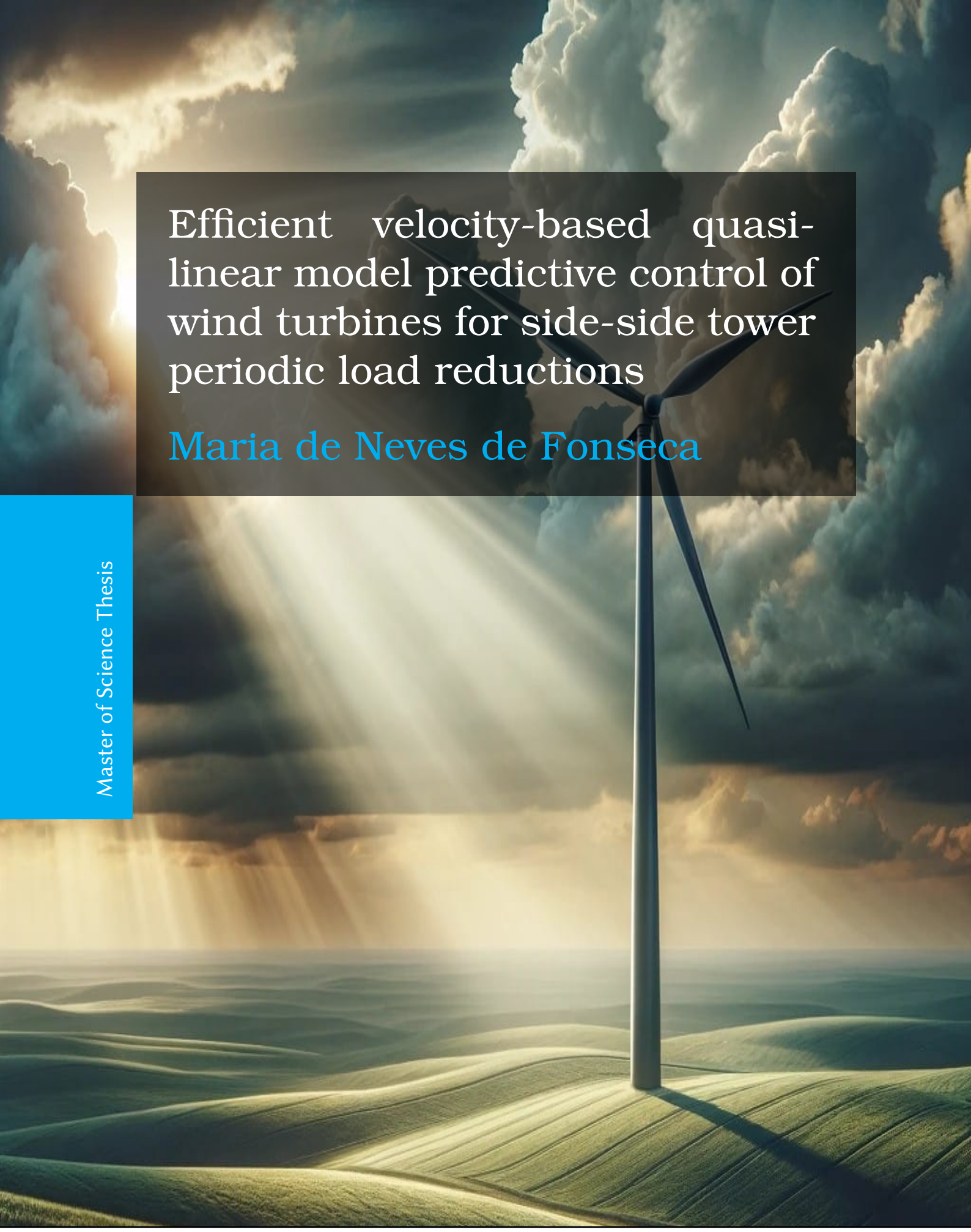

Efficient velocity-based quasi-linear model predictive control of wind turbines for side-side tower periodic load reductions

Maria de Neves de Fonseca

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Efficient velocity-based quasi-linear model predictive control of wind turbines for side-side tower periodic load reductions

Master of Science Thesis

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

Maria de Neves de Fonseca

June 22, 2024

Faculty of Mechanical Engineering (ME) · Delft University of Technology

The image on the cover page was generated with the assistance of AI.

