regulation and helping to reduce aerodynamic loads. This blade pitching can be done such that all of the blades have the same pitch, which is called Collective Pitch Control (CPC), or individually called IPC.

The method of IPC adjusts the pitch of the blades individually, such that asymmetric rotor loading, induced by factors like wind shear, yaw misalignment, and turbulence can be brought to a minimum [15]. Counteracting the asymmetric loads leads to more stable rotor speeds for better rotor speed regulation and reduced mechanical fatigue. A common technique used in IPC to handle these individual blade load effects is the Multi-Blade Coordinate (MBC) transformation. In this transformation, the loads measured on each rotating blade are converted from a rotating frame into a non-rotating reference frame. By doing so, it becomes much clearer which parts of the load signals correspond to periodic or asymmetric effects such as wind shear or yaw misalignment. The controller then uses this information to generate the individual pitch signals for each blade, thereby counteracting the uneven loads [15].

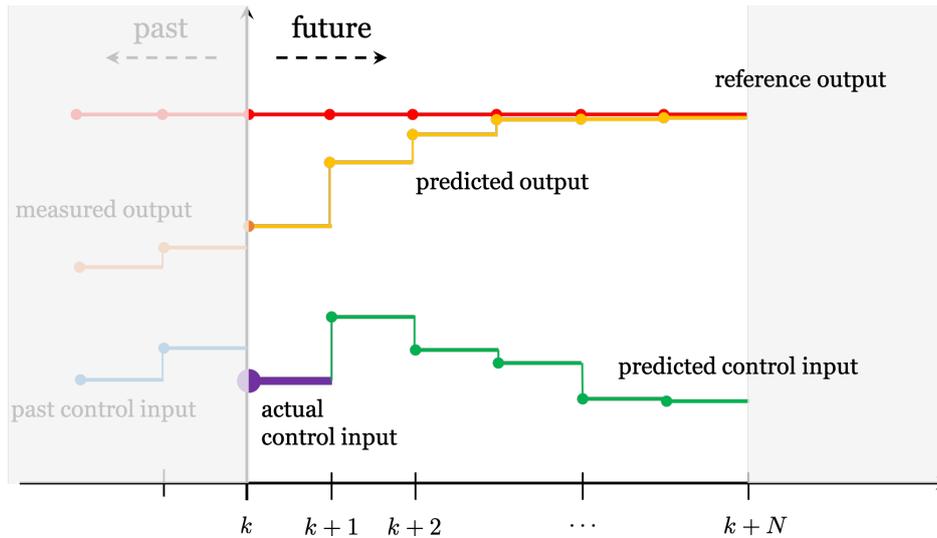
This is especially beneficial for large-scale wind turbines, where greater rotor diameters lead to greater wind speed differences across blades and therefore also different loads [16]. Since IPC involves processing multiple inputs, like measured loads, to determine outputs of three pitch actuators, it has a MIMO control nature. This capability makes it well-suited for handling the increasing computational and dynamic complexities associated with larger wind turbines.

#### Model Predictive Control

A control method that accommodates MIMO control is MPC. MPC was developed in 1979 [17] and has since been applied in a variety of industries, such as the oil and chemical industries [18].

A short visualization of the MPC algorithm is visible in Figure 1-3 with time-step  $k$  and control horizon  $N$ . With MPC, a plant model is utilized to predict optimal inputs, states,

and outputs over a specified future interval, known as the prediction horizon. A cost function also evaluates feasible input and output trajectories while considering constraints. An optimization algorithm finds an input trajectory that minimizes the cost function. This process is complemented by a receding horizon strategy, where the first control input of the computed input trajectory is applied. This is visible in Figure 1-3 where the predicted control input trajectory over horizon  $N$  is in green, and the actual used first control input is in purple. After this, the horizon advances one timestep, and the process is repeated [19].



**Figure 1-3:** Visualization of the receding horizon MPC with time-step  $k$  and control horizon  $N$  [20].

Looking at the previously discussed MPC description, there are more important features next to accommodating MIMO control. One of them is the property of being able to handle constraints. This could, for example, limit the pitch activity to stay within the physical limits of the pitch actuator's angle and pitch rate. Such a limit on the pitch activity can also reduce the fatigue load on the pitch actuator and prevent failure [21].

Furthermore, a key feature of MPC is the receding horizon strategy, which allows the incorporation of preview information such as future wind speed forecasts. This enables the controller to anticipate upcoming changes in wind speed and compensate for the phase delays introduced by rotor inertia. These delays become increasingly significant as wind turbines increase in size, as discussed before. In addition, due to its inherent MIMO structure, MPC can generate distinct pitch signals for each individual blade. This makes it possible to integrate IPC within the MPC framework, enabling independent pitch control per blade. As a result, the controller can reduce the negative effects of asymmetric loading, which are more pronounced in larger wind turbines, as discussed in the IPC subsection.

### Direct Data-Driven Predictive Control

Upscaling wind turbines accentuates nonlinear behaviors, complicating the creation of accurate, physics-based state-space models [22]. Developing such system models is costly and

complex, and assumptions or unmodeled dynamics can lead to mismatches, affecting control robustness and safety. This has driven interest in Data-Driven Control (DDC) control strategies, which uses data to identify a model [23].

In DDC, sensor data can be used to identify state-space models that serve as a basis for controller design. DDC is able to address issues like unmodeled dynamics and nonlinearity when the data accurately reflects these behaviors in the identified state-space model [23]. While DDC makes use of a state-space model, it can also be used in a MPC framework. This is called DDPC. With wind turbine data, such as pitch signals, rotor speed and wind speed data, being available, DDPC has been already researched with applications on wind turbines [24].

Direct DDPC bypasses the need for an explicit state-space model by using input-output data to derive future control actions [25]. Within direct DDPC, methods like Subspace Predictive Control (SPC) [26] and Data-enabled Predictive Control (DeePC) [27], similar to MPC, employ a receding horizon optimal control framework [28]. To address the bias introduced by noise in traditional SPC, closed-loop SPC has been proposed [29]. In addition, this variant requires fewer parameters, showing to be beneficial with the use of an adaptive filter in the next subsection. These direct DDPC methods can utilize constraints and receding horizons specific to wind turbine control, constraining pitch activity and including future wind speed information. This wind speed information can be provided by using LIDAR measurements as will also be discussed in the next subsection.

Despite direct DDPC being labeled "direct", it is important to note that these methods do not necessarily imply a direct mapping from data to optimal control input without any structural model. For instance, SPC uses an intermediate output predictor constructed from past input-output data, to implicitly describe the system's dynamics and compute control inputs. While this approach bypasses the need for explicit state-space identification, it still builds an output predictor model of the system behavior through data. This is an ongoing discussion about whether such methods truly qualify as "direct" DDPC [23].

## LIDAR Measurements

Transitioning to advancements in measurement technology, LIDAR has gained interest in the wind turbine sector for measuring wind speeds. LIDAR is a remote sensing technology that uses laser beams to measure distances and track air movement. It is commonly used for wind speed measurement, assessing wind potential at a site, and monitoring airflow around turbines. Unlike traditional anemometers on tall masts, LIDAR provides a flexible, contact-free method for accurate wind measurements at different heights and distances [30].

The fundamental principle of LIDAR is based on the emission of a laser beam, which interacts with airborne particles. A fraction of the light is scattered back toward the receiver, and the change in frequency due to the Doppler effect is used to infer the velocity of the particles in the beam's direction. This allows wind speed to be determined at various points in the atmosphere [31].

LIDAR systems are divided into two main types based on how they detect wind: coherent detection, which measures Doppler shifts by mixing the received signal with a stable reference laser, and direct detection, which uses spectral filtering techniques to analyze the frequency

shift of scattered light. Coherent detection, especially in the near-infrared range, is preferred for wind energy applications because it is highly sensitive and works without needing additional particles in the air. By contrast, direct detection relies more on atmospheric particles, such as aerosols or dust, to generate a sufficiently strong return signal, making it less effective in clean-air conditions [32].

LIDAR systems can also be distinguished by their emission waveform: Pulsed-Wave (PW) or Continuous-Wave (CW). PW LIDAR emits short bursts of laser energy and determines range by measuring the time of flight of the pulse. CW LIDAR, by contrast, emits a continuous beam and determines range by focusing the laser at specific distances [33]. How this LIDAR wind speed information can be used for feedforward control is discussed in the next subsection.

### **Feedforward control**

Recent research in wind turbine control highlight the potential of incorporating preview-based feedforward strategies within predictive control frameworks, such as MPC [34]. When using an anemometer, the MPC can use this measured wind speed as a constant wind speed over the prediction horizon due to unknown future wind trajectories. However, incorporating the discussed LIDAR measurements as current and future disturbance trajectories allows the controller to anticipate and adjust to upcoming wind changes, effectively creating a feedback-feedforward controller [35]. As demonstrated in [36], incorporating future wind speed information from LIDAR into the MPC framework can significantly reduce blade loads while simultaneously requiring less pitch actuation of the wind turbine.

Building on these developments within the MPC framework, similar concepts can be extended to the previously discussed direct DDPC, such as closed-loop SPC. The receding horizon principle in direct DDPC, like in MPC, also enables the integration of LIDAR-based feedforward information. An example of such an implementation is given in [37], where LIDAR disturbance preview knowledge is incorporated into the closed-loop SPC algorithm.

## **1-2-2 DDPC for Nonlinear Systems: State-of-the-art**

This section provides an overview of state-of-the-art frameworks suitable for DDPC applied to nonlinear systems, with a focus on methods capable of accurately capturing high-fidelity nonlinear wind turbine dynamics. While these methods are broadly applicable to nonlinear systems, their ability to represent complex dynamic behavior makes them particularly relevant for addressing the increased control complexity in wind turbines resulting from upscaling.

Among these methods, the Wiener-Hammerstein method has emerged as a promising technique. It transforms nonlinear input-output data into a linear framework by applying static nonlinear mappings represented by basis functions [38]. This method has shown successful implementation to model a wind turbine's torque and trust coefficient with suitable basis functions [39]. However, its effectiveness is constrained by being dependent on predefined basis functions, which may limit to capturing dynamics that extend beyond the sampled data. A similar idea of using basis functions underpins the Reproducing Kernel Hilbert Space (RKHS) approach. In contrast to Wiener-Hammerstein, RKHS maps the data into an infinite-dimensional feature space using kernel functions, which allows it to capture complex

nonlinear relationships. The so-called kernel trick enables this mapping without explicitly computing the high-dimensional features, keeping computations tractable [40]. The challenge, however, lies in selecting an appropriate kernel, as different choices can significantly influence system trajectories [41].

Another promising framework is the Linear Parameter-Varying (LPV) approach, which represents nonlinear dynamics as linear time-varying systems that are dependent on measurable scheduling variables [42]. This formulation enables the application of linear design techniques while accurately capturing nonlinear behavior. LPV models have been widely applied across various nonlinear control problems and have shown particular success in wind turbine control [43]. A downside of LPV models is that identifying the output predictor using closed-loop data introduces the curse of dimensionality, leading to computational challenges [44].

Lastly, Recursive Least Squares (RLS) filtering techniques handle nonlinear dynamics by approximating them as linear time-varying systems. This approach can be interpreted as a form of continuous local linearization, where the linear approximation adapts in real time to capture the behavior of the underlying nonlinear system. RLS is an adaptive filter, estimating system dynamics, and with the addition of an exponential forgetting factor, it can prioritize new data in its model estimation [45]. However, standard RLS with exponential forgetting can cause the covariance matrix to accumulate large eigenvalues in directions that are insufficiently excited by new data. This inflates the uncertainty in those directions, making parameter updates overly sensitive to incoming data points of recently less excited directions. This issue is addressed by directional forgetting, which maintains sufficient excitation in all directions by selectively discounting old data only when new information is available [46].

Applying exponential or directional forgetting RLS to closed-loop SPC for wind turbine control enables real-time adjustment of model parameters, to address nonlinear dynamics and wind variability. This approach demonstrated to show promising results applied to a small-scale wind turbine in [47]. Here, above-rated rotational speed regulation was maintained using adaptive closed-loop SPC with RLS and forgetting, demonstrating improved rotor speed tracking performance.

Integrating RLS with forgetting methods in closed-loop SPC offers an intuitive approach by dynamically updating the linear approximation of the system as conditions evolve, thereby improving the local representation of nonlinear behavior. This makes it particularly well-suited for systems with mild nonlinearities or slowly varying operating conditions, where the local data remains representative of the system dynamics. Furthermore, its computational efficiency makes it attractive for online implementation, as RLS eliminates the need to invert large data matrices at every time step. Given these advantages, closed-loop SPC techniques combined with RLS filtering will be further explored, aiming to enhance the predictive control of the discussed nonlinear wind turbine dynamics.

### 1-3 Research question

The preceding sections have underscored the importance of wind energy in meeting global energy demands and combating climate change. As wind turbines scale up in size to reduce the LCOE and increase capacity, new challenges arise in control systems due to amplified nonlinear dynamics. Despite advancements in control strategies like IPC and MPC, two key

aspects remain insufficiently addressed: accommodating the complex nonlinear behavior of larger turbines and at the same time integrating wind preview information to counteract phase delays.

A summary of the proposed controller features, along with the corresponding issues they aim to solve, is as follows:

- *Selected RLS-based adaptive filtering within SPC: Addresses the nonlinear rotor dynamics without relying on specific nonlinear models.*
- *Incorporation of LIDAR-derived wind preview information: Uses feedforward control to mitigate phase delays in rotor speed regulation that arise due to increased rotor inertia.*

It is hypothesized that integrating these features will lead to a control strategy capable of effectively managing the complex nonlinear dynamics of large-scale wind turbines and improving performance by proactively utilizing wind preview information. Thus, the central research question explores how to systematically combine these elements into a cohesive and effective control framework for large-scale wind turbines.

This leads to the formulation of the research question:

***How can closed-loop SPC incorporate both an adaptive filter and LIDAR-based wind preview information to achieve optimal above-rated rotor speed regulation of a wind turbine?***

To answer this question, the following sub-questions will be considered:

- *How can the adaptive closed-loop SPC framework incorporate previewed future wind speed information from LIDAR measurements?*  
Increased rotor inertia leads to slower responses to wind fluctuations, making it challenging to maintain rated rotor speed regulation. Incorporating LIDAR-based wind preview information enables anticipatory control actions, which can compensate for the delay introduced by the system's slow dynamics. This sub-question aims to develop a systematic method for integrating realistic LIDAR preview wind data into the adaptive closed-loop SPC framework, with the goal of enhancing rated rotor speed tracking performance under varying wind conditions.
- *How can standard RLS, exponential forgetting, and directional forgetting be used for feedforward closed-loop SPC to adapt to nonlinear rotor dynamics?*  
This subquestion investigates how different RLS-based adaptation strategies enable closed-loop SPC to approach the nonlinear and time-varying behavior of wind turbine dynamics. The focus lies on three specific methods: standard RLS without forgetting, exponential forgetting, and directional forgetting. Each strategy reflects a different trade-off between memory of past data in its parameter estimation and adaptability to time-varying system dynamics. Investigating how these different RLS filtering strategies affect the adaptive updating of the closed-loop SPC local linearizations may provide insight into effectively modeling the nonlinear behavior of wind turbines and maintaining rated rotor speed regulation under varying wind conditions.

- *How do the developed adaptive closed-loop SPC with wind preview algorithms perform in above-rated rotor speed regulation when applied to a nonlinear wind turbine simulation?*  
The adaptive closed-loop SPC algorithms are evaluated in nonlinear wind turbine simulations using QBlade [48]. The study focuses on the above-rated regime, where higher wind speeds lead to pronounced nonlinear rotor dynamics and increased control challenges. The simulations are used to investigate the impact of different RLS forgetting strategies within a closed-loop SPC framework, as well as the benefit of wind speed feedforward preview on maintaining stable rotor speed regulation.

## 1-4 Outline

This thesis is structured into four chapters. The first chapter emphasizes the critical role of wind energy in achieving sustainable energy goals. It elaborates on the specific challenges arising from the upscaling of wind turbines, and explains how larger scales amplify nonlinear dynamics. Furthermore, this chapter introduces the necessity and potential advantages of using DDPC control methods, setting the foundation for the subsequent research question.

The second chapter presents the formulation of an adaptive closed-loop SPC framework incorporating disturbance feedforward control. It details the systematic development of a data-driven output predictor built entirely from historical input-output measurements, describes the integration of RLS estimation to adaptively update system parameters, and integrates exponential and directional forgetting to manage changing operating conditions. This chapter addresses this research's first two sub-questions.

The third chapter applies this adaptive closed-loop SPC framework to a nonlinear wind turbine simulation called Qblade, with the goal of rated rotor speed regulation. It first outlines the simulation setup, including the Qblade and LIDAR simulators. Moreover, it explores the controller's performance across varying wind scenarios, such as gusts, ramps, and turbulent wind flows. The effectiveness of incorporating wind preview and different RLS forgetting methods into the adaptive closed-loop SPC is assessed through a comprehensive sensitivity analysis and performance comparison. This chapter addresses this research's third sub-question .

The final chapter summarizes the results of this research, highlighting the key advantages and disadvantages of the different proposed adaptive closed-loop SPC algorithms in effectively addressing nonlinearities and enhancing rated rotor speed tracking. It concludes with recommendations for future research directions.



# Adaptive closed-loop SPC with disturbance feedforward

This chapter presents an adaptive closed-loop Subspace Predictive Control (SPC) framework with disturbance feedforward, which is later applied in this thesis to a nonlinear wind turbine simulation. As a direct Data-Driven Predictive Control (DDPC) method, closed-loop SPC avoids explicit state-space modeling and requires fewer parameters than state-space-based predictive control. An adaptive Recursive Least Squares (RLS) filter continuously updates the linear system approximation. Wind disturbances, such as Light Detection and Ranging (LIDAR) measurements, are incorporated through a feedforward structure by integrating preview information into the receding horizon control framework. By jointly adapting the system dynamics and incorporating wind preview information, the approach targets improved rotor speed regulation in large-scale wind turbines.

To structure the development of this framework, Section 2-1 outlines the problem setup, including notation and key assumptions. Section 2-2 derives the output predictor of closed-loop SPC with disturbance information. Section 2-3 presents an adaptive version of closed-loop SPC where the output predictor is estimated via RLS, along with a square root estimation method incorporating exponential and directional forgetting. The resulting RLS closed-loop SPC with forgetting, which is subsequently discussed, builds upon the work in [37, 47].

## 2-1 Setup and assumptions

This section describes the fundamentals of the closed-loop SPC algorithm applied on an unknown system where input, output, and disturbance signals are accessible. This system features a single output, as well as a control and disturbance input channel.

### 2-1-1 Model structure

For the introduction of the SPC frameworks, an unknown discrete-time Linear Time-Invariant (LTI) system is assumed. The system is assumed to be in a minimal state-space representation and is observable, with Gaussian white noise having a zero mean [26]. The input-disturbance-output data sets are assumed to be generated by a discrete-time LTI system, represented in state-space form with an innovation model:

$$\begin{aligned} x_{k+1} &= Ax_k + B_u u_k + B_w w_k + K e_k, \\ y_k &= Cx_k + D_u u_k + D_w w_k + e_k, \end{aligned} \quad (2-1)$$

with  $x_k \in \mathbb{R}^n$ ,  $u_k \in \mathbb{R}^m$ ,  $w_k \in \mathbb{R}^q$ ,  $e_k \in \mathbb{R}^l$ , and  $y_k \in \mathbb{R}^l$ , representing the state vector, control input, external disturbances, innovation noise, and the measured output respectively. The index  $k \in \mathbb{Z}$  refers to discrete-time steps. The innovation noise  $e_k$  is assumed to be an ergodic, zero-mean white noise process with a covariance defined as  $\mathbb{E}[e_k e_k^\top] = W \delta_{kj}$ , where  $W$  is positive definite ( $W > 0$ ). The Kronecker delta  $\delta_{kj}$  equals one if  $k = j$  and zero otherwise. The system dynamics are represented using the matrices  $A \in \mathbb{R}^{n \times n}$  and  $B \in \mathbb{R}^{n \times m}$ . The system matrices  $A \in \mathbb{R}^{n \times n}$ ,  $B_u \in \mathbb{R}^{n \times m}$ ,  $B_w \in \mathbb{R}^{n \times q}$ ,  $K \in \mathbb{R}^{n \times l}$ ,  $C \in \mathbb{R}^{l \times n}$ ,  $D_u \in \mathbb{R}^{l \times m}$ , and  $D_w \in \mathbb{R}^{l \times q}$  are also referred to as the state, input, disturbance, Kalman gain, output, input-output direct feedthrough, and disturbance-output direct feedthrough matrices, respectively. Furthermore, it is assumed that the system with  $n$  states is in its minimal representation and controllable. The disturbances  $w_k$  are treated separately from the inputs  $u_k$  because they cannot be controlled. In the representation given by Equation 2-1, the term  $e_k$  is eliminated from the first equation, leading to a system representation in the predictor form:

$$\begin{aligned} x_{k+1} &= \tilde{A}x_k + \tilde{B}_u u_k + \tilde{B}_w w_k + K y_k \\ y_k &= Cx_k + D_u u_k + D_w w_k + e_k \end{aligned} \quad (2-2)$$

where the matrices  $\tilde{A} = A - KC$  and  $\begin{bmatrix} \tilde{B}_u & \tilde{B}_w \end{bmatrix} = \begin{bmatrix} B_u & B_w \end{bmatrix} - K \begin{bmatrix} D_u & D_w \end{bmatrix}$  are introduced for shorter notation. The  $(\tilde{\cdot})$  notation is used to indicate parameters associated with the predictor model in Equation 2-2.

### 2-1-2 Notation

Before deriving the equations for the SPC algorithm, it is important to define some notation first. The collected data is arranged in block-Hankel matrices. A block-Hankel matrix has the following form:

$$U_{i,s,N} = \begin{bmatrix} u_i & u_{i+1} & \cdots & u_{i+N-1} \\ u_{i+1} & u_{i+2} & \cdots & u_{i+N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{i+s-1} & u_{i+s} & \cdots & u_{i+N+s-2} \end{bmatrix} \in \mathbb{R}^{ms \times N}$$

where the first index  $i$  refers to the time index of its top left entry, the second index  $s$  refers to the number of block rows in the block-Hankel matrices, the third index  $N$  refers to the

number of columns. To construct the block-Hankel matrix,  $N + s - 1$  data samples are required. Similarly, the block matrices  $Y_{i,s,N} \in \mathbb{R}^{\ell s \times N}$ ,  $E_{i,s,N} \in \mathbb{R}^{\ell s \times N}$ , and  $W_{i,s,N} \in \mathbb{R}^{q s \times N}$  are defined to represent output, noise, and disturbance data, respectively. The second index  $s$  is omitted when the block-Hankel matrices have only a single block-row. This is applied to state sequences, resulting in  $X_{i,N} \in \mathbb{R}^{n \times N}$ , as defined by:

$$X_{i,N} = \begin{bmatrix} x_i & x_{i+1} & \cdots & x_{i+N-1} \end{bmatrix}.$$

Similarly, a data vector is represented as:

$$u_{i,s} = \begin{bmatrix} u_i^\top & u_{i+1}^\top & \cdots & u_{i+s-1}^\top \end{bmatrix}^\top.$$

Stacked data vectors containing outputs, inputs, and disturbances are defined as follows:

$$z_{i,s} = \begin{bmatrix} y_{i,s}^\top & u_{i,s}^\top & w_{i,s}^\top \end{bmatrix}^\top.$$

Additionally, block-Toeplitz matrices are defined as:

$$\mathcal{T}_s(\mathcal{A}, \mathcal{B}, C, \mathcal{D}) = \begin{bmatrix} \mathcal{D} & 0 & 0 & \cdots & 0 \\ C\mathcal{B} & \mathcal{D} & 0 & \cdots & 0 \\ C\mathcal{A}\mathcal{B} & C\mathcal{B} & \mathcal{D} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ C\mathcal{A}^{s-2}\mathcal{B} & C\mathcal{A}^{s-3}\mathcal{B} & \cdots & C\mathcal{B} & \mathcal{D} \end{bmatrix},$$

where  $\mathcal{A}$ ,  $\mathcal{B}$ ,  $\mathcal{D}$  and  $0$  are matrices of suitable dimensions, here  $0$  is the zero matrix, and the number of block rows is again denoted by  $s$ . Using this definition, and given a positive integer future window length  $f \in \mathbb{Z}_{>0}$  the following specific block-Toeplitz matrices are defined:

$$\begin{aligned} \mathcal{T}_f^u &= \mathcal{T}_f(A, B_u, C, D_u) \in \mathbb{R}^{\ell f \times m f}, \\ \mathcal{T}_f^w &= \mathcal{T}_f(A, B_w, C, D_w) \in \mathbb{R}^{\ell f \times q f}, \\ \mathcal{H}_f &= \mathcal{T}_f(A, K, C, I) \in \mathbb{R}^{\ell f \times \ell f}, \end{aligned}$$

where  $I$  is an identity matrix of appropriate dimensions. Moreover, an extended observability matrix is given by:

$$\Gamma_f = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{f-1} \end{bmatrix} \in \mathbb{R}^{\ell f \times n}.$$

Finally, the extended controllability matrix is defined as:

$$\begin{aligned} \mathcal{K}_p(\mathcal{B}) &= \begin{bmatrix} A^{p-1}\mathcal{B} & A^{p-2}\mathcal{B} & \cdots & \mathcal{B} \end{bmatrix} \in \mathbb{R}^{n \times p}, \\ \mathcal{K}_p^y &= \mathcal{K}_p(K) \in \mathbb{R}^{n \times \ell p} \quad \mathcal{K}_p^u = \mathcal{K}_p(B_u) \in \mathbb{R}^{n \times mp} \quad \mathcal{K}_p^w = \mathcal{K}_p(B_w) \in \mathbb{R}^{n \times qp}, \end{aligned}$$

with positive integer past window length  $p \in \mathbb{Z}_{>0}$  and  $\mathcal{K}_p^z = \begin{bmatrix} \mathcal{K}_p^y & \mathcal{K}_p^u & \mathcal{K}_p^w \end{bmatrix} \in \mathbb{R}^{n \times (\ell+m+q)p}$ .

Matrices  $\tilde{\Gamma}_f$ ,  $\tilde{\mathcal{T}}_f$  and  $\tilde{\mathcal{K}}_p$  are defined in the same way as their counterparts without the tilde, with  $A$ ,  $B_u$ , and  $B_w$  replaced by  $\tilde{A}$ ,  $\tilde{B}_u$ , and  $\tilde{B}_w$ , respectively. With exception of the block-Toeplitz matrix  $\tilde{\mathcal{H}}_f$ , which is defined as  $\tilde{\mathcal{H}}_f = \mathcal{T}_f(\tilde{A}, K, -C, I)$ .

### 2-1-3 Assumptions

The following assumptions are considered throughout this thesis:

- *Assumption 1.* A stationary Kalman gain  $K$  from Equation 2-2 exists, ensuring that the eigenvalues of  $\tilde{A}$  lie strictly within the unit circle [49].
- *Assumption 2.* The past window length  $p$  is selected to be sufficiently large, so that based on Assumption 1, the powers of  $\tilde{A}$  decay to zero exponentially. This allows the approximation  $\tilde{A}^p \approx 0$ , which is commonly adopted to simplify the analysis by neglecting the effect of the initial state [50].
- *Assumption 3.* The disturbance and input signal is considered quasi-stationary, ensuring the convergence of time averages used in computations involving the input sequence [51].
- *Assumption 4.* The input sequence  $u = \text{col}(u_1, \dots, u_k)$ , the disturbance sequence  $w = \text{col}(w_1, \dots, w_k)$ , and the output sequence  $y = \text{col}(y_1, \dots, y_k)$  are considered, where  $k$  denotes the total number of data points.

The inputs are assumed to be persistently exciting of order  $mL + n$  and disturbance sequence of  $qL + n$  with  $k \geq L$ . That is, the following Hankel matrices

$$\begin{bmatrix} U_{1,L,k-L} \\ Y_{1,L,k-L+1} \end{bmatrix}, \quad \begin{bmatrix} W_{1,L,k-L} \\ Y_{1,L,k-L+1} \end{bmatrix}$$

are of rank  $mL + n$  and  $qL + n$ , respectively. Furthermore, it is assumed that  $L \geq d$ , where  $d$  denotes the lag of the observability matrix. It then follows that the combined input and disturbance sequences provide sufficient excitation. Consequently, the output sequence becomes uniquely determined by the system dynamics and is representative of its behavior [27].

## 2-2 Disturbance feedforward SPC

This section describes the construction of an output predictor using data-organized Hankel matrices, incorporating feedforward disturbance information. Additionally, the closed-loop SPC variant is introduced, which is preferred over traditional SPC as it requires fewer parameters to estimate, making it especially suitable for the RLS parameter estimation method described in Section 2-3.

### 2-2-1 Obtaining the data equation

To characterize the system's behavior without relying on the identification of a particular state-space model, historical input-disturbance-output data is used to construct an output predictor that forecasts future outputs over a specified time horizon. This subsection introduces the fundamental data equations essential for deriving the output predictor used in SPC. By utilizing the previously introduced notation, the system outputs are described as a function of an initial state combined with output, input, disturbance, and innovation trajectories. The formulation begins at time step  $k = i$ , as illustrated in Figure 2-1, and considers a past window of length  $p$  and a prediction window of length  $f$ . The system in Equation 2-1 can be reformulated in a so-called lifted representation, a structure frequently employed in subspace identification methods [52]. In this context, the lifted form is obtained by iterating the state and output equations in Equation 2-1 over  $p$  steps to get the input and output at step  $k + p$ :

$$\begin{aligned} x_{k+p} &= A^p x_k + \sum_{j=0}^{p-1} A^{p-1-j} (B_u u_{k+j} + B_w w_{k+j} + K e_{k+j}) \\ y_{k+p} &= C A^p x_k + \sum_{j=0}^{p-1} C A^{p-1-j} (B_u u_{k+j} + B_w w_{k+j} + K e_{k+j}) \\ &\quad + D_u u_{k+p} + D_w w_{k+p} + e_{k+p}. \end{aligned} \quad (2-3)$$

By applying the previously introduced block-Toeplitz and extended controllability matrices, along with the indexing convention shown in Figure 2-1, the system and output evolution of Equation 2-3 can be expressed in a compact matrix form. In this formulation, the state  $x$  is propagated over  $p$  steps to obtain the future state  $x_{k+p}$ , referred to as  $x_{i_p}$ . Output  $y$  is propagated over a future window of length  $f$ , resulting in a stacked vector of outputs from step  $k + p$  to  $k + p + f - 1$ , denoted as  $y_{i_p, f}$ . This yields the following expressions:

$$x_{i_p} = A^p x_i + \mathcal{K}_p^u u_{i,p} + \mathcal{K}_p^w w_{i,p} + \mathcal{K}_p^y e_{i,p} \quad (2-4a)$$

$$y_{i_p, f} = \Gamma_f x_{i_p} + \mathcal{T}_f^u u_{i_p, f} + \mathcal{T}_f^w w_{i_p, f} + \mathcal{H}_f e_{i_p, f} \quad (2-4b)$$

Furthermore, the system in Equation 2-2 can be lifted similarly as:

$$x_{i_p} = \tilde{A}^p x_i + \tilde{\mathcal{K}}_p^u u_{i,p} + \tilde{\mathcal{K}}_p^w w_{i,p} + \tilde{\mathcal{K}}_p^y y_{i,p} \quad (2-5a)$$

$$y_{i_p, f} = \tilde{\Gamma}_f x_{i_p} + \tilde{\mathcal{T}}_f^u u_{i_p, f} + \tilde{\mathcal{T}}_f^w w_{i_p, f} + (I - \tilde{\mathcal{H}}_f) y_{i_p, f} + e_{i_p, f} \quad (2-5b)$$

For Data-Driven Control (DDC), it is desirable to minimize the effect of elements that cannot be directly measured, such as the initial states sequence  $x_i$  in Equation 2-5a. A common assumption in subspace identification, as stated in Assumption 2, is applied to eliminate the effect of the initial state sequence. Hence, the unknown state sequence can be rewritten as:

$$x_{i_p} = \tilde{A}^p x_i + \tilde{\mathcal{K}}_p^z z_{i,p} \approx \tilde{\mathcal{K}}_p^z z_{i,p} \Leftarrow p \gg 0. \quad (2-6)$$

When the approximation of Equation 2-6 gets applied to Equation 2-4b and Equation 2-5b, the previously defined data equations for the output can be rewritten respectively as:

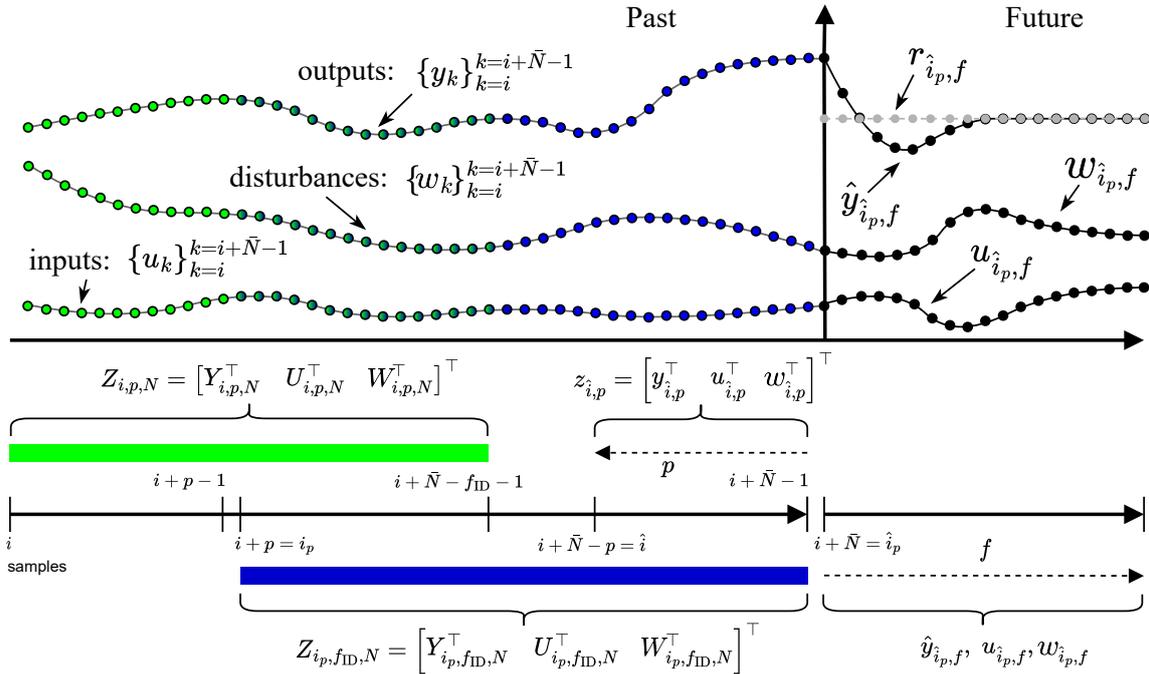
$$y_{i,p,f} = \Gamma_f \tilde{\mathcal{K}}_p^y y_{i,p} + \Gamma_f \tilde{\mathcal{K}}_p^u u_{i,p} + \Gamma_f \tilde{\mathcal{K}}_p^w w_{i,p} + \mathcal{T}_f^u u_{i,p,f} + \mathcal{T}_f^w w_{i,p,f} + \mathcal{H}_f e_{i,p,f} \quad (2-7a)$$

$$y_{i,p,f} = \tilde{\Gamma}_f \tilde{\mathcal{K}}_p^y y_{i,p} + \tilde{\Gamma}_f \tilde{\mathcal{K}}_p^u u_{i,p} + \tilde{\Gamma}_f \tilde{\mathcal{K}}_p^w w_{i,p} + \tilde{\mathcal{T}}_f^u u_{i,p,f} + \tilde{\mathcal{T}}_f^w w_{i,p,f} + (I - \tilde{\mathcal{H}}_f) y_{i,p,f} + e_{i,p,f} \quad (2-7b)$$

To further develop this framework, it is useful to explore the structural properties of the involved matrices. By comparing Equation 2-7a and Equation 2-7b and since  $\mathcal{H}_f$  is invertible it follows that [53]:

$$\begin{bmatrix} \Gamma_f \tilde{\mathcal{K}}_p^z & \mathcal{T}_f^u & \mathcal{T}_f^w & \mathcal{H}_f \end{bmatrix} = \tilde{\mathcal{H}}_f^{-1} \begin{bmatrix} \tilde{\Gamma}_f \tilde{\mathcal{K}}_p^z & \tilde{\mathcal{T}}_f^u & \tilde{\mathcal{T}}_f^w & I \end{bmatrix}. \quad (2-8)$$

This shows that the innovation form in Equation 2-7a and the predictor form in Equation 2-7b can be easily related to each other. This relation is demonstrated to be helpful in the next subsection.



**Figure 2-1:** In this figure, the offline data collection set is divided into two subsets, blue and green, to establish the identification of the output predictor for closed-loop SPC. In standard SPC, the identification horizon  $f_{ID}$  is typically chosen equal to the prediction horizon  $f$ , whereas in closed-loop SPC identification is instead carried out with a  $f_{ID} = 1$ . Furthermore, the most recent  $p$  measurements are used to initialize the optimal controller, and then the controller predicts the inputs and output over the future horizon of length  $f$ . This figure is based on a figure used in [37].

### 2-2-2 Constructing an output predictor

The equations presented earlier incorporate historical data from two distinct data windows, depicted in green and blue in Figure 2-1. This section focuses on deriving a causal predictor for future outputs used for closed-loop SPC, denoted by  $\hat{y}_{i_p,f}$ . The aim is to determine how the outputs from the blue section can be described using the input, output, and disturbance data from the green section, along with the input and disturbance information from the blue section. Furthermore, a key characteristic of closed-loop SPC is the use of a future window defined as  $f_{ID} = 1$  during the identification process. This approach involves focusing only on the top block-row of one of the data equations, as the top block-row remains identical. This selection is intentional, as it ensures that during closed-loop operation, the inputs and noise utilized for identification remain uncorrelated. By doing so, this method minimizes identification bias, which could otherwise degrade the controller's performance [54]. Additionally, this formulation requires estimating fewer parameters, which is particularly advantageous for the adaptive RLS algorithm discussed later.

The used dataset for identification has a length of  $\bar{N}$  and is typically selected to ensure that the number of columns  $N$ , where  $N = \bar{N} - p$ , in the block-Hankel matrices is significantly larger, typically by a factor of 100, than  $p$  and  $f_{ID}$  [26]. The historically collected data is divided into two overlapping sections: the "past" and the "future". These sections are visually represented as green and blue bars, respectively, as shown in Figure 2-1. Data from the green segment is structured into block-Hankel matrices, with inputs organized as  $U_{i,p,N}$ , disturbances as  $W_{i,p,N}$ , and outputs as  $Y_{i,p,N}$ . Similarly, the blue segment is used to construct matrices  $U_{i_p,1,N}$ ,  $W_{i_p,1,N}$ , and  $Y_{i_p,1,N}$ . For this purpose the starting index for the future window,  $i_p = i + p$ , is introduced.

To derive the closed-loop SPC algorithm, additional data matrices are incorporated. Specifically, the last  $p$  input-disturbance-output samples of the dataset, associated with a dashed arrow pointing left, are organized into  $y_{i,p}$ ,  $u_{i,p}$ , and  $w_{i,p}$ . Starting from  $\hat{i} = i + \bar{N} - p$ . The vectors  $\hat{y}_{i_p,f}$ ,  $u_{i_p,f}$  and  $w_{i_p,f}$  all corresponding to the dashed arrow pointing to the right, are considered in the future horizon. Among these,  $\hat{y}_{i_p,f}$  and  $u_{i_p,f}$  are a priori unknown and treated as optimization variables, whereas the future disturbance  $w_{i_p,f}$  is assumed to be fully known, as it is measurable ahead of time.

The objective of this disturbance-feedforward closed-loop SPC is computing optimal future inputs,  $u_{i_p,f}$ , such that  $\hat{y}_{i_p,f}$  follows a predefined reference trajectory  $r_{i_p,f}$  with the influence of a known disturbance  $w_{i_p,f}$ . This is done by the formulation of an output predictor that is linearly dependent on known and unknown data trajectories.

Following this concept, and based on Equation 2-7, the proposed output predictor of the innovation form and predictor form are respectively:

$$\hat{y}_{i_p,f} = \widehat{\Gamma}_f \widehat{\tilde{K}}_p^z z_{i,p} + \widehat{\mathcal{T}}_f^u u_{i_p,f} + \widehat{\mathcal{T}}_f^w w_{i_p,f}, \quad (2-9a)$$

$$\hat{y}_{i_p,f} = \widehat{\Gamma}_f \widehat{\tilde{K}}_p^z z_{i,p} + \widehat{\mathcal{T}}_f^u u_{i_p,f} + \widehat{\mathcal{T}}_f^w w_{i_p,f} + (I - \widehat{\mathcal{H}}_f) \hat{y}_{i_p,f}, \quad (2-9b)$$

Here,  $(\hat{\cdot})$  denotes that the vectors and matrices that are estimated. The output predictors in Equation 2-9 predict outputs over a horizon  $f$ , as indicated in Figure 2-1, where also

the indices of  $\hat{i}$  and  $\hat{i}_p$  are defined. In Equation 2-9a, corresponding to the innovation form, the output predictor is defined explicitly, as the predicted output  $\hat{y}_{i_p,f}$  appears only on the left-hand side. In contrast, the predictor form in Equation 2-9b is defined implicitly, as  $\hat{y}_{i_p,f}$  appears on both sides of the equation.

Closed-loop SPC utilizes only the top block row of the output predictors in Equation 2-9 for identification, which is identical for both the innovation and predictor forms due to their shared structure. This block row consists explicitly of  $C\tilde{\mathcal{K}}_p^z$ ,  $D_u$ , and  $D_w$ , and therefore yields the same identified matrices in both cases during the identification step. Although the full predictor structures differ, both formulations can be employed for closed-loop SPC, as will become clear later in this section.

The top block rows can be easily estimated with a linear regression approach, which can be solved through the following least squares formulation:

$$\left[ \widehat{C\tilde{\mathcal{K}}_p^z} \quad \widehat{D_u} \quad \widehat{D_w} \right] = \arg \min_{\theta} \left\| Y_{i_p,1,N} - \theta \begin{bmatrix} Z_{i_p,1,N} \\ U_{i_p,1,N} \\ W_{i_p,1,N} \end{bmatrix} \right\|_F^2, \quad (2-10)$$

where  $\|\cdot\|_F$  denotes the Frobenius norm and  $\theta$  is the optimization variable representing the estimated Markov parameters. This least squares equation for identifying the Markov parameters is constructed from the Hankel data matrices defined in Figure 2-1, with the data trajectories in the green and blue bars. The solution for the Markov parameters that are estimated with the solved identification problem in Equation 2-10 is unique if the input and disturbance data is sufficiently persisting exciting of order  $m(p + f_{ID}) + n$  and  $q(p + f_{ID}) + n$ , respectively, giving Assumption 4. Due to the persistently exciting conditions the data matrix on the right-hand side of the Markov parameters  $\theta$  in Equation 2-10 should have full row rank. Meaning the amount of columns  $N$  of the Hankel matrices should be  $N \geq (m + q + \ell)p + (m + q)f_{ID}$ . This condition of the full rank ensures that the right inverse of the data matrix on the right-hand side of the Markov parameters exist, enabling a unique solution to the least-squares problem.