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 Engineering (ME) for acceptance a thesis entitled

Efficient velocity-based quasi-linear model predictive control of wind turbines for side-side tower periodic load reductions

by

Maria de Neves de Fonseca

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: June 22, 2024

Supervisor(s):

Dr.ir. S.P. (Sebastiaan) Mulders

Reader(s):

Dr.ir. S.P. (Sebastiaan) Mulders

G. de Albuquerque Gleizer MSc

Dr. C. (Cosimo) Della Santina

T. (Tim) Dammann

Abstract

As the world shifts towards sustainable energy, wind power stands out as a compelling solution to combat climate change. Advances in wind technology, particularly larger turbines, have made it more cost-competitive. Taller towers and bigger rotors allow modern turbines to access better wind resources at greater heights, capturing more power and reducing costs.

Yet, achieving economically viable designs for taller towers requires reducing material usage. While this reduction decreases weight and costs, it also makes towers more prone to fatigue loading and material damage due to their inherent flexibility. This flexibility can lead to resonant behaviour, particularly in variable-speed wind turbines, exacerbated by factors like rotor mass and aerodynamic imbalances. Such resonances can cause material degradation, structural failure, and thus higher maintenance costs. The challenge of minimizing tower motion in wind turbines has been addressed with both active and passive conventional control strategies, largely for fore-aft tower movement. The current advanced control methods for managing tower side-side periodic loads rely on passive frequency-skipping techniques. However, there is a need for more advanced control methods capable of actively cancelling these loads. Therefore, this thesis aims to develop an efficient convex model predictive control method known as the velocity-based quasi-Linear Model Predictive Control (qLMPC) scheme, specifically designed for actively cancelling side-side periodic loads. This approach omits the need for extensive equilibrium input and state vectors, reducing memory usage while effectively capturing the system's nonlinear behaviour.

This study uses a Model Demodulation Transformation (MDT) technique to derive wind turbine tower dynamics, focusing on the real and imaginary parts of the tower top displacement and velocity signals. The goal is to minimize these outputs, reducing tower motions caused by periodic side-side forces. The model includes a wind turbine aerodynamic model, linearized through velocity-based linearization and controlled using a velocity-based qLMPC controller.

Optimization weights are carefully tuned to balance minimizing power output disruptions with mitigating disturbances from side-side periodic loads. Five simulation cases, with varying wind speeds and load profiles, were conducted. The results show the algorithm's simplicity, efficiency, and suitability for online applications. In all cases, the optimization objective is achieved, with the tower top position and velocity converging to zero, demonstrating the control algorithm's effectiveness in mitigating side-side periodic loads.

Table of Contents

Acknowledgements

In August 2021, I arrived in the Netherlands as an exchange master's student in electrical engineering at TU Delft. Little did I know that one year later, I would be starting my master's in Systems and Control at TU Delft. My experience at TU Delft as an exchange student was so great that I decided to apply to Systems and Control, and I was accepted, starting in August 2022.

Now, almost two years have passed, and I find myself finishing writing my master's thesis. Time has flown by, bringing with it a lot of memories, including stressful, tiring, and panicking moments, along with different experiences, food, cultural exchanges, new friends, life lessons and much more.

I would like to first thank my supervisor, Dr.ir. S.P. (Sebastiaan) Mulders, for his patience, competence and supervision during my thesis. Thank you for starting our Monday meetings by always asking about my weekend making the meetings more pleasant and comfortable. You have taught me a lot about writing more engagingly and I am so thankful for all the feedback you have provided during the writing of my thesis and literature survey. I have learned a lot from it and will for sure use most tips in the future during my PhD and when writing papers.

As an international student coming to the Netherlands in my early 20s, it has not always been easy. I've experienced homesickness, missing family and friends, and the adjustment to a completely different life. However, my friends from Systems and Control have made it a lot easier, always being there for me in happy and sad moments. I want to thank them for the support, therapy sessions, lunches, (way too many) coffee breaks, Fussball matches, and all the good memories that I will always keep with me. Also, I would like to take this moment to apologise to some of them who got stuck doing group projects with me. I know I can be a bit of a pain sometimes. A special note to Leon and Jep, two amazing friends I got from this master's program, and without whom I wouldn't have kept it together during the final stage of my thesis. I hope one day I can retribute all you have done for me.

Outside of the Systems and Control group, I would like to thank particularly Professor Karen Mulleners and all the PhDs I had the privilege to work with from the Unsteady Flow Diagnostics Laboratory at EPFL during my fellowship there in the summer of 2021. Even though my field of studies at the time was electrical engineering, I was grateful for the chance to learn more about fluid mechanics and to support PhD candidates with their experiments. The time I spent there in the lab was eye-opening to a whole world of opportunities that was now in front of me. I am thankful for the opportunity that Professor Karen Mulleners gave to me and for the motivation to pursue my dreams. I also want to thank my professors from my previous university in Portugal (FEUP), particularly Professors José

Carlos Alves, Bruno Ferreira, and Armando Sousa, for all the recommendation letters, motivational speeches, and for still checking on me once in a while.

To the good friends I made during my summer in Switzerland: Jacopo, Ayoub, Irena, Petar and Ivana. Thank you for being so supportive during my first time living away from home and for all the fun times, adventures, parties and for making me feel like I was understood. You will always be in my heart for making my first time living abroad so pleasant. Thank you Irena for being like an older sister and cooking for me, and listening to me crying every time I missed home. You all made me realize that with the right people around you, living far from home can be a little bit easier. Jacopo, who also was in the Netherlands with me for a short time, thank you for the fun moments both in Switzerland and in the Netherlands and for being such a good friend.

To the friends I left back in Portugal: I would like to show my appreciation, especially to Maria and José, the couple I most adore. Maria, who is now a doctor, thank you for all the coffees back in Portugal when I was visiting, for all the talks and therapy sessions. We have known each other for a while and you for sure have been a great help and friend while I was living abroad, always being there for me and my life crisis (all of them, which were a lot). I think you are one of the strongest people I know and wish you all the best in your next chapter. José, you are such a good friend and I am glad we have become closer during our third year in the bachelor. Thank you for being there for me and for all the opinions in motivation letters and assignments. I believe you have a life full of greatness ahead of you.

To my best friend Ana, thank you for being in my life for almost 19 years now. The friend with whom I shared most of my life adventures and misfortunes. The friend who's always been there for me no matter what. The one I consider to be my sister even though I am an only child. Thank you for being part of my life and for listening to me on the phone every time something went wrong with my thesis or when life was just hard here in the Netherlands. Thank you for all the phone calls, for all the hello's and goodbye's at the airport, and for being my best friend. On the same note, I would also like to thank Cláudia, Ana's mom, for calling me almost every week since I moved to the Netherlands to check on me and have a chat about life.

Finally, to my unconditional supporters - my family. To my closest cousins, Josua and Alex, thank you for all the deep conversations, laughs, and for all the good memories I have with you and our grandma Aulíria. Josua, I will never forget that it was you who introduced me to Rammstein, Epica, and Bring Me The Horizon who have been great supporters during my thesis. To my uncle, António, you are my crazy, chill uncle who makes me laugh and with whom I could just spend hours gossiping and talking about the most random subjects. Thank you for spending hours with me on the phone so I could relax a bit from my studies and for all the help you provided during my studies such that it would all be possible. And to the most important people in my life and who I love so much: I cannot begin to explain how grateful I am for my parents, Jorge and Emília. Since I was little, they always encouraged me to pursue my dreams and do my best. They have always worked so hard for me to be where I am today, and for that I am thankful. I wouldn't be who I am today if it wasn't for my parents, and for sure I wouldn't be brave enough to move to another country so young if they hadn't raised me strong and independent. Thank you for all the calls listening to me struggling with my thesis and courses, but also being completely energetic whenever things were going well. They know I can talk and talk and talk and never shut up. So thank you for always being there for me, for raising me to be who I am today, and for being so supportive. And Mom, thank you really for trying so hard to teach me how to cook Portuguese food, but it still hasn't worked out (yet). Dad, you are the best to have philosophical and mind-opening conversations with. I miss you both, every day.

I'm sorry for not always being there when you needed me because my head was all over the place, and for being cranky in some of our calls. I promise I will do better in my next chapter. (Portuguese side-note incoming: Não se preocupem que eu depois traduzo! :D)

Delft, University of Technology
June 22, 2024

Maria de Neves de Fonseca

Chapter 1

Introduction

As the world strives to transition to sustainable energy sources, wind power offers a promising solution to combat the escalating effects of climate change caused by our continued reliance on fossil fuels. Recent advancements in wind energy technology, particularly in scaling up wind turbines, have significantly improved its cost competitiveness. With taller towers and larger rotors, wind turbines can now tap into superior wind resources at higher altitudes and efficiently capture more power, driving down overall costs.