The goal is to construct one of the output predictors as introduced in Equation 2-9a and Equation 2-9b. This is needed to formulate the closed-loop SPC problem to predict the outputs over the full future horizon for the receding horizon control method. However, the least-squares identification step in Equation 2-10 provides only the first block row of this predictor. Constructing the rest of the block rows start with looking again at Assumption 2, which state that  $\tilde{A}^p \approx 0$ . With this assumption, the lower triangular block rows in  $\tilde{\Gamma}^f \tilde{\mathcal{K}}_p^u$  can be approximated to zero as shown here:

$$\tilde{\Gamma}^f \tilde{\mathcal{K}}_p^u = \begin{bmatrix} C\tilde{A}^{p-1}\tilde{B}_u & C\tilde{A}^{p-2}\tilde{B}_u & \dots & C\tilde{B}_u \\ 0 & C\tilde{A}^{p-1}\tilde{B}_u & \dots & C\tilde{A}\tilde{B}_u \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & C\tilde{A}^{p-1}\tilde{B}_u \end{bmatrix}, \quad (2-11)$$

with Assumption 2, the same form as shown above can be approximated for  $\tilde{\Gamma}^f \tilde{\mathcal{K}}_p^y$  and  $\tilde{\Gamma}^f \tilde{\mathcal{K}}_p^w$ . Looking at Equation 2-11, it becomes clear that the upper triangle block rows are time shifted versions of the estimated Markov parameter  $\widehat{C\tilde{\mathcal{K}}_p^z}$  in Equation 2-10. With the help of this

assumption, the matrices of the output predictor in predictor form matrices in Equation 2-9b can be fully formulated including  $\mathcal{H}_f$  [45]. Another way to obtain the full set of block rows in Equation 2-9b, starting from the first identified block row, is to form them recursively over the prediction horizon  $f$ . This procedure produces the same predictor form output predictor as in Equation 2-9b. Finally, to express the predictor in innovation form, the relationship in Equation 2-8 is used. This demonstrates how the predictor form matrices can be transformed into the innovation-form matrices, leading to the complete formulation in Equation 2-9a. Both output predictor formulations can be used within the receding horizon control implementation of closed-loop SPC.

## 2-3 Recursive solution of the parameter estimation problem

This section develops an adaptive SPC scheme by recursively estimating the system's Markov parameters. To maintain adaptability, an exponential forgetting factor is applied in the recursive least squares (RLS) algorithm, weighing old data exponentially less than recent data. A square root formulation is introduced to enhance numerical stability, and directional forgetting is employed to relax excitation conditions. In the context of this thesis, the method will eventually be used to approach the nonlinear dynamics in the wind turbine simulation by adaptively determining local linearizations from data.

### 2-3-1 Recursive Least Squares (RLS)

This subsection outlines the process of estimating the Markov parameters in real-time through a RLS approach. The derivation begins with a least-squares-based estimation of the predictor Markov parameters, as defined in Equation 2-10. The formulation in Equation 2-10 addresses offline identification from a batch of input-disturbance-output measurements gathered in Hankel matrices. For the RLS, a reformulation is applied to adapt to the arrival of new input-disturbance-output samples at each time instant. To arrive at the RLS, first the least-squares problem in Equation 2-10 is repeated in a shorter notation as follows:

$$\hat{\theta}_k = \arg \min_{\theta} \|Y_{i_p,1,N} - \theta \Phi_k\|_F^2, \quad (2-12)$$

with:

$$\Phi_k = \begin{bmatrix} Z_{i_p,1,N} \\ U_{i_p,1,N} \\ W_{i_p,1,N} \end{bmatrix}, \quad \theta = \begin{bmatrix} C\tilde{\mathcal{K}}_p^z & D_u & D_w \end{bmatrix},$$

The full-rank least-squares solution of Equation 2-12 is expressed as:

$$\theta_k = Y_{i_p,1,N} \Phi_k^\top (\Phi_k \Phi_k^\top)^{-1}. \quad (2-13)$$

Equation 2-14, denotes the data regressor  $\phi_{k+1}$  at timestep  $k+1$ , which is appended at each time step as an additional column to  $\Phi_k$  resulting in  $\Phi_{k+1}$ .

$$\phi_{k+1} = \begin{bmatrix} z_{i_p,1}^{\hat{}} \\ u_{i_p,1}^{\hat{}} \\ w_{i_p,1}^{\hat{}} \end{bmatrix} \quad (2-14)$$

Relating the time update  $k + 1$  to Figure 2-1 it can be interpreted as appending the newly obtained data to the past data sequence, giving  $N + 1$  total data points assuming  $i = 1$ . Similarly, the current output measurement  $y_{k+1}$  is added to  $Y_{i_p,1,N}$ . This ensures that the new data is incorporated into the Markov parameter estimation  $\hat{\theta}_{k+1}$  of the next time step.

With  $y_{k+1}$  and  $\phi_{k+1}$  being available, the least-squares problem can be formulated as:

$$\theta_{k+1} = \left( Y_{i_p,1,N} \Phi_k^\top + y_{k+1} \phi_{k+1}^\top \right) \underbrace{\left( \Phi_k \Phi_k^\top + \phi_{k+1} \phi_{k+1}^\top \right)^{-1}}_{=P_{k+1}=\mathcal{I}_{k+1}^{-1}}. \quad (2-15)$$

The covariance matrix  $P_k$  and the information matrix  $\mathcal{I}_k$  are defined as:

$$P_k = \left( \Phi_k \Phi_k^\top \right)^{-1}, \quad \mathcal{I}_k = \Phi_k \Phi_k^\top.$$

Since directly inverting  $\Phi_k^\top \Phi_k + \phi_{k+1} \phi_{k+1}^\top$  at every time step would be computationally expensive, we apply the Matrix Inversion Lemma [55]:

$$\begin{aligned} P_{k+1} &= \left( P_k^{-1} + \phi_{k+1} \phi_{k+1}^\top \right)^{-1} \\ &= P_k - P_k \phi_{k+1} \left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1} \phi_{k+1}^\top P_k. \end{aligned} \quad (2-16)$$

It is worth noting that this expression involves inverting a  $\ell \times \ell$  matrix only, so a scalar for  $\ell = 1$ . Substituting the expression for  $P_{k+1}$  from Equation 2-16 and the previous estimate  $\theta_k$  from Equation 2-13 into the parameter update in Equation 2-15 yields:

$$\begin{aligned} \theta_{k+1} &= \theta_k \left( I - \phi_{k+1} \left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1} \phi_{k+1}^\top P_k \right) + y_{k+1} \phi_{k+1}^\top P_{k+1} \\ &= \theta_k - \theta_k \phi_{k+1} \underbrace{\left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1} \phi_{k+1}^\top P_k}_{\text{underbraced part}} + y_{k+1} \phi_{k+1}^\top P_{k+1}. \end{aligned} \quad (2-17)$$

This equation updates the parameter vector  $\theta_{k+1}$  using the new data  $y_{k+1}$  and the updated covariance matrix  $P_{k+1}$ . The term  $\left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1}$  ensures that the update is weighted appropriately based on the new data. Furthermore, the underbraced part in Equation 2-17 can be rewritten in the following form:

$$\begin{aligned} &\left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1} \phi_{k+1}^\top P_k \\ &= \left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1} \left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} - \phi_{k+1}^\top P_k \phi_{k+1} \right) \phi_{k+1}^\top P_k \\ &= \phi_{k+1}^\top \left( P_k - P_k \phi_{k+1} \left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1} \phi_{k+1}^\top P_k \right) \end{aligned}$$

Substituting  $P_{k+1}$  of Equation 2-16 in the above equation results in:

$$\left( 1 + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1} \phi_{k+1}^\top P_k = \phi_{k+1}^\top P_{k+1}$$

This reformulation of the underbraced part in Equation 2-17 enables the parameter update to be expressed in a more compact form:

$$\theta_{k+1} = \theta_k + \phi_{k+1}^\top P_{k+1} (y_{k+1} - \theta_k \phi_{k+1}) \quad (2-18)$$

After computing the covariance update using Equation 2-16 and applying it in the final parameter update expression in Equation 2-18, the parameter vector  $\theta_{k+1}$  is updated based on the prediction error  $y_{k+1} - \theta_k \phi_{k+1}$  and the RLS gain  $\phi_{k+1}^\top P_{k+1}$ . This gain increases when the regressor  $\phi_{k+1}$  excites a direction with high uncertainty, and decreases when the direction is associated with higher confidence.

### 2-3-2 Exponential forgetting RLS

In practice, RLS without forgetting may suffer from reduced adaptability, as older data can dominate the estimation. To overcome this limitation, a forgetting mechanism is commonly incorporated to reduce the influence of past data and emphasize recent data.

In exponential forgetting, the forgetting factor  $0 < \lambda_{exp} < 1$  determines the relative weighing of past data. From a least-squares perspective, data that is  $n$  time steps old is weighted by  $\lambda_{exp}^n$ , which decreases exponentially with  $n$  and converges to zero. This ensures that older data has diminishing influence over time. A common way to relate the forgetting factor  $\lambda_{exp}$  to its effective memory length  $N_w$  is given by [56]:

$$N_w = \frac{1}{1 - \lambda_{exp}}. \quad (2-19)$$

This modification ensures that the RLS algorithm maintains a finite memory, rather than accumulating information indefinitely. Explicitly in RLS formulation, this approach enhances the adaptiveness of the parameter estimates by preventing the covariance matrix from becoming overly confident in certain directions, which would correspond to eigenvalues converging to zero. To implement exponential forgetting, the covariance matrix is scaled by the factor  $\lambda_{exp}^{-1}$  before applying the standard covariance update of Equation 2-16. This leads to the following intermediate step:

$$\bar{P}_k = \frac{1}{\lambda_{exp}} P_k. \quad (2-20)$$

Substituting  $\bar{P}_k$  into the covariance update in Equation 2-16 yields:

$$\begin{aligned} P_{k+1} &= \bar{P}_k - \bar{P}_k \phi_{k+1} \left( 1 + \phi_{k+1}^\top \bar{P}_k \phi_{k+1} \right)^{-1} \phi_{k+1}^\top \bar{P}_k \\ &= \frac{1}{\lambda_{exp}} P_k - \frac{1}{\lambda_{exp}} P_k \phi_{k+1} \left( \lambda_{exp} + \phi_{k+1}^\top P_k \phi_{k+1} \right)^{-1} \phi_{k+1}^\top P_k. \end{aligned} \quad (2-21)$$

In practical implementations, finite numerical precision can cause the covariance matrix in the RLS approach to lose its positive definiteness after repeated covariance updates. When the covariance matrix becomes ill-conditioned, its smallest eigenvalues becomes sensitive to rounding errors and may turn negative. As a result, the matrix can lose its positive definiteness. To prevent this, the implementation of the square-root algorithm described in [57] is

adopted in this work. Rather than updating the covariance matrix itself, this method work with a Cholesky factor and use orthogonal transformations at each iteration. This ensures the factor remains numerically stable and preserves the covariance matrix its positive definiteness.

For the square-root method  $P_k = R_k R_k^\top$ , with  $R_k$  being the lower-triangular Cholesky factor of  $P_k$ . The covariance update of Equation 2-16 with Cholesky factor  $R_k$  incorporated is given as [57]:

$$R_{k+1} R_{k+1}^\top = \frac{1}{\lambda_{exp}} R_k R_k^\top - \frac{1}{\lambda_{exp}} R_k R_k^\top \phi_{k+1} \left( \lambda_{exp} + \phi_{k+1}^\top R_k R_k^\top \phi_{k+1} \right)^{-1} \phi_{k+1}^\top R_k R_k^\top. \quad (2-22)$$

The above equation can the subsequently be factorized in a pre- and post-array as follows:

$$\underbrace{\begin{bmatrix} 1 & \lambda_{exp}^{-\frac{1}{2}} \phi_{k+1}^\top R_k \\ 0 & \lambda_{exp}^{-\frac{1}{2}} R_k \end{bmatrix}}_{\text{pre-array}} Q = \underbrace{\begin{bmatrix} \gamma_{k+1}^{-\frac{1}{2}} & 0 \\ G_{k+1} \gamma_{k+1}^{-\frac{1}{2}} & R_{k+1} \end{bmatrix}}_{\text{post-array}}, \quad (2-23)$$

with:

$$\gamma_{k+1} = \left( 1 + 1/\lambda_{exp} + \phi_{k+1}^\top R_k R_k^\top \phi_{k+1} \right)^{-1}$$

$$G_{k+1} = R_k R_k^\top \phi_{k+1} \left( \lambda_{exp} + \phi_{k+1}^\top R_k R_k^\top \phi_{k+1} \right)^{-1}$$

By taking the transpose of both sides of Equation 2-23 and multiplying them, it can be verified that this results in Equation 2-22. The process involves forming a pre-array as in Equation 2-23 and applying orthogonal transformations  $Q$  to achieve a lower-triangular post-array. From this post-array, the updated Cholesky factor  $R_k$  is derived. The parameter update is then expressed as follows:

$$\theta_{k+1} = \theta_k + G_{k+1} (y_{k+1} - \theta_k \phi_{k+1}) \quad (2-24)$$

### 2-3-3 Directional forgetting RLS

With exponential forgetting, uniformly forgetting all past information can cause the information matrix  $\mathcal{I}$  to become singular over time if the regressors  $\phi_k$  do not explore all directions in  $\mathbb{R}^n$ , a problem linked to persistent excitation. To address this, in [46] a directional forgetting method is introduced, which decomposes the information matrix as  $\mathcal{I}_k = \mathcal{I}_k^{(1)} + \mathcal{I}_k^{(2)}$ , where updates are orthogonal to  $\mathcal{I}_k^{(1)}$  and  $\mathcal{I}_k^{(2)}$  is a rank-one matrix satisfying  $\mathcal{I}_k^{(2)} \phi_k = \mathcal{I}_k \phi_k$ . This decomposition ensures only the direction of new data is forgotten. Similar to exponential forgetting, an intermediate update step is performed on the covariance matrix. In the case of directional forgetting, this update is first expressed using the information matrix  $\bar{\mathcal{I}}_k$ , which is then transformed back into covariance form  $\bar{P}_k$  to complete the covariance update  $P_{k+1}$ .

Directional forgetting, represented by  $\bar{\mathcal{I}}_k$ , is defined as:

$$\bar{\mathcal{I}}_k = \mathcal{I}_k \Pi_{\phi_k}^\perp + \lambda_{dir} \mathcal{I}_k \Pi_{\phi_k}, \quad (2-25)$$

where  $\Pi_{\phi_k}$  and  $\Pi_{\phi_k}^\perp$  project onto  $\phi_k$  and its orthogonal complement, respectively. They are defined such that both projections are symmetric, ensuring that  $\bar{\mathcal{I}}_k$  inherits the symmetry of  $\mathcal{I}_k$ , which is crucial since  $\bar{\mathcal{I}}_k$  will later be inverted to obtain the covariance matrix for the RLS update. The orthogonal projection matrices in Equation 2-25 are defined as:

$$\Pi_{\phi_k} = \frac{\phi_k \phi_k^\top \mathcal{I}_k}{\phi_k^\top \mathcal{I}_k \phi_k}, \quad \Pi_{\phi_k}^\perp = I - \Pi_{\phi_k}. \quad (2-26)$$

Next, to arrive at a formulation where it becomes clear how  $\lambda_{dir}$  is applied across the full  $\mathcal{I}_k$  the expression  $\Pi_{\phi_k}^\perp = I - \Pi_{\phi_k}$  is substituted into (2-25) yielding:

$$\bar{\mathcal{I}}_k = \mathcal{I}_k - (1 - \lambda_{dir}) \mathcal{I}_k \Pi_{\phi_k} \quad (2-27)$$

Defining the matrix  $M_k$  as [46]:

$$M_k = \frac{(1 - \lambda_{dir}) \phi_k \phi_k^\top \mathcal{I}_k}{\phi_k^\top \mathcal{I}_k \phi_k}$$

and substituting the projection matrix  $\Pi_{\phi_k}$  from Equation 2-26 into Equation 2-27 results in the compact form:

$$\bar{\mathcal{I}}_k = (I - M_k) \mathcal{I}_k \quad (2-28)$$

In the expression above,  $I - M_k$  is called the forgetting matrix. It has a single eigenvalue of  $\lambda_{dir}$  along the direction of  $\phi_k$ , while all other eigenvalues remain equal to one, ensuring that forgetting is applied selectively in the subspace containing new information.

As mention before, in RLS a covariance matrix is updated, so  $P_k = \mathcal{I}_k^{-1}$  is needed. To get this inverse, the inverse of Equation 2-28 is taken and results in:

$$\bar{P}_k = P_k + \alpha_k \phi_k \phi_k^\top, \quad (2-29)$$

where  $\alpha_k = \frac{1 - \lambda_{dir}}{\lambda_{dir} \phi_k^\top \mathcal{I}_k \phi_k} \geq 0$ .

To make sure that  $\alpha$  is correctly specified,  $\phi_k$  needs to be nonzero. This can be easily implemented with a dead zone defined as  $\alpha_k = 0$ , if  $\|\phi_k\|_2^2 < \epsilon$ . This can also be used to set  $\epsilon$  bigger than zero based on the noise present in the data. This prevents forgetting in directions of  $\phi_k$ , which carries mostly noise rather than informative data. Looking at Equation 2-29, clearly no-forgetting is applied with  $\alpha_k = 0$ .

A limitation of the directional forgetting algorithm is its reliance on the matrix  $\mathcal{I}_k$  to compute  $\alpha_k$  in equation Equation 2-29. This requirement necessitates either simultaneous updates of the information and covariance matrices or repeated matrix inversion of  $\mathcal{I}_k$ . A more efficient alternative is the previously discussed square-root formulation, leveraging the Cholesky decomposition  $P_k = R_k R_k^\top$ .

Substituting this into the update Equation 2-29 yields:

$$\bar{R}_k \bar{R}_k^\top = R_k R_k^\top + \alpha_k \phi_k \phi_k^\top$$

which can be implemented as:

$$\underbrace{\begin{bmatrix} \sqrt{\alpha_k} \varphi_k & R_k \end{bmatrix}}_{\text{pre-array}} Q = \underbrace{\begin{bmatrix} 0 & \bar{R}_k \end{bmatrix}}_{\text{post-array}}, \quad (2-30)$$

Here,  $Q$  serves as an orthogonal transformation to make the first column of the post-array exist out of zeros. With the help of Equation 2-30,  $\alpha_k$  can now be written as:

$$\alpha_k = \frac{1 - \lambda_{dir}}{\lambda_{dir}} \frac{1}{\|R_k^{-1} \phi_k\|_2^2},$$

which avoids explicit matrix inversion by using forward substitution.

In short, directional forgetting integrates seamlessly into an RLS framework, ensuring forgetting occurs only when sufficient new data is available, reducing the need for persistent excitation.

# Simulation setup and results

## 3-1 Simulation setup

This section introduces the simulation environment used to evaluate the adaptive closed-loop Subspace Predictive Control (SPC) with wind preview integration for rated rotor speed tracking. The high-fidelity wind turbine simulator QBlade and the DTU 10 MW reference wind turbine are described, which together provide a realistic and nonlinear test environment representative of large-scale wind turbine operation. In addition, a Continuous-Wave (CW) Light Detection and Ranging (LIDAR) simulator is presented to model the effects of upstream wind measurements and generate preview information for disturbance feedforward used in the adaptive closed-loop SPC algorithm. Finally, the procedure for open-loop data collection is outlined, which provides the data used to initialize the output predictor with the estimated initial Markov parameters, later used for the controller design in section 3-2.

### 3-1-1 Wind turbine simulator and wind turbine model

#### Wind turbine simulator

To evaluate the performance of the adaptive closed-loop SPC on a wind turbine simulation, the high-fidelity simulation environment QBlade is utilized. To this end, QBlade has been specifically designed to enable the application of advanced simulation models within the wind turbine design and certification process, aiming to balance high-fidelity modeling, with computational efficiency. This has been done by exploiting modern hardware capabilities, such as GPU-based parallelization, to perform accurate and large-scale aeroelastic simulations at manageable computational costs [48]. Among its advanced aerodynamic solvers, QBlade employs the Unsteady Blade Element Momentum (UBEM) method to accurately model rotor flow dynamics, which is crucial for analyzing blade load distributions. UBEM builds upon the classical Blade Element Momentum (BEM) theory, which assumes steady-state conditions and relies on a balance of aerodynamic forces at each blade section. By incorporating time-dependent effects, such as dynamic inflow and wind shear, UBEM extends this approach

to capture time-varying dynamics caused by changing wind conditions and wind turbine dynamics [58]. These load distributions directly influence the aerodynamic torque generated by the rotor, and therefore play a critical role in shaping the rotor speed dynamics relevant to the rotor speed tracking.

Furthermore, a notable feature of QBlade is its Software in Loop (SIL) interface, allowing the simulation engine to function as a Dynamic Link Library (DLL). This enables external applications to programmatically load projects, control simulation loops, and exchange data in real-time with QBlade, facilitating integration with tools like Python and MATLAB. In this thesis, the SIL feature of QBlade is used in combination with MATLAB.

In summary, QBlade's high-fidelity dynamics make it a powerful tool for realistic wind turbine simulations, providing a detailed representation of the nonlinear behaviors inherent to large-scale turbines. However, this level of detail introduces additional challenges for control design, as the system can exhibit variability depending on the operating point due to the nonlinearity. To address this, adaptive closed-loop SPC will be employed in this thesis to continuously update the predictive model, thereby enabling adaptation to changing wind turbine operating conditions. This approach helps bridge the gap between theoretical controller design and real-world performance, where adaptation to varying conditions is essential when using high-fidelity models.

## Wind turbine model

The reference wind turbine used in this study is the DTU 10 MW wind turbine model, developed by the Technical University of Denmark (DTU) [59]. Given the ongoing trend toward larger wind turbines, the 10 MW rating of the DTU wind turbine makes it representative of current large-scale onshore wind turbine designs. Additionally, this reference turbine ensures compatibility with the utilized simulation framework QBlade. An overview of the main design characteristics of the wind turbine are summarized in Table 3-1.

**Table 3-1:** Overview of main design parameters for the DTU 10 MW reference wind turbine [60].

Description	Value
Rated power	10 MW
Rotor setup	Upwind configuration with three blades
Control strategy	Variable speed and collective blade pitch
Drivetrain	Medium-speed drivetrain with a multi-stage gearbox
Rotor Diameter / Hub Diameter	178.3 m / 5.6 m
Hub height	119 m
Wind Speed Limits (cut-in / rated / cut-out)	4 m/s / 11.4 m/s / 25 m/s
Rotor Speed (cut-in / rated)	6 RPM / 9.6 RPM
Rated generator torque	$1.99 \times 10^5$ Nm
Blade pitch range	0° to 90°
Maximum blade pitch rate	10°/s

### 3-1-2 LIDAR simulator

Accompanying the simulated wind turbine will be a LIDAR simulator which can mimic the effects of a real-life LIDAR system. To utilize the LIDAR simulator, a choice has to be made between the Pulsed-Wave (PW) and CW, which were discussed in section 1-2-1. Each approach has advantages and disadvantages. PW LIDAR provides range resolution, meaning it can capture wind speed at multiple distances along the beam path simultaneously by sending out short laser pulses and measuring the time of flight of the backscattered signal. In contrast, CW LIDAR emits a continuous beam and sets the measurement distance optically, which creates a range-dependent weighting of the backscattered signal. In [33] it is shown that for ranges under 125 m, CW LIDAR achieves a lower Root Mean Squared Error (RMSE) for wind estimations than PW LIDAR. In this work, the relevant measurement distances remain at a single distance below 125 m, making CW LIDAR the chosen method.

The measurement process of CW LIDAR relies on focusing a continuous laser beam at a specific range, defining the region around a focal point from which most of the backscattered light is collected. This region is known as the probe volume, and it determines where along the beam axis the wind information is most effectively gathered. The system measures Doppler shifts in the backscattered signal to determine wind speed. The strength of the backscatter signal from each axial position contributes to the final wind speed reading. To model the spatial sensitivity of the measurement, a Lorentzian weighting function  $F(\Delta)$  is employed. In the simulator, the CW LIDAR measurement is computed as a weighted sum of wind velocities along the beam axis, where the weights of the Lorentzian profile peak at the focal distance and decay symmetrically with the distances above and below it. This Lorentzian function is expressed as:

$$F(\Delta) = \frac{\Gamma}{\pi(\Delta^2 + \Gamma^2)} \quad (3-1)$$

where  $\Delta$  is the axial distance from the focal plane. The parameter  $\Gamma$  in the Lorentzian is the half-width of the weighting function, meaning the distance at which  $F(\Delta)$  drops to half of its maximum value [61]. The range resolution in CW LIDAR is therefore determined by the optical system's focusing capability rather than time gating, as in PW LIDAR systems.

In Equation 3-1  $\Gamma$  is defined as follow:

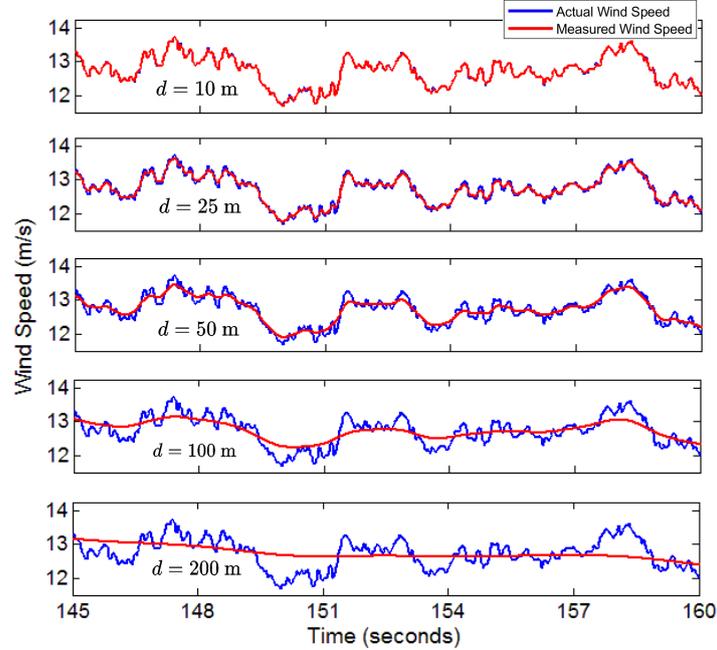
$$\Gamma = \frac{\psi d^2}{\pi A^2},$$

For the equation of  $\Gamma$ , parameters for the industry-used ZephIR LIDAR system are available. The parameters which mimic the LIDAR effect of the ZephIR LIDAR are [33]:

- $\psi = 1.55 \times 10^{-6}$  m: The laser wavelength.
- $A = 28 \times 10^{-3}$  m: The beam radius,

and  $d$  the focal distance in meters. At longer distances, the probe volume in CW LIDAR expands significantly, which can lead to averaging effects and reduced spatial resolution.

Figure 3-1 illustrates how increasing the focal distance  $d$  reduces high-frequency fluctuations of the turbulent wind field measurements, confirming that a larger probe volume leads to stronger averaging effects.



**Figure 3-1:** Effect of focal distance  $d$  on ZephIR LIDAR measurements: wind speed measurements obtained using a Lorentzian-weighted ZephIR LIDAR model are compared with the true wind speed. The figure shows how increasing the focal distance  $d$ , resulting in a broader probe volume, averages out high-frequency wind fluctuations of the wind field.

### 3-1-3 Simulation overview

The simulation environment is configured by initializing QBlade with the 10 MW DTU wind turbine model and applying the simulation parameters defined in this section via MATLAB, utilizing QBlade's SIL interface. The simulation parameters are chosen to ensure that all relevant data required for the design of the adaptive closed-loop SPC for rated rotor speed tracking will be accessible through QBlade. The key simulation parameters are summarized in Table 3-2. The used sampling time of  $0.05$  s is chosen to match the high-fidelity multiphysics solver of QBlade.

The wind turbine is actuated through a single input channel ( $n_u = 1$ ) representing the pitch angle command  $\beta$ . The pitch actuator is subject to the following operational limits defined in the QBlade DTU 10MW project file:

$$\beta_{\min} = 0^\circ, \quad \beta_{\max} = 27^\circ, \quad \Delta\beta_{\max} = 10^\circ/\text{s}.$$

The simulations feature a single measured output ( $n_y = 1$ ), the rotor speed  $\Omega$ . Furthermore, a single disturbance channel ( $n_w = 1$ ) is defined by the wind speed  $v$ .

The rated generator torque is set to  $T_{\text{rated}} = 1.99 \times 10^5$  Nm, corresponding to the torque necessary to achieve rated power when operating at the rated rotor speed.

Additionally, the discussed CW LIDAR simulator, based on the commercial ZephIR system described in subsection 3-1-2, is also included in the simulation. An overview of the corresponding parameters is provided in Table 3-2.

**Table 3-2:** Simulation parameters used in QBlade for the 10 MW DTU wind turbine model and the CW LIDAR simulator.

<b>QBlade simulator parameters</b>	
<b>Parameter</b>	<b>Value</b>
Sampling time, $T_s$	0.05 s
Number of outputs, $n_y$	1
Number of inputs, $n_u$	1
Number of disturbances, $n_w$	1
Rated generator torque, $T_{\text{rated}}$	$1.99 \times 10^5$ Nm
Reference: rated rotor speed, $\Omega_r$	9.6 rpm
Minimum pitch control angle, $\beta_{\text{min}}$	$0^\circ$
Maximum pitch control angle, $\beta_{\text{max}}$	$27^\circ$
Maximum pitch control rate, $\Delta\beta_{\text{max}}$	$10^\circ/\text{s}$
<b>LIDAR simulator parameters</b>	
<b>Parameter</b>	<b>Value</b>
Sampling time, $T_s$	0.05 s
Laser wavelength, $\psi$	$1.55 \times 10^{-6}$ m
Beam radius, $A$	$28 \times 10^{-3}$ m

### 3-1-4 Open loop data collection

For the design of the adaptive closed-loop SPC, an initial dataset is collected to initialize the output predictor and its Markov parameters. In this work, the initial dataset is obtained through open-loop data collection, allowing free design of input and disturbance excitations within the system's physical limits. In contrast, closed-loop operation may limit the input variation due to the controller, potentially resulting in less informative data for identification. The relevant signals for the parameter identification are the pitch angle  $u_k = \beta_k$ , which acts as the input to the system; the rotational speed  $y_k = \Omega_k$ , representing the measured output; and the wind speed  $w_k = v_k$ , serving as a measurable disturbance. The data collection is performed by stimulating the wind turbine simulation with excitation input and disturbance signals. For the blade pitch angle, a pseudo-random binary sequence that generates pitch inputs with a variation of  $\pm 2^\circ$  is employed. The input variations saturate with the maximum pitch rate of the wind turbine to realistically capture the actuator's response time. The pitch input variation in  $^\circ/\text{s}$  is set so that it has enough time to saturate before the next input is applied. The wind speed is perturbed around a nominal value by a stochastic component defined as

$$v_k = 15 + 0.5 \xi_k,$$

where  $\xi(k)$  is a normally distributed random variable with a variance of one, introducing a wide spectrum of frequency components into the system. These input and disturbance signals are designed in an attempt to make the excitation sufficiently rich of order  $p + f_{id} + n$ ,

allowing the dynamics to be accurately captured. However, since the underlying wind turbine simulation exhibits nonlinear dynamics, the identified parameters will possibly only provide a local linear approximation.

The input, disturbance and outputs signals are measured and preprocessed by centering and scaling them. The open-loop signals are scaled from  $[-1, 1]$  via a symmetric min-max approach. Each variable  $x$  is first centered by  $x_{\text{center}} = (x_{\text{max}} + x_{\text{min}})/2$  and then scaled by  $x_{\text{halfRange}} = (x_{\text{max}} - x_{\text{min}})/2$ . Here  $x_{\text{min}}$  and  $x_{\text{max}}$  are the minimum and maximum expected values of the data. The symmetric min-max transformation is formulated as:

$$\tilde{x}_k = \frac{x_k - x_{\text{center}}}{x_{\text{halfRange}}}.$$

This symmetric min-max scaling mitigates differences in magnitude of the data and improves numerical conditioning for the Hankel-based identification.

After preprocessing the data, the Hankel matrices are formed and the Least Squares (LS) solution of Equation 2-10 can be solved to compute the Markov parameters in terms of the Moore-Penrose inverse denoted with  $\dagger$  [26]:

$$\hat{\theta}_0 = Y_{i_p,1,N} \begin{bmatrix} Y_{i,p,N} \\ U_{i,p,N} \\ W_{i,p,N} \\ U_{i_p,1,N} \\ W_{i_p,1,N} \end{bmatrix}^{\dagger}$$

Here,  $\hat{\theta}_0$  is defined as the initial guess of Markov parameters,  $\left[ \widehat{C\tilde{K}_p^z} \quad \widehat{D}_u \quad \widehat{D}_w \right]$ , which form the first block row of the output predictors in Equation 2-7 used for closed-loop SPC. The length of the past windows  $p$ , future window  $f$ , and the column dimension  $N$  will be selected based on a sensitivity analysis in the next section.

## 3-2 Adaptive closed-loop SPC controller design

The goal of this section is to formulate an adaptive closed-loop SPC controller that can be applied to the DTU 10 MW wind turbine model in QBlade for rated rotor speed tracking under realistic and nonlinear operating conditions. To achieve this, the section first defines the adaptive closed-loop SPC control problem by specifying the required initialization parameters, data structures, cost function, and constraints, followed by the complete control algorithm described in Algorithm 1. Subsequently, the influence of key design parameters is examined, including the past and future windows, cost weights, forgetting factors, and initialization settings, with particular attention to their relevance in the context of wind turbine control.

### 3-2-1 Optimal control problem formulation

In this subsection, the algorithm of the adaptive closed-loop SPC is formulated. Before the formulation of the algorithm, the initial parameters and data sequences to start the algorithm are defined.

To construct the regressor and determine the dimensions of the relevant matrices introduced later in this subsection, the past window  $p$  and the future window  $f$  for closed-loop SPC must first be specified. The future window  $f$  defines the receding horizon length and sets the horizon for both future output prediction and optimal control in this case. Another key parameter required for initializing the algorithm is the initial  $\theta_0$ , consisting out of the Markov parameters in Equation 3-2 obtained from the open loop data described in subsection 3-1-4. The initial uncertainty of the estimated parameters  $\theta_0$  is defined with the covariance matrix  $P_0$  and is also used for the initial Recursive Least Squares (RLS) update of the Markov parameters when new data becomes available. To adapt the model to changing dynamics, the discussed forgetting factors  $\lambda_{\text{exp}}$  and  $\lambda_{\text{dir}}$  are introduced, influencing the covariance matrix update during parameter estimation.

The tracking performance is weighted by the positive semi-definite matrix  $Q$ , while the control effort is penalized by the positive definite matrix  $R$ . These matrices shape the relative importance between following the output reference trajectory and minimizing input variations. The reference trajectory  $r_{i_p, f}$  will be used as a reference for the the rated rotor speed tracking. To ensure physically feasible inputs, input constraints and rate constraints are incorporated. Specifically, the input sequence  $u$  is constrained to belong to the set  $\mathcal{U}$ , determined by actuator saturation limits of the pitch angle  $\beta_{\min} \leq u \leq \beta_{\max}$  as defined in Table 3-2. The rate of change of the input  $\Delta u$  is constrained by  $\Delta \mathcal{U}$ , reflecting actuator rate limitations, specifically  $\Delta u \leq \Delta \beta_{\max}$ , also defined in Table 3-2. Once these quantities have been specified, the adaptive closed-loop SPC algorithm is fully defined and proceeds according to Algorithm 1.

---

**Algorithm 1** Adaptive closed-loop SPC
 

---

**start loop**

1: With Markov parameters  $\theta_k$  recursively obtain the matrices for the predictor form output predictor  $\hat{y}_{i_p,f}$ , as defined in Equation 2-9b.

2: Solve the predictive control problem:

$$\min_{\Delta u_{i_p,f}, \hat{y}_{i_p,f}} (\hat{y}_{i_p,f} - r_{i_p,f})^\top Q (\hat{y}_{i_p,f} - r_{i_p,f}) + \Delta u_{i_p,f}^\top R \Delta u_{i_p,f}$$

$$s.t. \quad \hat{y}_{i_p,f} = \widehat{\Gamma}_f \widehat{\mathcal{K}}_p^z z_{i_p} + \widehat{\mathcal{T}}_f^u u_{i_p,f} + \widehat{\mathcal{T}}_f^w w_{i_p,f} + (I - \widehat{\mathcal{H}}_f) \hat{y}_{i_p,f}$$

$$u \in \mathcal{U}, \quad \Delta u \in \Delta \mathcal{U},$$

3: Apply the first element  $u_{i_p,1}$  of the optimal input sequence as control input

4: Construct regressor vector  $\phi_{k+1}$  with past data  $z_{i_p}$ , control input  $u_{i_p,1}$ , and known disturbance  $w_{i_p,1}$ , following the structure in Equation 2-14.

5: Update the covariance matrix  $P_k$  to  $P_{k+1}$  with new data regressor  $\phi_{k+1}$ , using the square-root method outlined in Equation 2-23.

Based on the chosen forgetting method, a preliminary covariance update is employed, resulting in the use of  $\bar{P}_k$  for the covariance update  $P_{k+1}$  instead of  $P_k$ . The expression for  $\bar{P}_k$  is given in Equation 2-16 for exponential forgetting and in Equation 2-29 for directional forgetting.

6: Update Markov parameters  $\theta_k$  to  $\theta_{k+1}$  using the square-root RLS parameter update described in Equation 2-24. This square-root RLS update is implicitly equivalent to the traditional RLS parameter update, which is stated here for simplicity:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + P_{k+1} \phi_{k+1} (y_{k+1} - \hat{\theta}_k \phi_{k+1})$$

7:  $k \leftarrow k + 1$ ;

**end loop**


---

As visible in the adaptive closed-loop SPC Algorithm 1, it first constructs the output predictor in predictor form. Then, it formulates and solves an optimization problem over the future horizon  $f$  to determine the optimal control inputs, aiming to achieve the desired tracking of the output reference and stick to the system's constraints. The optimization algorithm is similar to that of Model Predictive Control (MPC), but the difference is that, in the case of SPC, a direct data-driven output predictor is applied to the predictive control framework instead of a state-space model. Similarly, as in MPC, only the first control input within the optimized control horizon  $f$  is applied to the system. Subsequently, the output predictor in predictor form is updated using the RLS algorithm, which may be applied without forgetting, or with exponential or directional forgetting, depending on the selected adaptation strategy. Next, the window in Figure 2-1 moves with one sample to the right by including the newly available data, and the algorithm will repeat itself again. In this work, the optimization problem in step 5 of Algorithm 1 is solved with the help of the Quadratic Programming (QP) solver provided by CasADi [62].

### 3-2-2 Effect of parameter variations

This subsection presents an examination of the effect of parameter variations in the adaptive closed-loop SPC scheme described in Algorithm 1. It considers key design parameters involved in the formulation of the output predictor and controller, including the past and

future windows, cost weights, and initial covariance settings. The forgetting factors are also examined, consisting of the exponential and directional forgetting strategies. The relevance of these parameters in the context of wind turbine control is discussed throughout.

### **Past window $p$**

The parameter  $p$  defines the number of system parameters that are used for estimation of the output predictors in Equation 2-7, together with the later discussed future window  $f$ . In offline identification,  $p$  is typically set to be larger than the expected maximum model order, often several times, to ensure that all relevant dynamics are captured [52]. However, in recursive schemes such as RLS, selecting a larger  $p$  has important implications. It directly increases the computational effort, since more parameters must be updated at every iteration. Moreover, it slows down the convergence rate of the of the parameter estimation algorithm, as more parameters require sufficient excitation and data to be accurately estimated [51]. Additionally, to satisfy Assumption 2,  $p$  is set large enough so that  $\tilde{A}^p \approx 0$ , allowing the effect of the initial state to be neglected. Also,  $p$  plays a role in satisfying Assumption 4, enabling the formulation of a unique output predictor based solely on input-disturbance-output data. Nevertheless, while  $p$  must be sufficiently large to capture the essential wind turbine dynamics, it should not be excessively large to avoid overfitting. An overly large  $p$  may result in the model capturing high-frequency noise, such as wind-speed measurement noise or pitch-sensor noise, which do not represent the true system behavior.

### **Future window $f$**

For the case of this thesis, the future window  $f$  equals both the prediction and control horizon of the adaptive closed-loop SPC, meaning the same window is used to predict system outputs and to determine the number of future control inputs to be optimized. The prediction horizon  $f$  is selected so that the prediction interval encompasses the key dynamics of the system. For wind turbine control, it is particularly important that  $f$  is large enough to incorporate the slow dynamics inherent in the system, such as the rotor speed response to the pitch input caused by inertia. Additionally, the control horizon  $f$  defines how many future control moves the optimizer can adjust, this creates a trade-off between the ability to plan inputs proactively and the associated computational complexity. Moreover, extending the future window increases the role of the model accuracy, as prediction errors propagate over time and can result in suboptimal control inputs on the real system.

In this thesis, the future window is extended by incorporating wind speed predictions obtained from the LIDAR simulation. Incorporating wind speed LIDAR information allows the adaptive closed-loop SPC to foresee upcoming wind variations, leading to potentially improving disturbance rejection and overall performance. However, because these LIDAR predictions contain uncertainty, a longer horizon can also propagate prediction errors if the wind speed forecasts deviate from actual wind conditions. In summary, choosing  $f$  involves balancing the benefits of additional predictive insight against the risks of computational complexity and potential forecast inaccuracy.

### **Weights $Q$ and $R$ , constraints and reference**

Within the framework of adaptive closed-loop SPC, the weighting matrices  $Q$  and  $R$  are used to trade-off between disturbance rejection and pitch actuator usage. Furthermore, constraints are set to meet the pitch saturation limits and pitch rate constraints, ensuring robust and safe system performance. The constraints, are defined in Table 3-1, and include a pitching

rate limit of  $10^\circ/\text{s}$ , with pitch angle limits set between  $0^\circ$  and  $90^\circ$ . Furthermore, the rated rotor speed is regulated with weights applied to deviations from the 9.6 RPM reference, also specified in Table 3-1.

### Forgetting factor $\lambda$

The forgetting factors  $\lambda_{exp}$  of exponential forgetting and  $\lambda_{dir}$  of directional forgetting are essential parameters in adding adaptiveness to RLS. To prevent the covariance matrix to shrink unbounded during the covariance matrix update, the forgetting factors must be chosen such that  $0 < \lambda < 1$  for both methods.