However, achieving economically feasible designs for taller towers necessitates a reduction in material usage. While this reduction confers benefits such as increased flexibility and reduced structural weight, it also exposes these towers to higher levels of fatigue. This susceptibility arises due to the inherent flexibility of tall, slender towers, which can induce resonant behaviour, particularly in variable-speed wind turbines. Several factors contribute to this side-side resonance, including rotor mass and aerodynamic imbalances, exacerbating the effects of periodic loading. When the side-side resonance frequency coincides with the rotor's rotational frequency or blade-pass frequency, the resulting amplified periodic loads lead to material fatigue and accelerated material degradation. Hence, this highlights the critical need for advanced control methods capable of managing the various structural loads encountered by wind turbines. Consequently, this thesis aims to develop an efficient convex model predictive control method that mitigates side-side tower periodic loads by combining existing frameworks found in the literature.

To set the foundation for this research, the upcoming chapter explores recent trends in wind energy and provides an overview of conventional wind turbine control strategies, specifically for tower load mitigation. It also outlines the current landscape of wind turbine control methodologies and the associated challenges. Finally, it clarifies the main research questions found upon reading the existing literature, goals, and contributions of this thesis, ending with a summary of its organizational structure.

1-1 Sustainability, climate relief, and wind energy

Climate change has emerged as one of the defining challenges of the 21st century, precipitating unprecedented environmental, societal, and economic consequences on a global scale. As temperatures continue to rise, weather patterns become increasingly erratic, and natural disasters become more frequent and severe, it is more crucial than ever to mitigate the impacts of climate change. Renewable energy has emerged as a critical solution to these challenges, offering a pathway to climate resilience and sustainability [1]. Renewable energy sources, such as wind, solar, hydroelectric, and geothermal power, have significant environmental benefits when compared to traditional fossil fuels. Unlike coal, oil, and natural gas, which emit greenhouse gases and contribute to global warming, renewable energy technologies produce negligible emissions during power generation, except for biomass and the biodegradable fraction of waste [2].

Currently, the world is moving towards more ambitious goals, such as a net 55% or greater reduction below 1990 levels by 2030, and a climate-neutrality objective by 2050. Achieving these goals will require even higher emission cuts by transitioning from fossil fuels to clean, renewable energy. This means halting deforestation, using land sustainably, and restoring nature until the release of greenhouse gases into the atmosphere is balanced with the capture and storage of these gases in our forests, oceans, and soil [3].

Wind power, in particular, has witnessed remarkable expansion in recent decades, exemplified by the global wind industry's record installation of 117 GW of new capacity in 2023 - an unprecedented feat signifying its ascension to prominence. This represents a 50% year-on-year increase from 2022. Furthermore, in 2023, the wind power industry saw continued global growth, with 54 countries representing all continents building new wind power. Hence, the Global Wind Energy Council (GWEC) revised its 2024-2030 growth forecast upwards by 10%, in response to the establishment of national industrial policies in major economies, gathering momentum in offshore wind and promising growth among emerging markets and developing economies [4]. This surge in wind energy adoption has been propelled by declining costs, making it increasingly competitive with traditional fossil fuel sources. For instance, onshore wind turbines have a Levelized Cost Of Energy (LCOE) range of 3.94 to 8.29 €Cent/kWh, which positions them as the second-most cost-effective technology for electricity production. However, offshore wind turbines are significantly more expensive, with an LCOE range of 7.23 to 12.13 €Cent/kWh, despite their higher annual mean Full Load Hours (FLH) of up to 4500 hours per year. The increased costs of offshore wind energy generation result from the more expensive installation, operation, and financing, which range from 3000 to 4000 €/kW [5]. Consequently, while offshore wind power has the potential for high returns, it may require additional investment and a longer payback period.

It is worth noting that the reduction in the LCOE for wind turbines is primarily due to the increase in their size over the years - see Figure 1-1. Larger capacity turbines help lower the LCOE by spreading the fixed costs over a greater number of megawatts. Furthermore, taller towers enable turbines to access higher wind speeds, which improves their capacity factor. Additionally, larger rotors capture more energy from the wind, resulting in more megawatt hours of electricity generated per MW of capacity installed, yielding a higher capacity factor, especially when paired with a reduction in specific power. This means that a larger rotor captures more energy from the wind flowing through the rotor swept area, allowing the generator to run closer to or at its rated capacity for a greater percentage of the time. All these factors contribute to the decrease in LCOE and make wind energy more cost-effective as discussed in the literature [1, 6, 7].