In the case of exponential forgetting with  $\lambda_{exp}$  strictly less than unity, older data points are exponentially discounted, thereby allowing the estimator to adapt more effectively to recent changes. The effective memory length of the estimator for exponential forgetting is approximated by:

$$N_w = \frac{1}{1 - \lambda_{exp}}.$$

If this effective window is too short, the rapid discounting of past information can lead to loss of excitation. This results in an illconditioned covariance matrix with eigenvalues that blow up in the unexcited directions. Consequently, selecting the forgetting factor is a trade-off between the speed of adaptation to parameter changes and maintaining relevant information to maintain excitation. When the current Markov parameters do not adequately reflect shifts in current wind conditions, a lower forgetting factor enables the model to adapt faster to recent wind conditions. While the forgetting factor of exponential forgetting is applied to the entire covariance matrix, directional forgetting only discounts old data which lies in the direction of new incoming data. This selective forgetting helps preserve information in persistently unexcited directions, thereby maintaining numerical stability while still enabling adaptation in the directions where system dynamics are changing.

### Initial covariance $P$

The initial covariance  $P$  is typically chosen to balance the acceleration of parameter convergence during the initial phase of estimation with the level of confidence in the initial parameter estimates. This choice determines the flexibility in adjusting the Markov parameters during the RLS process, especially with no-forgetting [51].

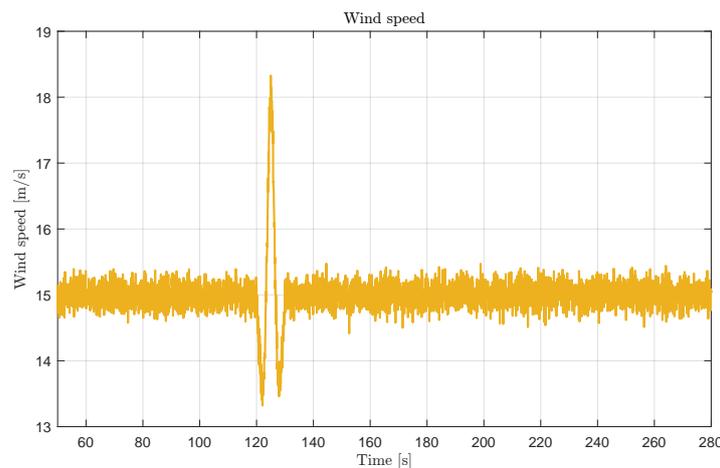
### 3-3 Wind cases

This section introduces the wind inflow scenarios used to evaluate the performance of the adaptive closed-loop SPC algorithm under different operating conditions. Three wind cases are defined: a Mexican hat wind gust (wind case I), a wind speed ramp (wind case II), and a turbulent wind field with LIDAR preview measurements (wind case III). These cases are designed to test the adaptive closed-loop SPC both with and without feedforward wind preview, for disturbance rejection capabilities and adaptability. The defined wind cases will be used in the following section to analyze the simulation results and compare the effect of different forgetting strategies in adaptive closed-loop SPC.

#### 3-3-1 Wind case I: Mexican hat wind gust

This subsection utilizes a wind profile consisting of a steady-state wind speed of 15 m/s, on which a Mexican hat-shaped gust is superimposed. This gust has a total duration of 10 s, and reaches a peak amplitude of 3 m/s. The gust is symmetric around its center and includes both an increase and decrease in wind speed, causing a disturbance that temporarily pushes the system out of its operating point. Moreover, ergodic zero-mean white noise with a standard deviation of 0.15 m/s is superimposed on the wind signal. This is an attempt to add excitation to the data which can be beneficial for the accuracy of the RLS Markov parameter estimations, especially during steady-state conditions where excitation levels are typically low.

The transition of the gust does not permanently move the system to a new operating point, but rather perturbs it briefly, making it a good candidate for analyzing the system's ability to reject disturbances and return to the original steady-state without fully moving to a new operating point. It is particularly valuable to evaluate the adaptive closed-loop SPC in scenarios where adaptation to a new operating region is not yet the goal. This wind case assumes perfect wind preview, meaning the controller has access to exact future wind speed values.

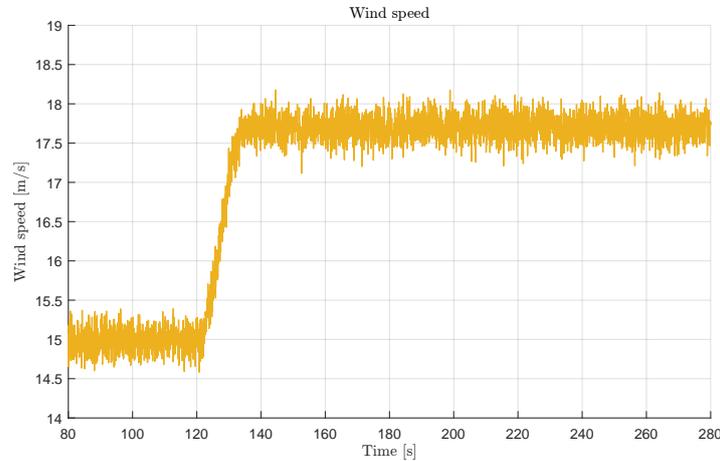


**Figure 3-2:** 15 m/s baseline wind speed profile with a Mexican hat gust starting at 120 seconds, amplitude of 3 m/s, and duration of 10 seconds. Ergodic zero-mean white noise with a standard deviation of 0.15 is superimposed for excitation.

### 3-3-2 Wind case II: Wind ramp

In this subsection, a wind speed ramp is introduced to test the adaptive closed-loop SPC under more pronounced changes in operating conditions. The wind profile consists of a steady inflow of 15 m/s onto which a ramp is superimposed with a duration of 20 s and a ramp rate of  $0.18 \text{ m/s}^2$ . The complete wind profile is shown in Figure 3-4. As in wind case I, ergodic zero-mean white noise with a standard deviation of 15 m/s is added to maintain excitation for the RLS estimator.

This case is particularly relevant for analyzing how the output predictor of the adaptive closed-loop SPC adapts when the system transitions to a new operating point. In contrast to wind case I, which focused on disturbance rejection near a fixed operating point, wind case II directly tests the ability of the controller to adapt its model in response to a persistent change in wind conditions. As in wind case I, perfect wind preview is assumed for this scenario.

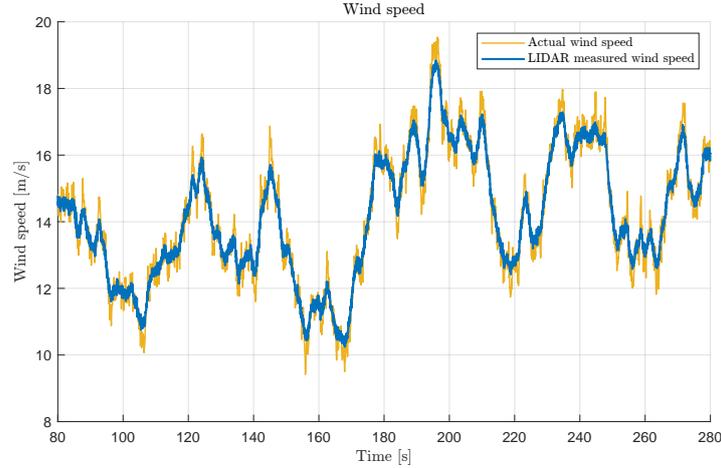


**Figure 3-3:** A 15 m/s baseline wind speed with a wind ramp, lasting for 20 s, and increasing at a rate of  $0.18 \text{ m/s}^2$ . Additionally ergodic zero-mean white noise is superimposed on the wind profile with a standard deviation of 0.15 m/s.

### 3-3-3 Wind case III: Turbulent wind

In this section, a turbulent wind field with an average wind speed of 15 m/s and a turbulence intensity of 16% is simulated. This wind field is treated as a wind preview in a more realistic way by applying a LIDAR simulator. For this, the ZephIR LIDAR described in subsection 3-1-2 is used with a focal distance  $d$  of 60 m. The effect of the LIDAR simulator on the actual wind speed is shown in Figure 3-11. Additionally, ergodic zero-mean white noise with a standard deviation of 0.1, m/s is added to the LIDAR measurements to simulate measurement noise.

The turbulent wind, due to its constantly fluctuating wind speed, provides variety in excitation directions that can benefit the parameter estimation, but also creates a challenge to the slow dynamics of the wind turbine. Moreover, this case tests the performance of the adaptive closed-loop SPC under imperfect preview conditions, where the disturbance feed-forward relies on LIDAR measurements. This allows evaluation of the controller's ability to reject disturbances, adapt its model to changing conditions, and maintain performance despite uncertainty in the wind preview.



**Figure 3-4:** Turbulent wind field with an average wind speed of 15 m/s, turbulence intensity of 16%, and superimposed ergodic zero-mean white noise with a standard deviation of 0.15 m/s. The corresponding LIDAR measured wind profile of the turbulent wind preview simulation is also shown.

### 3-4 Simulation results

This section presents simulation results that showcase the performance of the adaptive closed-loop SPC controller when applied to above-rated rotor speed regulation. The analysis begins by assessing the controller's sensitivity to tuning parameters and the impact of different forgetting strategies in the RLS estimator, which affect the adaptation of the output predictor. Reference tracking accuracy and control effort are both evaluated for the selection of suitable parameter values. With the selected parameters fixed, further simulations explore how the controller performs under varying wind conditions, including the integration of LIDAR-based wind preview information. These scenarios illustrate the controller's ability to adapt to changing dynamics and to anticipate wind disturbances.

#### 3-4-1 Sensitivity analysis

This subsection presents a sensitivity analysis to identify suitable tuned parameters for the adaptive closed-loop SPC controller. To evaluate the impact of these parameters, a performance metric based on the accumulated cost is defined. This cost is computed separately for the control input variation  $\Delta u$  and the output tracking error  $(y - r)$  as:

$$\begin{aligned} J_y &= \sum \left[ \Delta u_k^\top \Delta u_k \right] \\ J_u &= \sum \left[ (y_k - r_k)^\top (y_k - r_k) \right]. \end{aligned} \quad (3-2)$$

Here,  $J_y$  and  $J_u$  represent the accumulated costs for output tracking error and input variation, respectively. The output-related cost is selected as the primary performance metric since the control objective of this thesis is rated rotor speed reference tracking. This objective is directly captured by the accumulated output tracking error. The change in input is included

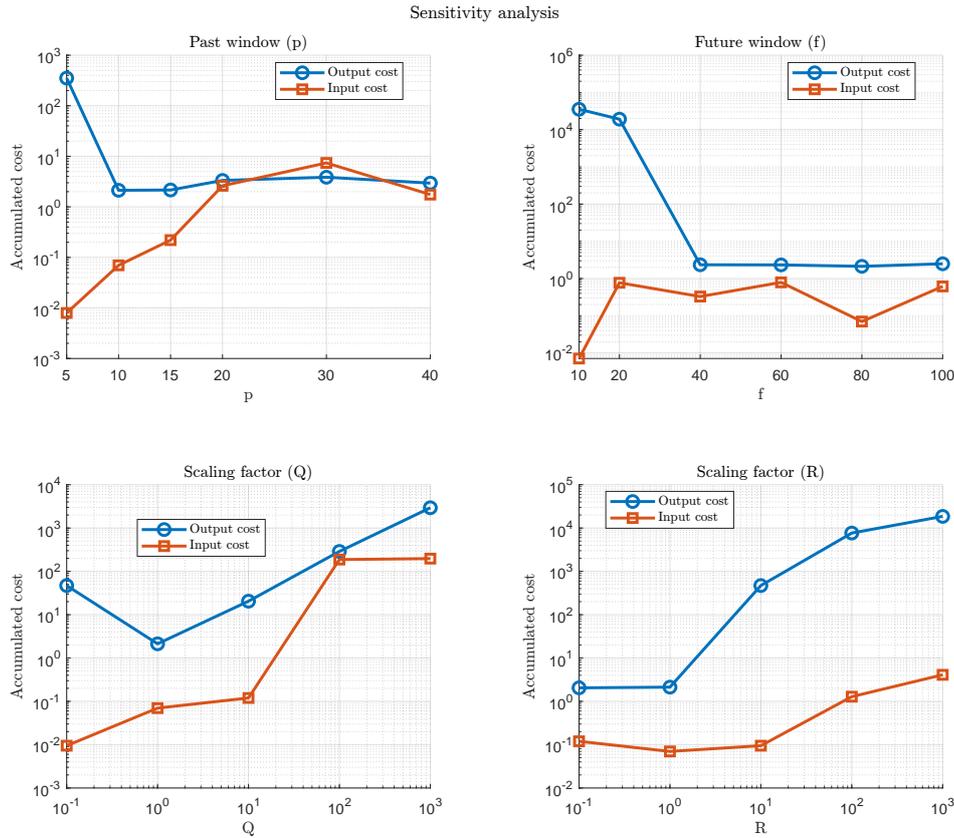
as a secondary metric to provide insight into the required control effort. While smoother pitch actuation can reduce actuator wear and dynamic loads, such load reduction is not explicitly targeted in the control design of this thesis.

Before presenting the sensitivity analysis, a few setup details should be clarified. The initial Markov parameters are estimated from open-loop data, as described in subsection 3-1-4, using a batch size of  $N = 1000$ . Slightly lower or higher values of this batch size did not affect the sensitivity analysis results. The pitch controller input is sampled with a period of  $T_s = 0.05$  s, matching the sampling time used in the QBlade simulation. The initial covariance matrix  $P_0$  is initialized as an identity matrix with appropriate dimensions, as this choice has been found to offer a good balance between adaptability and confidence in the initial estimate of the Markov parameters. Moreover, the simulation is allowed to reach a steady state during the first 80 seconds, after which the sensitivity analysis is conducted for the remainder of the simulation, up to a total duration of 280 seconds. To enhance excitation, small perturbations were introduced to the system by adding Gaussian noise with a standard deviation,  $\sigma = 0.01$ , to the pitch input.

The first sensitivity analysis addresses the impact of the controller parameters: the past window length  $p$ , future window length  $f$ , and the weight matrices  $Q$  and  $R$ . This analysis is performed under the wind conditions described for wind case I in subsection 3-3-1, with the wind profile being available as a feedforward preview and the gust starting at  $t = 120$  s. This simulation attempts to capture both steady-state behavior and disturbance rejection of the Mexican hat wind gust. The tuning of the past window length  $p$ , future window length  $f$ , and weight matrices  $Q$  and  $R$  was performed iteratively. This yielded the values  $p = 10$ ,  $f = 80$ ,  $Q = 1$ , and  $R = 1$  to obtain the best performance by balancing tracking accuracy and control activity. The optimal values were determined through a sensitivity analysis, where each parameter was varied independently while fixing the others at their previously identified optimal values. As shown in Figure 3-5, this analysis confirms that these settings minimize the cumulative cost of input and output.

The past window length  $p = 10$  is expected to give the best performance due to the influence of  $p$  on the uniqueness of the output predictor based on Assumption 4 and reducing the effect of the unknown initial state bias based on Assumption 2. The worse results for  $p > 10$  can possibly be explained by the presence of past values in the regressor vector that reflect outdated dynamics during the wind gust. The moderate value of  $p$  also prevents overfitting to noise and avoids a slow convergence rate in the parameter estimation process of RLS. The choice of a future window length  $f = 80$  seems to capture the slow dynamics of the wind turbine while providing enough foresight to anticipate and reject the effect of the wind gust. A longer  $f$  could increase sensitivity in performance to model inaccuracies in the output predictor. Moreover, the selected values  $Q = 1$  and  $R = 1$  provide a balanced trade-off between the competing objectives of reference tracking and control effort. If  $Q$  is set larger relative to  $R$ , the controller places too much emphasis on tracking the output, potentially causing abrupt pitch changes and higher control activity, this effect becomes evident for  $Q = 10^2$  and larger in Figure 3-5. Furthermore, increasing  $R$  may slow down the system's response and leads to higher tracking errors, which becomes noticeable starting from  $R = 10$  in Figure 3-5.

For the tuning of the forgetting factors, wind case II from subsection 3-3-2, with the ramp starting at  $t = 120$  s, is used to assess whether adaptivity through forgetting in RLS improves



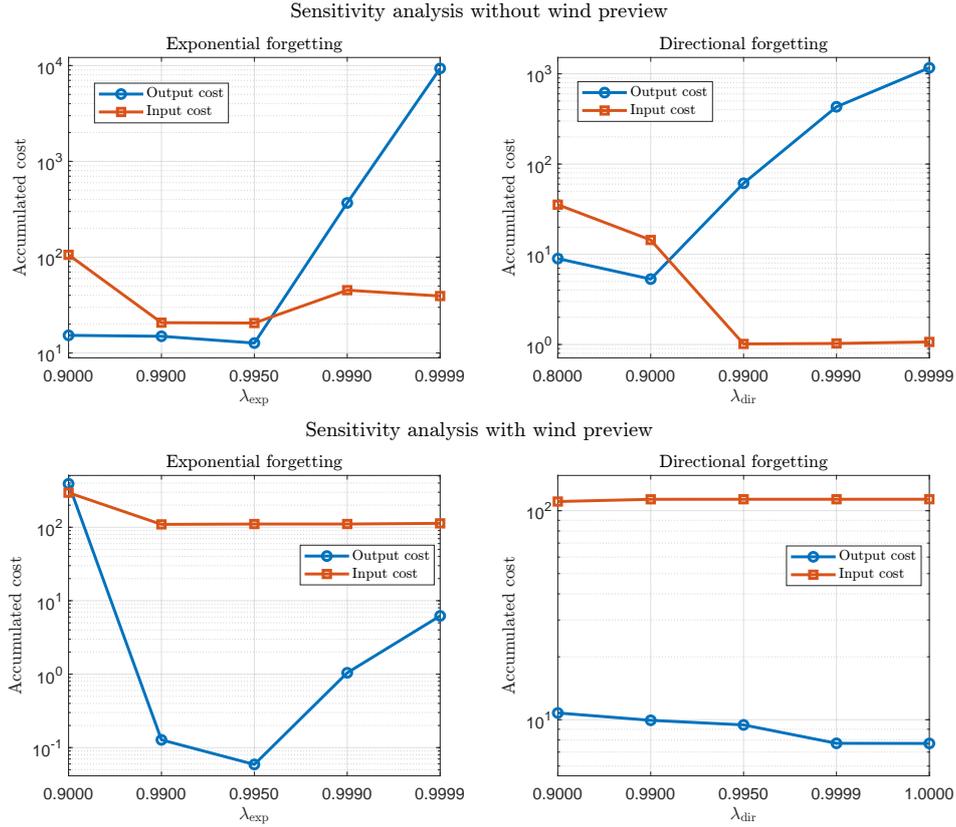
**Figure 3-5:** Sensitivity analysis, with wind case I, of the past window  $p$ , future window  $f$  and the weights  $Q$  and  $R$  with wind preview.

performance when the system must transition to a new operating point due to a wind ramp. In this analysis, the past window length  $p = 10$ , future window length  $f = 80$ , and weight matrices  $Q = 1$  and  $R = 1$  are fixed to the previously identified optimal values. The sensitivity analysis focuses on varying the forgetting factors for both exponential ( $\lambda_{exp}$ ) and directional ( $\lambda_{dir}$ ) forgetting. These were tuned iteratively under two scenarios: without wind preview and with perfect wind preview. The corresponding performance results are presented in Figure 3-6.

Without wind preview, the best performance in terms of combined accumulated input and output cost was achieved using exponential forgetting with  $\lambda_{exp} = 0.995$ . For directional forgetting, the lowest cost was obtained with  $\lambda_{dir} = 0.9$ . In exponential forgetting, the inverse of the forgetting factor is applied to the prior covariance matrix  $P_k$  to obtain the preliminary covariance matrix  $\bar{P}_k$ , inflating all directions equally regardless of recent excitation. This uniform inflation of uncertainty maintains its adaptability. In contrast, directional forgetting inflates the covariance matrix selectively, only in the subspace excited by the most recent regressor. This targeted inflation avoids unnecessary forgetting in unexcited directions by only modifying the covariance matrix in the direction of the most recently excited regressor. As a result, when forgetting is applied, a lower forgetting factor such as  $\lambda_{dir} = 0.9$  is needed to introduce sufficient uncertainty.

With wind preview available, exponential forgetting again yielded the lowest cost at  $\lambda_{exp} =$

0.995. This suggests that this forgetting factor seems to maintain the relevant approximate data memory window  $N_w$  of Equation 2-19, short enough for adaptability and long enough to maintain enough excitation. Notably, the performance of directional forgetting improved as  $\lambda_{dir} \rightarrow 1$ , indicating that it offered no clear advantage in this scenario. As shown in Figure 3-6, reducing the forgetting factor below unity did not yield performance gains. A possible reason is that directional forgetting inflates the parameter covariance in directions already well-explained by the measured disturbance, leading to unnecessary adaptation and no significant improvement in parameter estimation accuracy.



**Figure 3-6:** Sensitivity analysis of the exponential forgetting factor  $\lambda_{exp}$  and directional forgetting factor  $\lambda_{dir}$  with and without wind preview of wind case II.

For the sensitivity analysis of the forgetting factors under wind case III, characterized by turbulent wind conditions with LIDAR preview, the same iterative tuning procedure was followed. Once again, exponential forgetting with  $\lambda_{exp} = 0.995$  yielded the lowest combined cost in terms of accumulated input and output performance. Similar to the results observed in wind case II with preview, directional forgetting did not offer any performance improvement over the case with no-forgetting. For this scenario, and similar to the earlier assumption that the presence of a turbulent wind preview already provides sufficient information for the estimator, the effectiveness of selectively inflating uncertainty through directional forgetting is reduced.

A summary of the selected controller parameters, including the tuned forgetting factors for both with wind preview and without wind preview scenarios, is provided in Table 3-3.

**Table 3-3:** Selected controller parameters used for the adaptive closed-loop SPC algorithm. Including sampling time, window lengths, and weights. It also lists the excitation signal properties on the inputs and the tuned forgetting factors for both scenarios: with (w/) and without (w/o) wind preview.

Parameter	Value
Sampling time, $T_s$	0.05 s
Initialization time to reach steady state, $t_{ss}$	80 s
Past window length, $p$	10
Future window length, $f$	80
Output weight, $Q$	1
Input weight, $R$	1
Initial covariance matrix, $P_0$	$I$
Superimposed excitation signal on pitch input	Gaussian noise, $\sigma = 0.01$
Exponential forgetting factor, $\lambda_{exp}$ (w/o wind preview)	0.995
Directional forgetting factor, $\lambda_{dir}$ (w/o wind preview)	0.9
Exponential forgetting factor, $\lambda_{exp}$ (w/ wind preview)	0.995
Directional forgetting factor, $\lambda_{dir}$ (w/ wind preview)	1 (no-forgetting)

### 3-4-2 Wind case I without controller, excitation and wind preview

To investigate the behavior of the parameter adaptation with RLS under minimal excitation, a scenario is considered where the wind turbine operates at steady-state conditions with a constant input pitch angle of  $u = 10.45$ , and the closed-loop SPC controller is disabled. The Gaussian white noise excitation on the pitch input, as described in Table 3-3, is switched off after 80 seconds. Throughout the remainder of the simulation, no further input excitation is introduced, and wind preview is also disabled. For clarity, recall that the RLS gain is defined as  $\phi_{k+1}^T P_{k+1}$ , which is a crucial part of the Markov parameter update in Equation 2-18. Moreover, wind case I is used, where a wind gust occurs at  $t = 800$  seconds. However, since wind preview is not available in this scenario, the gust is treated by the estimator as an unknown disturbance. This setup allows for an analysis of how a period without excitation influences the evolution of the Markov parameters and their sensitivity to prediction errors caused by the varying wind speed of the gust from the different forgetting methods.

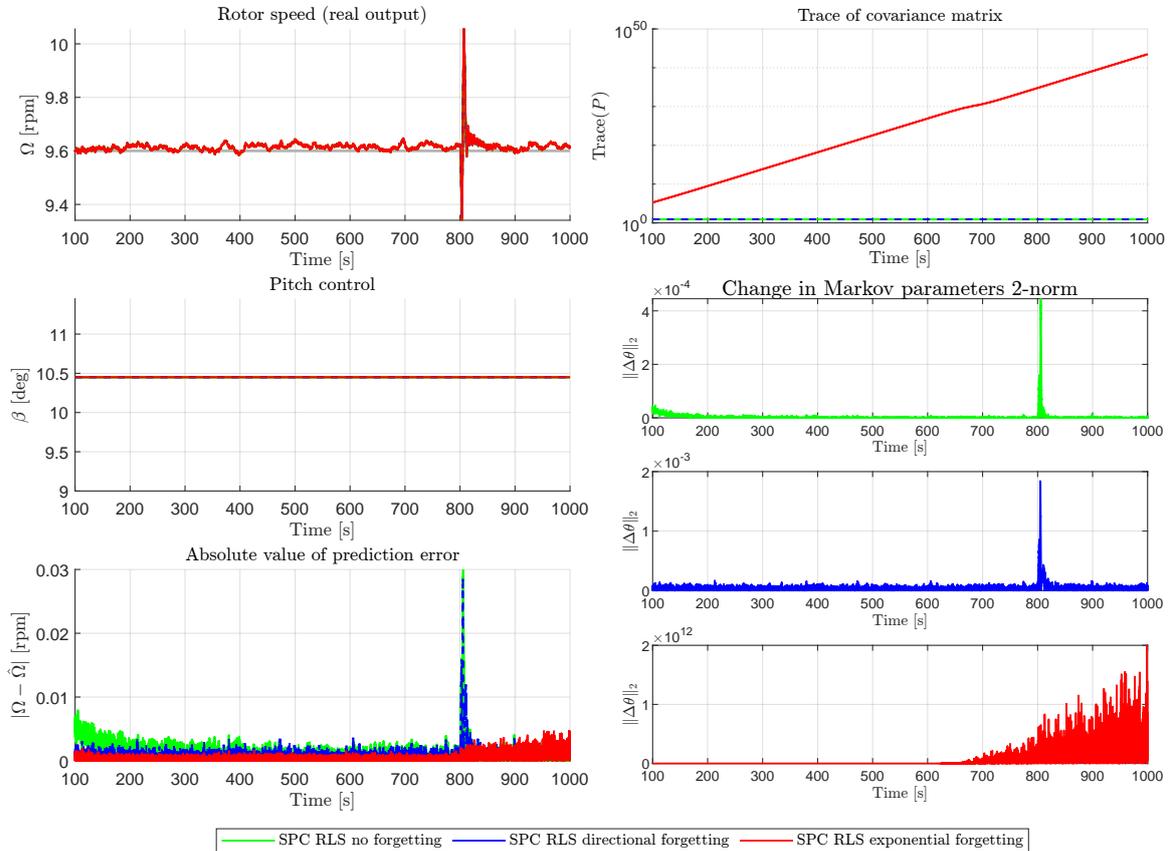
In the case of exponential forgetting, the trace of the covariance matrix increases significantly over time, as shown in Figure 3-8. The forgetting factor  $\lambda_{exp} = 0.995$  is applied uniformly to all directions of the prior covariance matrix. As a result of the lack of persistent input excitation, many directions of the covariance matrix remain poorly excited and accumulate large eigenvalues, which generally results in high parameter uncertainty. Due to the inflated covariance, even a slight change in the excitation direction can lead to a large RLS gain, resulting in a significant Markov parameter update, as visible in Figure 3-8. Apart from the wind gust at  $t = 800$  s, it is already visible that small prediction errors during steady-state conditions lead to large Markov parameter updates, reaching magnitudes on the order of  $10^{12}$ . This highlights the estimator's extreme sensitivity with the lack of input excitation.

In contrast, the no-forgetting strategy results in a more bounded covariance matrix as visible in Figure 3-7. This is due to the continued shrinking of the covariance in the initially excited directions, leading the estimator to become increasingly confident in its parameter estimates

over the steady-state period. Consequently, when the gust causes a relatively large prediction error, the regressor gives a small RLS gain due to the high confidence captured in the covariance matrix, resulting in only small parameter updates.

Directional forgetting applies forgetting selectively in the directions of new input excitation. At steady state, these directions continue to be forgotten, which prevents overconfidence. As a result, the covariance matrix retains more uncertainty compared to the no-forgetting case. When the gust appears, the resulting prediction error leads to parameter updates that are higher than in the no-forgetting case, but less aggressive than the exponential forgetting case. This results in a more balanced adaptation under low-excitation conditions.

While the true system parameters are unknown and, therefore, a direct validation of the estimated Markov parameters is not possible, the simulation results still showcase some characteristic behavior of each forgetting strategy in the absence of excitation and with the disabled controller.



**Figure 3-7:** Comparison of simulation results using RLS under different forgetting strategies without wind preview and control, using wind case I. The excitation described in Table 3-3 superimposed on the input is removed after  $t_{ss} = 80$  s. The left column illustrates the rotor speed, pitch control and absolute value of prediction error of the rotor speed ( $|\Omega - \hat{\Omega}|$ ). The right column presents the trace of the covariance matrix ( $\text{Trace}(P)$ , log-scale) and the change in Markov parameters ( $\|\Delta\theta\|_2$ ). Three distinct simulations are compared: SPC RLS without forgetting, with directional forgetting ( $\lambda_{dir} = 0.9$ ), and with exponential forgetting ( $\lambda_{exp} = 0.995$ ). The bottom-right subplot ( $\|\Delta\theta\|_2$ ) stacks the three plots, each corresponding to one forgetting method, highlighting the evolution of parameter adaptation over time.

### 3-4-3 Wind case II without wind preview

In this scenario, the adaptive closed-loop SPC simulations are analyzed with the wind ramp used in wind case II without wind preview. In this case, the system is driven to a new operating point by applying the wind ramp of the earlier described wind case II with the ramp starting at  $t = 120$  s.

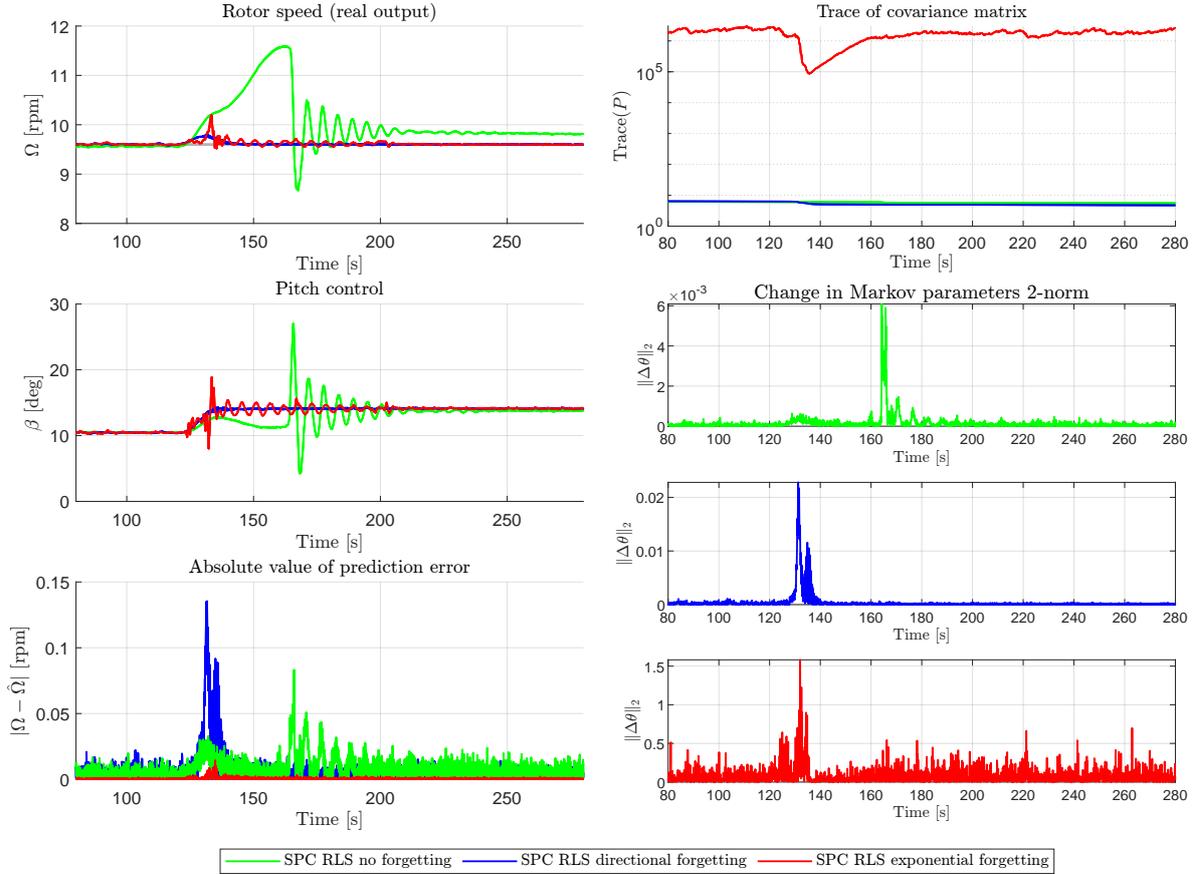
As shown in Figure 3-8, the SPC algorithm without any forgetting does not fully reject the disturbance and the rotor speed settles around 9.8 rpm, showing a steady-state tracking error. It is also visible that the trace of the covariance matrix remains bounded for no-forgetting. However, what is not visible in the figure is that in the absence of forgetting, the eigenvalues of the covariance matrix tend to converge to zero over time, particularly during the steady-state phase before the ramp. This makes the covariance matrix increasingly confident, especially in the directions corresponding to the steady-state operating point. This results in small updates in the estimated Markov parameters when new dynamics are introduced. Looking again at Figure 3-8, when the rotor speed increases sharply, passing the rotor speed of 11 rpm, the controller reacts with a large pitch input to compensate. The input reaches the maximum pitch angle of  $27^\circ$ , and both the rotor speed and pitch subsequently oscillate before settling. As soon as this oscillation starts, it induces a significant spike in the prediction error and a noticeable update in the Markov parameters. This suggests that despite the confident covariance matrix, the algorithm still shows some degree of adaptiveness in response to prediction errors, together with the additional excitation from the oscillating pitch input. This behavior can be explained by the RLS gain increasing as the regressor excites directions in the covariance matrix that were previously less excited. Consequently, the matrix multiplication  $\phi_{k+1}^\top P_{k+1}$  produces larger values and, together with the rising prediction error, the Markov parameter updates increase. As a result, the model shows more adaptiveness to the new operating point than at the beginning of the wind ramp. Although the rotor speed and pitch input eventually settle, the Markov parameters after the oscillations and increasing prediction errors still lack sufficient adaptiveness and accuracy to make the controller fully return to the rated rotor speed reference.

When exponential forgetting is employed, the covariance matrix inflates to a significantly high covariance trace compared to no-forgetting and directional forgetting. This causes the RLS estimator to maintain higher RLS gains. The inflated uncertainty allows the algorithm to adapt quickly to recent data but also leads to covariance windup, where poorly excited directions in the covariance matrix accumulate large eigenvalues. This windup effect on the covariance matrix is visible in Figure 3-8 with a large trace of order  $10^6$ . As shown in Figure 3-8, the arrival of the wind disturbance ramp causes a sudden increase in the prediction error, resulting in the estimator making large parameter updates in an effort to improve the accuracy of the estimated Markov parameters. These abrupt updates modify the identified relationship between the pitch input and rotor speed. Consequently, the closed-loop SPC generates fluctuating rotor speed predictions over time, resulting in a non-smooth control input trajectory. This results in oscillations in the input, as seen in Figure 3-8, and introduces additional excitation, which temporarily reduces the trace of the covariance matrix. At one point, the parameters seem to be more accurately adapted to the new operating point, as can be seen in the performance of the rated rotor speed tracking and pitch control activity. Even as the response settles, the parameters continue to update at a noticeable rate due to exponential forgetting, which persistently discounts older data and inflates the covariance

matrix sufficiently to maintain sensitivity to new samples. Furthermore, while looking at the trace of exponential forgetting, it is visible that after a certain time, the increase of the trace becomes more constant around the order of  $10^6$ . This behavior can be understood from the covariance update in Equation 2-21. Here, it is visible that the first term  $\frac{1}{\lambda_{\text{exp}}}P_k$  increases the covariance trace. However, this increase in uncertainty also leads to a larger product of  $P_k\phi_{k+1}$ , even when the excitation does not increase. Eventually, the increase in the covariance trace due to  $\frac{1}{\lambda_{\text{exp}}}P_k$  gets balanced with the subtraction in Equation 2-21 related to the corrections of the new data with a larger product of  $P_k\phi_{k+1}$ .

In contrast with exponential forgetting, directional forgetting limits its forgetting to just the direction excited by the new data, sparing directions that are not driven by the disturbance. By not forgetting all prior information exponentially over time, it prevents large Markov parameter updates and a possibly smoother transition to the new operating point. This is also visible at the trace of the covariance matrix in Figure 3-8, which is significantly lower than exponential forgetting and around the same as with no-forgetting. The difference with no-forgetting is that directional forgetting avoids shrinking the uncertainty for excited directions, thereby avoiding the singular values in these direction to go to zero. As a result, it has a higher level of uncertainty in these directions, maintaining the adaptability. For the controller, directional forgetting provides a middle ground. The controller can still adjust to unmodeled disturbances or slow changes by adding uncertainty to the covariance matrix with forgetting, but it does so in a more controlled manner that avoids the violent swings in parameter updates that occur with exponential forgetting. Furthermore, the covariance matrix's trace remains bounded, so the Markov parameters remain more constant. The Markov parameter changes shown in Figure 3-8 stay significantly below exponential forgetting, but are larger than without forgetting. Once the system settles after the wind ramp, the parameter updates remain closer to zero, demonstrating that directional forgetting balances stability and adaptiveness without the aggressive Markov parameter changing characteristic of exponential forgetting. As a result, the transitions in control inputs and rotor speed remain relatively smooth and stable even during adaptation. This is visible in Figure 3-8, where after the wind ramp the rotor speed deviates briefly but then returns steadily toward the reference with only a small oscillation in the pitch input. This shows that the closed-loop SPC with directional forgetting can respond to disturbances in this case without introducing instability in the form of oscillations.

To quantify performance, the metric defined in Equation 3-2 is used again, which evaluates both the accumulated reference tracking error and the accumulated change in control inputs. Without explicitly computing this metric, Figure 3-8 already visually suggests that the no-forgetting case performs the worst. To qualitatively compare exponential and directional forgetting, the accumulated cost metrics are evaluated. Directional forgetting yields an accumulated input cost of 5.00 and an output cost of 13.9, whereas exponential forgetting results in a higher input cost of 13.4 and output cost of 22.3. These results confirm that directional forgetting achieves the best performance in this case among the three strategies, offering a favorable trade-off between tracking accuracy and control effort.



**Figure 3-8:** Comparison of simulation results using closed-loop SPC with RLS under different forgetting strategies without wind preview, using wind case II. The left column illustrates the rotor speed, pitch control and absolute value of prediction error of the rotor speed ( $|\Omega - \hat{\Omega}|$ ). The right column presents the trace of the covariance matrix ( $\text{Trace}(P)$ , log-scale) and the change in Markov parameters ( $\|\Delta\theta\|_2$ ). Three distinct simulations are compared: SPC RLS without forgetting, with directional forgetting ( $\lambda_{dir} = 0.9$ ), and with exponential forgetting ( $\lambda_{exp} = 0.995$ ). The bottom-right subplot ( $\|\Delta\theta\|_2$ ) stacks the three plots, each corresponding to one forgetting method, highlighting the evolution of parameter adaptation over time.

### 3-4-4 Wind case II with wind preview

In this subsection, the wind preview is available. Here, the adaptive closed-loop SPC incorporates the anticipated wind ramp directly into its feedforward structure, transforming the disturbance from an unknown input into a known one. Again using the wind case II where the wind ramp starts at  $t = 120$  s.

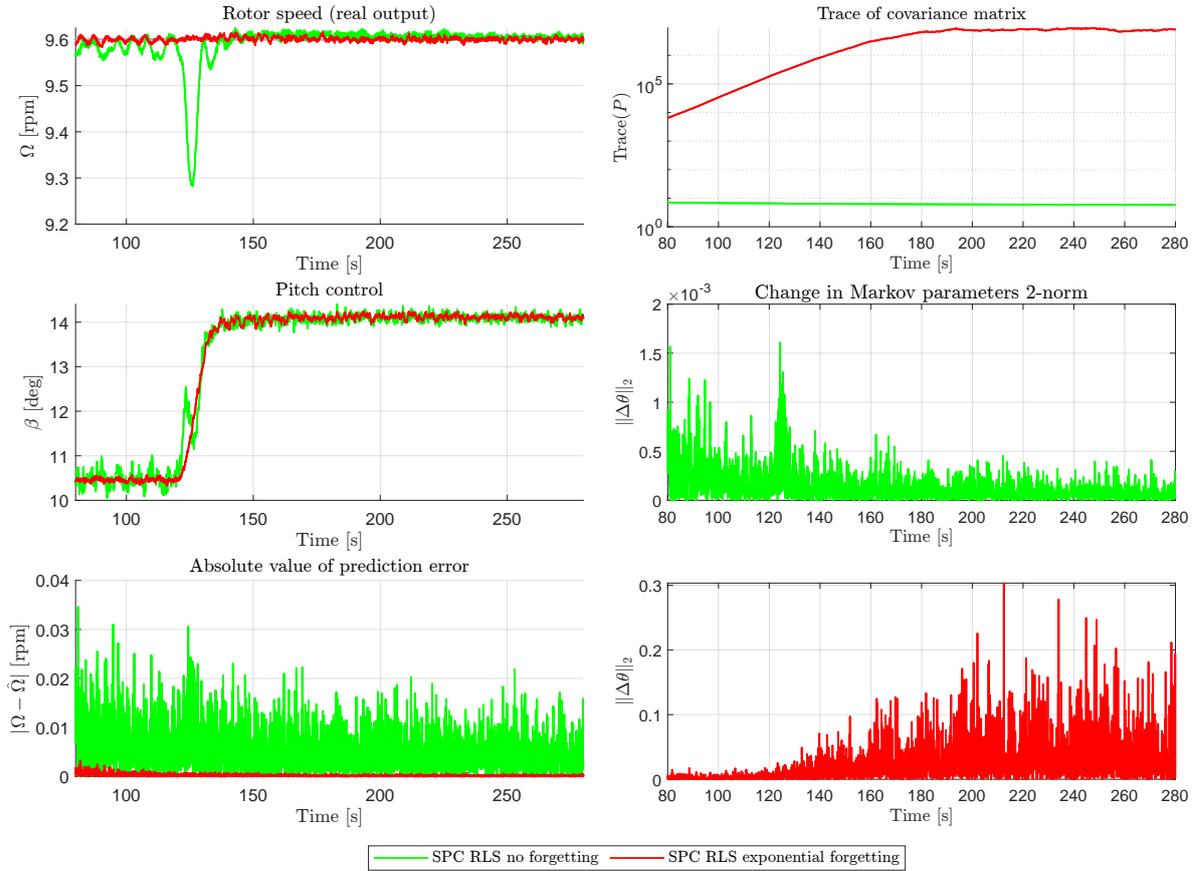
Looking at the no-forgetting case in Figure 3-9, it is visible that the trace of the covariance matrix remains on the order of  $10^0$ , and the changes in the estimated Markov parameters show a negligible peak when the wind speed ramp begins. This limited change arises because, prior to the ramp, the no-forgetting RLS has again already become confident in certain directions, leaving less adaptiveness for further adjustment during the disturbance. Consequently, the Markov parameters remain nearly unchanged, as visible in Figure 3-9. As the future wind ramp enters the prediction horizon, the controller, relying on a somewhat mismatched output

predictor, responds with an excessively high pitch to compensate for the wind increase, causing a temporary dip in rotor speed. Nonetheless, despite this brief tracking error and the lack of parameter update, the no-forgetting RLS closed-loop SPC still maintains reasonable control performance. This suggests that incorporating wind preview into the output predictions can partially compensate for the model's near-static nature, enabling a partial disturbance rejection.