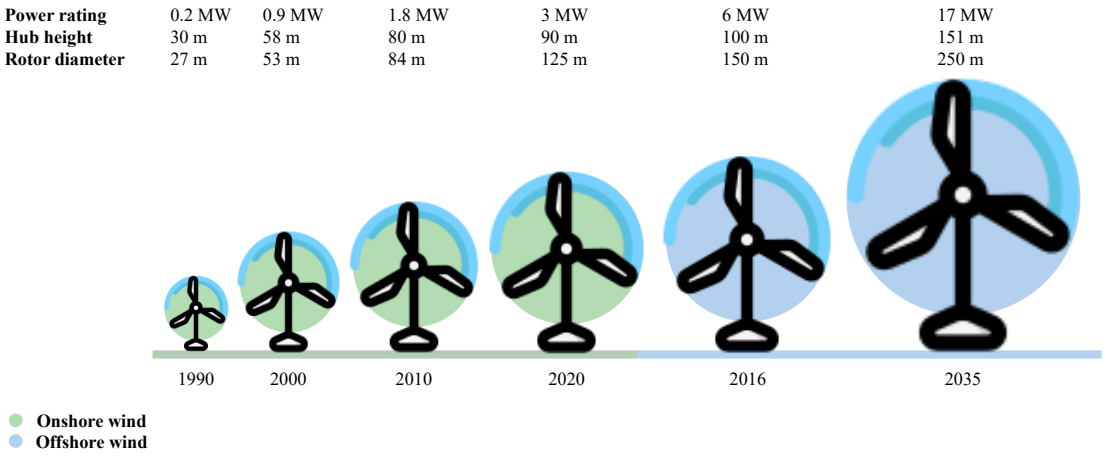

Figure 1-1: Wind turbines have grown in size over the past few decades. Wider rotors increase power capture, and taller towers allow high-quality wind to be harnessed at greater heights. This data was collected from [8, 9].

Overall, the widespread adoption and expansion of wind energy infrastructure offer a viable and economic pathway towards achieving carbon neutrality and addressing the urgent challenges posed by the climate crisis.

It is worth noting that while taller towers allow turbines to reach higher wind speeds, they also present some challenges. In the following section, the challenges posed by larger, slimmer, and more flexible towers are discussed.

1-2 Challenges of larger, slimmer, and flexible towers

The evolution of wind turbine tower design represents a critical frontier in advancing wind energy infrastructure, not only as a significant investment in wind power projects but also as a gateway to new markets in regions with limited wind resources [10]. Central to tower design is the dual objective of minimizing costs while ensuring structural integrity and manufacturability. This requires a delicate balance between material mass and manufacturing expenses, with the tower needing to withstand diverse operating conditions and extreme weather events. As turbine designers pursue greater hub heights, transportation challenges loom larger, necessitating an approach that augments tower base diameter while minimizing wall thickness [10]. For onshore applications, transport and installation difficulties may prevent a further increase in size, but for offshore applications, the difficulties are fewer. The reduction of operational and other costs could justify the size increase in offshore applications [11].

A key consideration in tower design is overall stiffness, impacting its natural frequency and structural dynamics. Stiff-stiff towers, with a natural frequency higher than the blade passing frequency, offer advantages in stability but may incur higher costs due to increased mass and material requirements. Conversely, soft-stiff towers, with a lower natural frequency, and soft-soft towers, with frequencies below both rotor and blade passing frequencies, are generally less expensive and lighter but demand meticulous dynamic analysis to prevent resonance excitation [12] - see Figure 1-2.