When wind preview is available, exponential forgetting proves beneficial, as shown by the near-perfect rotor speed reference tracking in Figure 3-9. Exponential forgetting continuously discounts older data while emphasizing the most recent wind conditions, which are more relevant to the current operating scenario. Because the forgetting is applied in all directions, the parameter estimates adapt more rapidly. As demonstrated in Figure 3-9, this adaptation allows the controller to generate smoother control signals, tightly regulating the rotor speed during the ramp in the wind speed. Furthermore, Figure 3-9 shows that the trace of the covariance matrix remains significantly high on the order of  $10^8$ . However, unlike the previous example where a high covariance trace led to overcorrection of the Markov parameters, this issue does not arise when wind speed preview information is incorporated.

Directional forgetting did not appear to provide performance improvement over no-forgetting as discussed in the sensitivity analysis and is, therefore, left out of the results.

To quantitatively compare no-forgetting and exponential forgetting, the accumulated cost metrics are evaluated again. No-forgetting yields an accumulated output cost of 6.24 and an input cost of 7.83, whereas exponential forgetting results in a significantly lower output cost of 1.93 and input cost of 0.0872. These results quantitatively confirm that exponential forgetting achieves the best performance in this case.



**Figure 3-9:** Comparison of simulation results using closed-loop SPC with RLS under different forgetting strategies with perfect wind preview, using wind case II. The left column illustrates the rotor speed, pitch control and absolute value of prediction error of the rotor speed ( $|\Omega - \hat{\Omega}|$ ). The right column presents the trace of the covariance matrix ( $\text{Trace}(P)$ , log-scale) and the change in Markov parameters ( $\|\Delta\theta\|_2$ ). Two simulations are compared: SPC RLS without forgetting and with exponential forgetting ( $\lambda_{exp} = 0.995$ ). The bottom-right subplot ( $\|\Delta\theta\|_2$ ) stacks the two plots, each corresponding to one forgetting method, highlighting the evolution of parameter adaptation over time.

### 3-4-5 Wind case II with and without wind preview

In this example, wind case II with the ramp starting at  $t = 120$  s is used. Only this time at  $t = 150$  s, the wind speed preview information is lost and set to zero. When the wind preview is available, exponential forgetting showed to be beneficial in the previous example with improved rotor-speed tracking performance. This exponential forgetting approach will be compared to a combined method that simultaneously applies both directional and exponential forgetting. The motivation for this stems from the observation made earlier in this section that directional forgetting can be advantageous when no wind preview is available. Intuitively, merging exponential and directional forgetting may improve performance when the wind preview is initially present but later disappears.

The RLS algorithm incorporating both directional and exponential forgetting first applies directional forgetting, followed by exponential forgetting within the same time step. Referring

to the preliminary covariance modifications introduced for the individual forgetting methods, this combined approach can be described as follows. First, the preliminary update for directional forgetting is applied, as defined in Equation 2-29:

$$\bar{P}_k = P_k + \alpha_k \phi_k \phi_k^\top,$$

followed by the preliminary update for exponential forgetting, as defined in Equation 2-20:

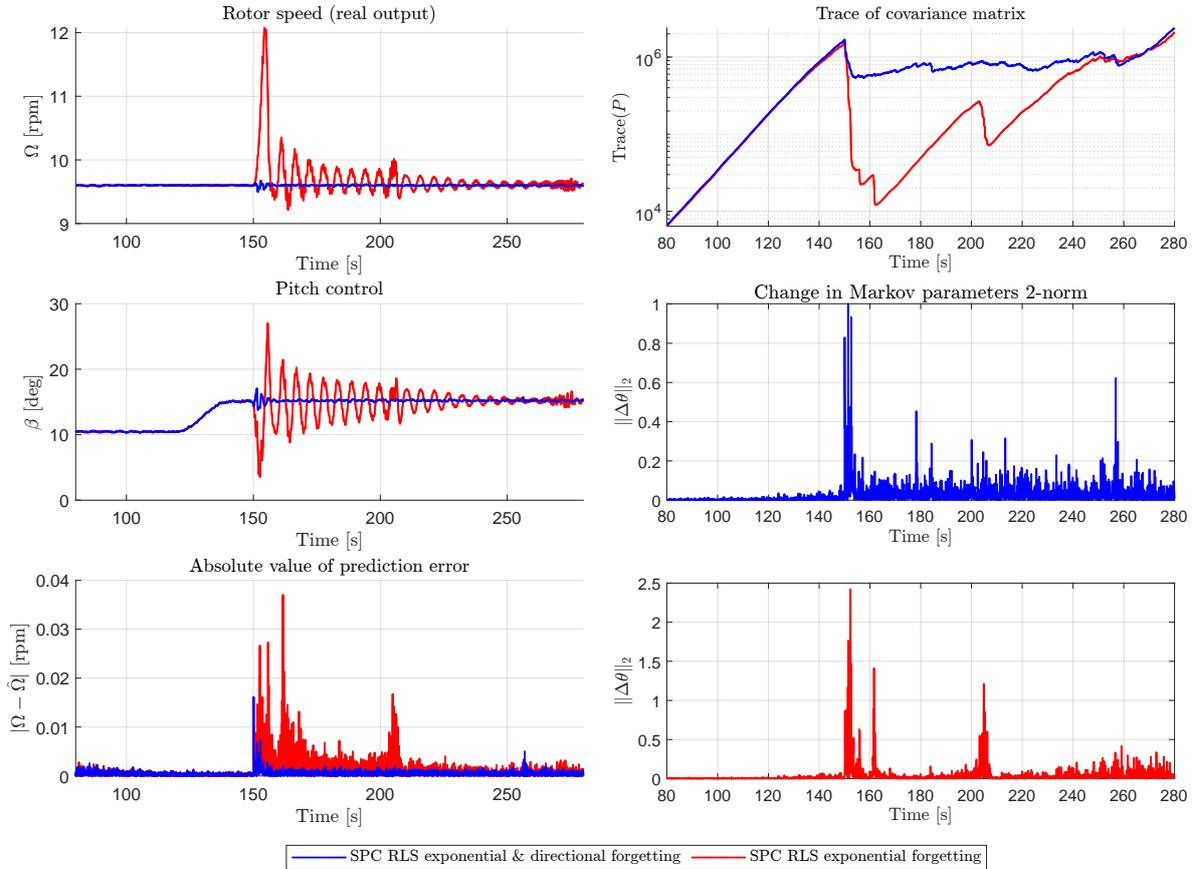
$$\bar{P}_k = \frac{1}{\lambda_{exp}} P_k.$$

For this combined directional and exponential forgetting algorithm, the directional forgetting factor  $\lambda_{dir} = 0.9$  is used, which was determined as optimal from the sensitivity analysis with wind case II without wind preview. Moreover, the exponential forgetting factor is set to  $\lambda_{exp} = 0.995$ , which was found to be optimal in the sensitivity analysis with wind case II for both cases, with and without wind preview.

Looking at Figure 3-10, it is visible that with only exponential forgetting, the wind-speed ramp initially produces good reference tracking. However, once the wind preview disappears at 150 seconds, the purely exponential forgetting method shows significant oscillatory behavior. In Figure 3-10 a large peak in the change of Markov parameter can be noted as the algorithm attempts to compensate for the increase in prediction error after losing the wind preview information when set to zero. To reject the disturbance within the pitch-to-rotor speed relationship, exponential forgetting once again over-corrects the Markov parameters with the the controller increasing the pitch input up to its maximum value of  $27^\circ$ . The resulting oscillatory behavior in both the input and output introduces additional excitation, which temporarily decreases the trace of the covariance matrix, as shown in Figure 3-10. Although the oscillations eventually decrease, convergence to the reference remains slow.

When the adaptive closed-loop SPC with both exponential and directional forgetting is used, Figure 3-10 shows that the performance during the period with wind-speed preview remains similar to that of exponential forgetting alone. As indicated by the previous example, where directional forgetting on its own did not improve rotor-speed tracking, exponential forgetting suggests to add the most relevant adaptation to this part of the simulation. Once the wind preview disappears, the benefits of directional forgetting seem to become more evident. The algorithm selectively identifies the direction of excitation, and it more effectively learns the relationship between the pitch input and the rotor-speed output in the absence of accurate wind speed information. It seems that, under the changing conditions in this example, the combined forgetting method captures the best aspects of both approaches. Also visible in Figure 3-10 is that the change in parameter, after losing the wind preview, is more moderate as with only exponential forgetting. Furthermore, the trace of the covariance matrix does not seem to suddenly decrease as much as with exponential forgetting. This can be caused by the less exciting input trajectory compared to the oscillating input with only exponential forgetting.

For a quantitative performance comparison, the accumulated costs of the performance metric obviously confirms the better performance of the combined forgetting method. The input and output costs for the combined forgetting method are 0.349 and 5.89, respectively, while the exponential forgetting method yields significantly higher costs of 311 for the input and 99.8 for the output.



**Figure 3-10:** Comparison of simulation results using closed-loop SPC with RLS under different forgetting strategies with loss of wind preview at  $t = 120s$ , using wind case II. The left column illustrates the rotor speed, pitch control and absolute value of prediction error of the rotor speed ( $|\Omega - \hat{\Omega}|$ ). The right column presents the trace of the covariance matrix ( $\text{Trace}(P)$ , log-scale) and the change in Markov parameters ( $\|\Delta\theta\|_2$ ). Two simulations are compared: SPC RLS with exponential forgetting ( $\lambda_{exp} = 0.995$ ), and with exponential and directional forgetting combined ( $\lambda_{exp} = 0.995$  and  $\lambda_{dir} = 0.9$ ). The bottom-right subplot ( $\|\Delta\theta\|_2$ ) stacks the two plots, each corresponding to one forgetting method, highlighting the evolution of parameter adaptation over time.

### 3-4-6 Wind case III with LIDAR preview

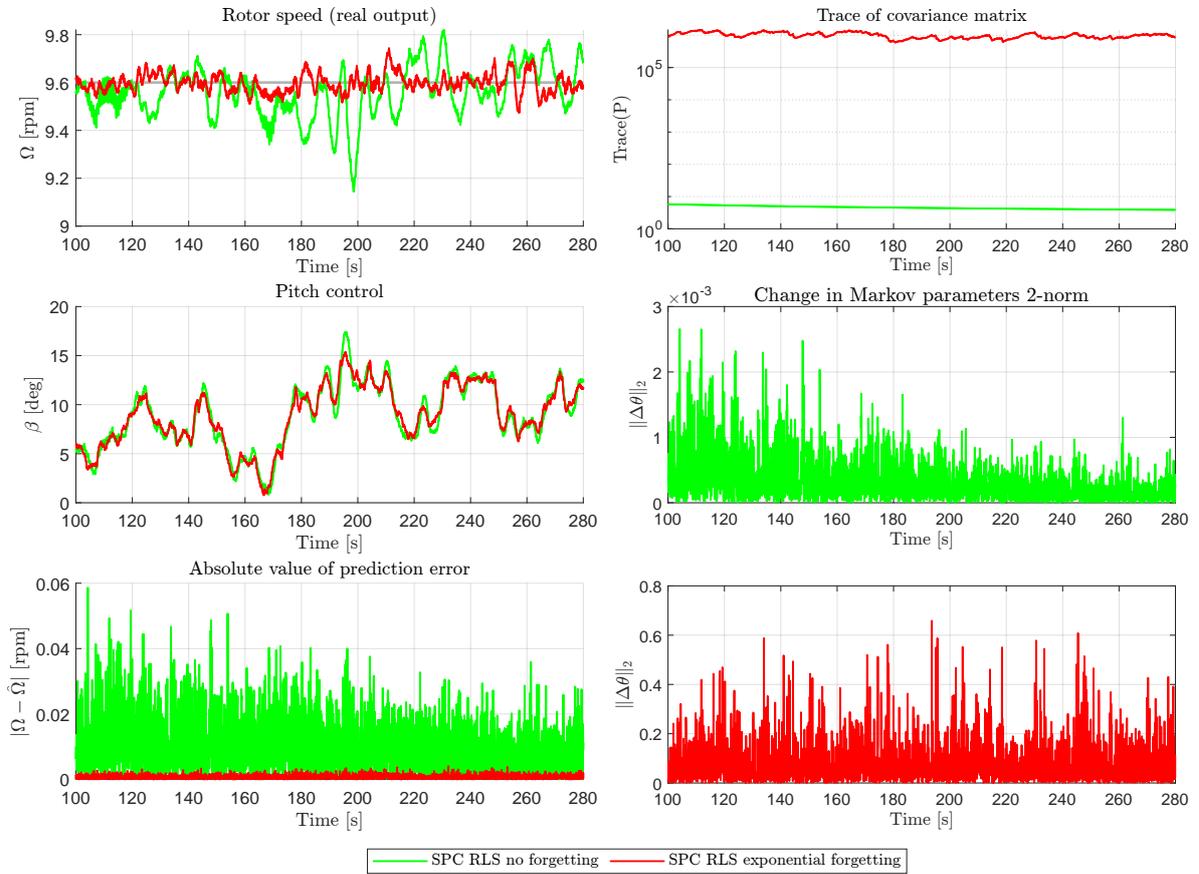
In this example, the turbulent wind case III with LIDAR wind preview measurements is used. Figure 3-11 shows that the method without forgetting struggles to track the reference rotor speed during the LIDAR measured turbulence wind speed simulation, using wind case III. While good performance has been demonstrated with more stable wind speeds, the fluctuating nature of turbulent wind causes the relatively static Markov parameters, even with available wind speed information, to struggle in consistently tracking the rotor speed with the closed-loop SPC. The relatively high prediction errors in Figure 3-11 suggest that the control input may be based on flawed predictions, leading to suboptimal results.

Exponential forgetting delivers the best performance, as visible in Figure 3-11, by capturing short term variations in dynamics more effectively and reducing prediction errors. Even

though the wind preview obtained from the LIDAR simulator contains measurement errors compared to the true wind speed, the adaptive closed-loop SPC still tracks the reference rotor speed tightly under exponential forgetting.

Once again, directional forgetting did not appear to provide performance improvement over no-forgetting with LIDAR wind preview, as mentioned in the sensitivity analysis, and is therefore left out of the results.

According to the performance metric, the no-forgetting case resulted in an input cost of 46.3 and an output cost of 44.8, while exponential forgetting achieved a significantly lower input cost of 5.53 and an output cost of 35.2. This reflects a substantial better performance for exponential forgetting in both control effort and tracking accuracy under turbulent wind speed conditions with LIDAR preview.



**Figure 3-11:** Comparison of simulation results using closed-loop SPC with RLS under different forgetting strategies with LIDAR wind preview, using wind case III. The left column illustrates the rotor speed, pitch control and absolute value of prediction error of the rotor speed ( $|\Omega - \hat{\Omega}|$ ). The right column presents the trace of the covariance matrix ( $\text{Trace}(P)$ , log-scale) and the change in Markov parameters ( $\|\Delta\theta\|_2$ ). Two simulations are compared: SPC RLS without forgetting and with exponential forgetting ( $\lambda_{exp} = 0.995$ ). The bottom-right subplot ( $\|\Delta\theta\|_2$ ) stacks the two plots, each corresponding to one forgetting method, highlighting the evolution of parameter adaptation over time.

### 3-5 Concluding remarks on the simulations

Exponential forgetting continuously discounts older data, which maintains adaptability but sometimes leads to significant parameter swings and overshoot for example in the case where wind preview is unavailable. However, it has shown to be very effective in scenarios where wind preview is available, as it is able to quickly adapt parameters to recent wind conditions. The adaptive closed-loop SPC with no-forgetting and without wind preview during the wind ramp case performed the worst among all tested scenarios. In this configuration, the absence of wind speed information and the lack of parameter adaptation significantly limited the controller's ability to maintain rated rotor speed tracking during and after the wind ramp. Nonetheless, when wind speed information was incorporated through feedforward control, a significant improvement in performance was observed. The addition of wind preview enabled the controller to better anticipate future disturbances, partially compensating for the lack of model adaptation.

Directional forgetting inflates covariance only in directions excited by new information, providing more controlled adaptation. It outperforms no- and exponential forgetting in the case where no wind preview is available. Although, its advantages disappear when explicit wind disturbance modeling is available, leading to results similar to no-forgetting. A combined approach, using both exponential and directional forgetting, achieved swift adaptation with wind preview and maintained performance when preview became unavailable, offering robust performance across diverse wind scenarios. During turbulent wind conditions, exponential forgetting again showed better performance by rapidly tracking frequent wind speed changes. In this case no-forgetting also tracks the rated rotor speed effectively but remains less responsive when observing changes in the Markov parameters. This reduced responsiveness may explain its lower performance compared to exponential forgetting.

As shown in Table 3-4, the accumulated input and output costs for each forgetting strategy across the evaluated wind scenarios show how the different adaptive closed-loop SPC configurations perform relative to each other. These cost metrics quantify how well the controller tracks the rated rotor speed ( $J_y$ ) while minimizing control effort ( $J_u$ ). Notably, the adaptive closed-loop SPC with exponential forgetting and wind preview consistently outperforms the other forgetting strategies across all evaluated wind scenarios. This performance gain is attributed to its ability to rapidly adapt to recent wind conditions while compensating for the slow rotor dynamics using wind preview information.

**Table 3-4:** Accumulated input and output costs sorted by wind case and forgetting strategy.

Wind case	Forgetting strategy	Output cost $J_y$	Input cost $J_u$
II (no preview)	No-forgetting	30.7	$1.53 \cdot 10^3$
II (no preview)	Directional	5.00	13.9
II (no preview)	Exponential	13.4	22.3
II (with preview)	No-forgetting	6.24	7.83
II (with preview)	Exponential	1.93	0.0872
II (preview loss)	Exponential	99.8	$3.11 \cdot 10^2$
II (preview loss)	Exponential + directional	5.89	0.349
III (LIDAR preview)	No-forgetting	44.8	46.3
III (LIDAR preview)	Exponential	35.2	5.53



## Conclusion

### 4-1 General conclusion

Large-scale wind turbines encounter significant control challenges, caused by stronger nonlinear dynamics and increased rotor inertia. To address these challenges, an adaptive feedforward closed-loop Subspace Predictive Control (SPC) controller is proposed to achieve optimal rotor speed regulation in the above-rated region. It is designed to manage the nonlinear dynamics by continuously updating the model, while also utilizing wind preview measurements to counteract the phase loss imposed by a large rotor inertia. In line with the aforementioned challenges, the research question addressed in this thesis is:

***How can closed-loop SPC incorporate both an adaptive filter and Light Detection and Ranging (LIDAR)-based wind preview information to achieve optimal above-rated rotor speed regulation of a wind turbine?***

To clarify how each aspect of the research question can be answered, a set of subquestions were formulated. To conclude the corresponding findings of this thesis, the subquestions will be answered in this section:

- *How can an adaptive closed-loop SPC framework incorporate previewed future wind speed information from LIDAR measurements?*

The closed-loop SPC framework builds upon the Linear Time-Invariant (LTI) state-space formulation in [26], which does not include external disturbances. To incorporate previewed wind speed information, this formulation is extended by treating the wind field as a known disturbance input. Based on this augmented state-space model, an output data equation is derived over a finite horizon, yielding a linear output predictor that depends on input, disturbance, and output data. This output predictor is integrated into a receding-horizon optimization function, where the disturbance is given by the feedforward wind preview to improve rotor speed regulation and counteract the phase loss imposed by the inertia, following [37].

This thesis builds on the work of [37] by incorporating a simulated Continuous-Wave (CW) LIDAR, modeled based on the commercial ZephIR LIDAR system [33], which

estimates wind via Doppler shifts over a focused range. This introduces spatial averaging that filters high-frequency wind content. However, at the distances used in this thesis, the averaging remains acceptable and provides a realistic wind preview for use in closed-loop SPC.

- *How can standard Recursive Least Squares (RLS), exponential forgetting, and directional forgetting be used for feedforward closed-loop SPC to adapt to nonlinear rotor dynamics?*

To make closed-loop SPC adaptive, it is reformulated in Chapter 2 using a RLS structure, in which the output predictor parameters are continuously updated based on new data, using a covariance matrix to weigh recent observations and adjust parameter estimates accordingly. In standard RLS without forgetting, adaptability is limited because all data is weighed equally, reducing the algorithm's response to new data over time. Exponential forgetting reduces the influence of old data by applying a forgetting factor uniformly to the covariance matrix. While this enhances adaptability to time-varying dynamics, it may lead to an accumulation of uncertainty in persistently unexcited directions due to the absence of informative data. When these directions are eventually excited, the associated high covariance can result in large parameter updates, potentially leading to inaccurate parameter estimates, as there is insufficient recent informative data in those directions. Directional forgetting applies the forgetting factor only along the direction of new data, preserving certainty in unexcited directions and preventing covariance inflation when new data is insufficiently informative. However, because it preserves past information in unexcited directions, directional forgetting may adapt more slowly to changes compared to exponential forgetting. Thus, while directional forgetting mitigates overconfidence issues, exponential forgetting is generally more adaptive in its parameter estimation due to its uniform forgetting of all parameter directions. The impact of each method on wind turbine simulations is further explored in the next subquestion using QBlade simulations for rated rotor speed tracking.

- *How do the developed adaptive closed-loop SPC with wind preview algorithms perform in above rated rotor speed regulation when applied to a nonlinear wind turbine simulation?*

In Chapter 3, the simulations are performed with the high fidelity QBlade simulator on a 10 MW DTU wind turbine using the adaptive closed-loop SPC with wind preview feedforward to counteract phase loss and forgetting methods for parameter adaptation. The control performance is evaluated based on two accumulated costs, the rotor speed tracking error and the change in pitch.

To start, the developed methods were tested by conducting a sensitivity analysis on the intended wind cases to make a selection of the control parameters. Based on this analysis, the past and future window lengths were selected, which are used to construct the output predictor and regressor required for closed-loop SPC. The selected past window likely led to improved performance by having a large enough window to reach a unique output predictor and reduce the impact of the unknown initial state. The selected future window appears to offer a good compromise between anticipating the turbine's slow dynamic response and avoiding excessive sensitivity to model mismatches in the output predictor. Additionally, this sensitivity analysis determined the tuning of the controller weights  $Q$  and  $R$ , which define the trade-off between rotor speed regulation and pitch actuator effort. It also set the forgetting factors used in the adaptive RLS

algorithm.

Controller performance and adaptivity to the Markov parameters was then evaluated using the selected control parameters. This was done by comparing the results from steering the wind turbine to a new operating point under a wind ramp and turbulent wind field, using no-forgetting, exponential forgetting, and directional forgetting with closed-loop SPC. During the tuning of the forgetting factors, it was observed that directional forgetting offered no performance improvement over no-forgetting when wind preview was available, across all evaluated wind scenarios. Consequently, it was excluded from the experimental results involving wind preview. Directional forgetting possibly misallocates added uncertainty on the covariance matrix in directions already well-explained by the measured disturbance, thereby offering no significant advantage.

In the following experimental results, the forgetting strategies were evaluated on the wind ramp case without wind preview. No-forgetting resulted in noticeable oscillations, which can be attributed to the increased confidence reflected in the covariance matrix, limiting the controller's ability to quickly adapt to changing conditions. In the same case, exponential forgetting tracked the reference more effectively, but also with significant pitch activity and rotor speed oscillations. During the wind ramp, new data excites directions that previously lacked excitation and, therefore, have inflated covariance. As a result, the incoming data causes large and potentially inaccurate parameter updates in those directions. At one point, these parameter estimations seem to become more accurate, looking at the performance, as the controller input reintroduces excitation with the oscillations. Directional forgetting was able to track the reference with less aggressive parameter updates and smoother rotor speed and pitch transitions. This is likely due to its selectively added uncertainty in excited directions, allowing for moderate adaptation without excessive parameter updates observed with exponential forgetting.

With wind preview enabled, exponential forgetting was able to track the rotor speed almost perfectly under the same wind ramp, with its ability of prioritizing recent wind measurements in its parameter estimation. This performance can be attributed to the parameter updates being more responsive to the current wind conditions, with forgetting applied uniformly across all directions. For the same wind case with wind preview, the no-forgetting scheme resulted in a temporary dip in rotor speed during the ramp, likely due to the limited adaptability of the estimator, which had become overconfident in its prior parameter estimates. Despite this dip, the no-forgetting scenario still showed significantly better performance than without the wind preview.

In the wind ramp scenario where the wind preview was suddenly removed, the combined use of directional and exponential forgetting demonstrated smooth transitions. In contrast, exponential forgetting alone exhibited large oscillations at the point of losing preview. This is consistent with earlier observations that exponential forgetting can overreact in its parameter updates with the absence of wind preview. In contrast, directional forgetting provides more moderate adaptation and smoother controller performance under unmodeled disturbances, which seems to hold even when combined with exponential forgetting.

Lastly, for turbulent wind with the ZephIR LIDAR simulated wind preview, exponential forgetting again surpassed no-forgetting in performance. By continuously adapting to

short-term wind fluctuations, exponential forgetting maintained better performance, whereas no-forgetting remained more rigid in its parameter updates, resulting in reduced tracking performance.

As can be seen in the performance overview in Table 3-4, exponential forgetting combined with wind preview delivered the best performance across the tested wind scenarios by enabling rapid adaptation to changing wind conditions and effectively counteracting the phase loss imposed by a large rotor inertia, outperforming all other strategies.

These findings directly address the research question of how previewed future wind speed information from LIDAR measurements can be incorporated into the adaptive closed-loop SPC framework. The results demonstrate that such preview information can be effectively utilized by integrating it with RLS forgetting methods. In particular, exponential forgetting in combination with wind preview was found to deliver the best overall performance across the applied wind scenarios investigated during this research. The wind preview enables the controller to anticipate incoming wind disturbances, while the exponential forgetting manages rapid adaptation to changing wind conditions, resulting in effective disturbance rejection.

## 4-2 Future work and recommendations

This thesis presented an implementation of an adaptive feedforward closed-loop SPC concept within a QBlade simulation environment. The approach demonstrated encouraging above-rated rotor speed tracking results under varying wind conditions. Nonetheless, additional exploration and validation are necessary to further evaluate and extend these findings. Below are the main suggestions for future research:

- *Investigate RLS forgetting strategies under linearized condition.*  
In this thesis, the estimated parameters from the investigated RLS forgetting strategies showed some notable results such as directional forgetting not adding any performance advantaged over no-forgetting with wind preview. To better understand how accurately these forgetting strategies capture the true wind turbine dynamics, future research could isolate the system's complexity by testing on a simplified, linear wind turbine model. Such a setting enables more direct, theoretically based comparisons of the estimated parameters against ground truth dynamics.
- *Explore multi-objective formulations.*  
While rotor speed regulation was the primary focus, large-scale turbines also require fatigue load reduction, blade load balancing, and tower damping. These objectives can be integrated into the closed-loop SPC formulation via additional cost terms or constraints. Individual Pitch Control (IPC), which modulates each blade's pitch independently, offers an effective means of reducing asymmetric blade loads and can be incorporated as additional inputs in the predictive model. Weighted multi-objective tuning can further balance rotor speed regulation with structural load mitigation, enhancing the life span of the wind turbine.
- *Extend adaptive filtering strategies to floating wind turbines.*  
The onshore turbine experiments conducted in this research provide a valuable foundation for evaluating adaptive closed-loop SPC performance. Building on this, floating

wind turbines present an exciting opportunity for further exploration due to their more complex dynamic behavior. With additional degrees of freedom and more pronounced nonlinearities, they represent an attractive case for evaluating the effectiveness of the adaptive filters in closed-loop SPC in handling these complex dynamics.

- *Examine alternate adaptive filters and forgetting mechanisms.*  
While exponential and directional forgetting RLS showed good performance in capturing local nonlinear behavior, each approach presents trade-offs and is limited by the use of a constant forgetting factor. Future work could investigate different forgetting methods, such as this adaptive forgetting scheme [63], where the forgetting factor dynamically adjusts based on the prediction error.



---

# Bibliography

- [1] International Energy Agency. Energy supply statistics data browser, 2023. url: <https://www.iea.org/data-and-statistics/data-tools/energy-statistics-data-browser>.
- [2] International Energy Agency. Executive Summary – Empowering Urban Energy Transitions, June 2023. url: <https://www.iea.org/reports/empowering-urban-energy-transitions/executive-summary>.
- [3] Jerry L. Holechek, Hatim M. E. Geli, Mohammed N. Sawalhah, and Raul Valdez. A Global Assessment: Can Renewable Energy Replace Fossil Fuels by 2050? *Sustainability*, 14(8):4792, January 2022.
- [4] Ørste. Hornsea 2 offshore wind farm, 2022. url: <https://orsted.co.uk/energy-solutions/offshore-wind/our-wind-farms/hornsea2>.
- [5] Joyce Lee, Feng Zhao, Ben Backwell, Emerson Clarke, and Rebecca Williams. GWEC global wind energy report 2022. Technical report, GWEC, 2022.
- [6] Ryan Wiser, Joseph Rand, Joachim Seel, Philipp Beiter, Erin Baker, Eric Lantz, and Patrick Gilman. Expert elicitation survey predicts 37% to 49% declines in wind energy costs by 2050. *Nature Energy*, 6(5):555–565, May 2021. Publisher: Nature Research.
- [7] Matt Shields, Philipp Beiter, Jake Nunemaker, Aubryn Cooperman, and Patrick Duffy. Impacts of turbine and plant upsizing on the levelized cost of energy for offshore wind. *Applied Energy*, 298:117189, September 2021.
- [8] Henk Polinder, S.W.H. Haan, Maxime Dubois, and Johannes Slootweg. Basic Operation Principles and Electrical Conversion Systems of Wind Turbines. *EPE Journal (European Power Electronics and Drives Journal)*, 15:43–50, October 2005.
- [9] Navdeep Singh, Bhanu Pratap, and Akhilesh Swarup. Robust control design of variable speed wind turbine using quantitative feedback theory. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 232(6):691–705, September 2018. Publisher: IMECHE.

- [10] H Polinder and J Morren. Developments in wind turbine generator systems: 8th International conference on modeling and simulation of electric machines, converters and systems, Hammamet, Tunisia. *ELECTRIMACS 2005, Proceedings 8th International conference on modeling and simulation of electric machines, converters and systems*, pages 1–11, 2005.
- [11] Lucy Y. Pao and Kathryn E. Johnson. Control of Wind Turbines. *IEEE Control Systems Magazine*, 31(2):44–62, April 2011.
- [12] Kyung-Hyun Kim, Tan Luong Van, Dong-Choon Lee, Seung-Ho Song, and Eel-Hwan Kim. Maximum Output Power Tracking Control in Variable-Speed Wind Turbine Systems Considering Rotor Inertial Power. *IEEE Transactions on Industrial Electronics*, 60(8):3207–3217, August 2013.
- [13] Peter Jamieson and Laurence Morgan. Trends, prospects and R&D directions in wind turbine technology. In *Comprehensive Renewable Energy*, pages 817–853. Elsevier, 2022.
- [14] Michael Borg and Maurizio Collu. Offshore floating vertical axis wind turbines, dynamics modelling state of the art. Part III: Hydrodynamics and coupled modelling approaches. *Renewable and Sustainable Energy Reviews*, 46:296–310, June 2015.
- [15] Shuai Xiao, Geng Yang, and Hua Geng. Individual pitch control design of wind turbines for load reduction using. In *2013 IEEE ECCE Asia Downunder*, pages 227–232, June 2013.
- [16] E. A. Bossanyi. Individual Blade Pitch Control for Load Reduction. *Wind Energy*, 6(2):119–128, 2003.
- [17] Manfred Morari and Jay H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, May 1999.
- [18] James Blake Rawlings, David Q. Mayne, and Moritz Diehl. *Model predictive control: theory, computation, and design*. Nob Hill Publishing, Madison, Wisconsin, 2nd edition edition, 2017.
- [19] E. F. Camacho and C. Bordons. *Model Predictive control*. Advanced Textbooks in Control and Signal Processing. Springer London, London, 2007.
- [20] Sergio Grammatico. Model Predictive Control (SC42125). Lecture slides, Delft University of Technology, Delft, Netherlands, 2022 url: <https://sites.google.com/site/grammaticosergio/teaching/mpc>.
- [21] Yanhua Liu, Ron J. Patton, and Shuo Shi. Wind Turbine Load Mitigation Using MPC with Gaussian Wind Speed Prediction. In *2018 UKACC 12th International Conference on Control (CONTROL)*, pages 32–37, September 2018.
- [22] G. a. M. van Kuik, J. Peinke, R. Nijssen, D. Lekou, J. Mann, and J. N. Sørensen. Long-term research challenges in wind energy – a research agenda by the European Academy of Wind Energy. *Wind Energy Science*, 1(1):1–39, February 2016.
- [23] Zhong-Sheng Hou and Zhuo Wang. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235:3–35, June 2013.

- 
- [24] Xiuxing Yin and Xiaowei Zhao. Data driven learning model predictive control of offshore wind farms. *International Journal of Electrical Power & Energy Systems*, 127:106639, May 2021.
- [25] Florian Dörfler, Jeremy Coulson, and Ivan Markovskiy. Bridging Direct and Indirect Data-Driven Control Formulations via Regularizations and Relaxations. *IEEE Transactions on Automatic Control*, 68(2):883–897, February 2023.
- [26] Wouter Favoreel, Bart De Moor, and Michel Gevers. SPC: Subspace Predictive Control. *IFAC Proceedings Volumes*, 32(2):4004–4009, July 1999.
- [27] Jeremy Coulson, John Lygeros, and Florian Dorfler. Data-Enabled Predictive Control: In the Shallows of the DeePC. In *2019 18th European Control Conference (ECC)*, pages 307–312, Naples, Italy, June 2019. IEEE.
- [28] Felix Fiedler and Sergio Lucia. On the relationship between data-enabled predictive control and subspace predictive control. In *2021 European Control Conference (ECC)*, pages 222–229, June 2021.
- [29] Jianfei Dong, Michel Verhaegen, and Edward Holweg. Closed-loop Subspace Predictive Control for Fault Tolerant MPC Design. *IFAC Proceedings Volumes*, 41(2):3216–3221, January 2008.
- [30] Antoine Borraccino, David Schlipf, Florian Haizmann, and Rozenn Wagner. Wind field reconstruction from nacelle-mounted lidar short-range measurements. *Wind Energy Science*, 2(1):269–283, May 2017.
- [31] T. Mikkelsen. Lidar-based Research and Innovation at DTU Wind Energy – a Review. *Journal of Physics: Conference Series*, 524(1):012007, June 2014.
- [32] J Allen Zak. Atmospheric Boundary Layer Sensors for Application in a Wake Vortex Advisory System. 2003.
- [33] Eric Simley, Lucy Pao, Rod Frehlich, Bonnie Jonkman, and Neil Kelley. Analysis of Wind Speed Measurements using Continuous Wave LIDAR for Wind Turbine Control. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Orlando, Florida, January 2011. American Institute of Aeronautics and Astronautics.
- [34] N. W. Verwaal, G. J. van der Veen, and J. W. van Wingerden. Predictive control of an experimental wind turbine using preview wind speed measurements. *Wind Energy*, 18(3):385–398, 2015.
- [35] Arne Koerber and Rudibert King. Combined Feedback–Feedforward Control of Wind Turbines Using State-Constrained Model Predictive Control. *IEEE Transactions on Control Systems Technology*, 21(4):1117–1128, July 2013. Conference Name: IEEE Transactions on Control Systems Technology.
- [36] Martin D. Spencer, Karl A. Stol, Charles P. Unsworth, John E. Cater, and Stuart E. Norris. Model predictive control of a wind turbine using short-term wind field predictions. *Wind Energy*, 16(3):417–434, 2013.

- [37] Rogier Dinkla, Tom Oomen, Jan-Willem Van Wingerden, and Sebastiaan P Mulders. Data-driven LIDAR feedforward predictive control. 2023.
- [38] Yingzhao Lian and Colin N. Jones. Nonlinear Data-Enabled Prediction and Control. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, pages 523–534. PMLR, May 2021. ISSN: 2640-3498.
- [39] Gijs van der Veen, Jan-Willem van Wingerden, and Michel Verhaegen. Global Identification of Wind Turbines Using a Hammerstein Identification Method. *IEEE Transactions on Control Systems Technology*, 21(4):1471–1478, July 2013. Conference Name: IEEE Transactions on Control Systems Technology.
- [40] Linbin Huang, John Lygeros, and Florian Dörfler. Robust and Kernelized Data-Enabled Predictive Control for Nonlinear Systems. *IEEE Transactions on Control Systems Technology*, 32(2):611–624, March 2024.
- [41] Hassan A. Kingravi, Harshal Maske, and Girish Chowdhary. Kernel controllers: A systems-theoretic approach for data-driven modeling and control of spatiotemporally evolving processes. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 7365–7370, December 2015.
- [42] Jan-Willem van. Wingerden. *Control of wind turbines with 'smart' rotors : proof of concept and LPV subspace identification*. PhD thesis, 2008. ISBN: 9789090235837 Publisher: [s.n.].
- [43] J. W. van Wingerden, I. Houtzager, F. Felici, and M. Verhaegen. Closed-loop LPV identification of the time-varying dynamics of a variable-speed wind turbine. *IFAC Proceedings Volumes*, 41(2):8866–8871, January 2008.
- [44] Jan-Willem van Wingerden and Michel Verhaegen. Subspace identification of Bilinear and LPV systems for open- and closed-loop data. *Automatica*, 45(2):372–381, February 2009.
- [45] Ivo Houtzager, Jan-Willem van Wingerden, and Michel Verhaegen. Recursive Predictor-Based Subspace Identification With Application to the Real-Time Closed-Loop Tracking of Flutter. *IEEE Transactions on Control Systems Technology*, 20(4):934–949, July 2012.
- [46] Liyu Cao and Howard Schwartz. Directional forgetting algorithm based on the decomposition of the information matrix. *Automatica*, 36:1725–1731, November 2000.
- [47] G. J. Van der Veen. *Identification of wind energy systems*. PhD thesis, 2013.
- [48] D. Marten. *QBlade: A Modern Tool for the Aeroelastic Simulation of Wind Turbines vorgelegt von*. PhD thesis, TU Berlin, 2019.
- [49] Brian D. O. Anderson and John B. Moore. *Optimal Filtering*. Courier Corporation, 1979. Google-Books-ID: iYMqLQp49UMC.
- [50] Alessandro Chiuso. The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 43(6):1034–1048, June 2007.

- 
- [51] Ljung Lennart. *System Identification: Theory for the User*. Upper Saddle River, NJ, January 1999.
- [52] Michel Verhaegen and Vincent Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, Cambridge, 2007.
- [53] I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Fast-array Recursive Closed-loop Subspace Model Identification. *IFAC Proceedings Volumes*, 42(10):96–101, January 2009.
- [54] Rogier Dinkla, Sebastiaan P. Mulders, Jan-Willem van Wingerden, and Tom Oomen. Closed-loop Aspects of Data-Enabled Predictive Control. *IFAC-PapersOnLine*, 56(2):1388–1393, January 2023.
- [55] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. JHU Press, October 1996.
- [56] S. Chandrasekaran and Ali H. Sayed. Stabilizing the Generalized Schur Algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4):950–983, October 1996. Publisher: Society for Industrial and Applied Mathematics.
- [57] Ali Sayed and Thomas Kailath. Recursive Least-Squares Adaptive Filters. In *Digital Signal Processing Fundamentals*, volume 20094251, pages 1–40. CRC Press, November 2009.
- [58] H. Glauert. Airplane Propellers. In William Frederick Durand, editor, *Aerodynamic Theory: A General Review of Progress Under a Grant of the Guggenheim Fund for the Promotion of Aeronautics*, pages 169–360. Springer, Berlin, Heidelberg, 1935.
- [59] Christian Bak, Robert Bitsche, Anders Yde, Taeseong Kim, Morten H Hansen, Frederik Zahle, Mac Gaunaa, José Blasques, Mads Døssing, Jens-Jakob Wedel Heinen, and Tim Behrens. Light Rotor: The 10-MW reference wind turbine. *Proceedings of EWEA 2012 - European Wind Energy Conference & Exhibition, 2012*.
- [60] Christian Bak. The DTU 10-MW Reference Wind Turbine. *DTU Library*, 2013.
- [61] C. M. Sonnenschein and F. A. Horrigan. Signal-to-Noise Relationships for Coaxial Systems that Heterodyne Backscatter from the Atmosphere. *Applied Optics*, 10(7):1600–1604, July 1971.
- [62] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, March 2019.
- [63] Xiangdong Sun, Jingrun Ji, Biying Ren, Chenxue Xie, and Dan Yan. Adaptive Forgetting Factor Recursive Least Square Algorithm for Online Identification of Equivalent Circuit Model Parameters of a Lithium-Ion Battery. *Energies*, 12(12):2242, January 2019.



---

# Glossary

## List of Acronyms

<b>DTU</b>	Technical University of Denmark
<b>GWEC</b>	Global Wind Energy Council
<b>LCOE</b>	Levelized Cost of Energy
<b>IPC</b>	Individual Pitch Control
<b>CPC</b>	Collective Pitch Control
<b>LIDAR</b>	Light Detection and Ranging
<b>CW</b>	Continuous-Wave
<b>PW</b>	Pulsed-Wave
<b>MPC</b>	Model Predictive Control
<b>DDC</b>	Data-Driven Control
<b>DDPC</b>	Data-Driven Predictive Control
<b>MIMO</b>	Multiple Input Multiple Output
<b>DeePC</b>	Data-enabled Predictive Control
<b>SPC</b>	Subspace Predictive Control
<b>LTI</b>	Linear Time-Invariant
<b>RKHS</b>	Reproducing Kernel Hilbert Space
<b>LPV</b>	Linear Parameter-Varying
<b>RLS</b>	Recursive Least Squares
<b>LS</b>	Least Squares
<b>BEM</b>	Blade Element Momentum
<b>UBEM</b>	Unsteady Blade Element Momentum
<b>DLL</b>	Dynamic Link Library
<b>SIL</b>	Software in Loop
<b>RMSE</b>	Root Mean Squared Error
<b>MBC</b>	Multi-Blade Coordinate