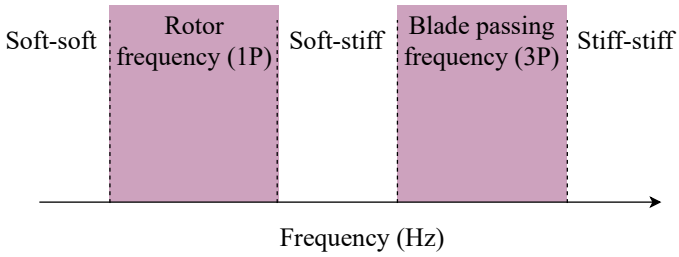

Figure 1-2: Stiff-stiff towers, with a natural frequency higher than the blade passing frequency, offer advantages in stability but may incur higher costs due to increased mass and material requirements. Conversely, soft-stiff towers, with a lower natural frequency, and soft-soft towers, with frequencies below both rotor and blade passing frequencies, are generally less expensive and lighter but demand meticulous dynamic analysis to prevent resonance excitation [12, 13].

Traditionally, towers have constituted a substantial proportion of the total turbine capital costs [10, 14], necessitating a careful trade-off between mass and manufacturing expenses. Conventional towers face constraints due to land-based transportation limitations, leading to thicker walls for taller towers and increased weight and costs. However, as wind turbine power rates continue to escalate, novel technical solutions emerge to facilitate the development of cost-effective tall towers by relaxing frequency constraints [15]. In this context, soft-soft tower configurations present a promising avenue for realizing tall towers. With reduced tower diameters and diminished wall thickness, soft-soft towers offer increased flexibility. Therefore, these towers can exhibit natural frequencies within the turbine operational range, increasing the risk of resonance excitation. Hence, the effective implementation of such innovative tower designs hinges on integrating advanced control solutions to actively attenuate and prevent prolonged excitation of undesirable frequencies [10, 15].

The next section addresses conventional and advanced control methods for tower load reduction that have been studied in the literature.

1-3 Control methods for tower load reduction

Wind turbines are complex nonlinear systems exposed to intricate mechanical stresses induced by wind loads, necessitating a strategic focus on mitigating structural loads for safe and reliable operation.

Of particular concern is the reduction of tower fatigue loads, which is especially pertinent in offshore wind turbines. This is underscored by the fact that the tower and foundation costs can comprise up to 40% of the total wind turbine expenditure [14]. Notably, wind turbine towers have limited damping and are thus susceptible to excitation by wind forces, particularly in the side-side direction, whereas the fore-aft direction benefits from significant aerodynamic damping.

The subsequent subsections delve into conventional active and passive control methods employed to alleviate tower loads. These approaches primarily leverage controllers from the Proportional-Derivative-Integral (PID) family. Additionally, recent advancements in predictive active and passive control techniques are explored, which offer the capability to address multivariable systems, take into account system constraints, and effectively balance multiple control objectives.

1-3-1 Conventional active control methods

To control turbine tower motion induced by wind excitations, conventional methods involve active load control. The numerically integrated tower-top acceleration measurements are scaled with a gain that is inversely proportional to the tower-top velocity. This effectively increases the tower damping, making it more resistant to fatigue loading [13]. In case of fore-aft tower motion, collective pitch is actuated to provide additional damping. Studies on this have been conducted and can be found in literature [16, 14, 17]. For side-side tower motion, either generator torque [18] or individual pitching [19] can be used. However, generator torque can affect power quality as it is directly coupled with power generation. On the other hand, individual pitching is more expensive due to increased wear and tear of the pitch mechanism.

1-3-2 Conventional passive control methods

Soft-soft towers, while designed to be flexible and efficient, can increase the chances of tower resonance due to rotor frequency excitation. This can lead to severe fatigue loading on the structure, especially when an intense 1P loading is present due to rotor imbalance caused by aerodynamics or extra masses. However, this problem can be addressed through a passive control technique called frequency skipping by speed exclusion zones [13].

Speed exclusion zones, also known as rotational speed windows or tower resonance bridging, can be helpful when the rotor speed (1P) or blade passing frequency (3P) excites a structural resonance at a particular operating point [20]. This technique modifies the control system to ensure that no stable operating points exist within the critical speed range. As a result, the rotor speed drives through the critical speed range without severely exciting the natural frequency. Typically, this is achieved by modifying the speed-torque curve of the generator, as explored in [21, 22, 23].

1-3-3 Advanced predictive active control methods

The challenge of reducing tower motion in wind turbines has been addressed in the literature through active and passive control strategies, as seen in the above subsections. These strategies, which often rely on PID feedback control loops, aim to prevent prolonged resonance excitation. However, they may not be able to balance the objectives of maximizing energy production and minimizing fatigue-induced loads. To address this issue, advanced control strategies such as Model Predictive Control (MPC) have been proposed due to their ability to accommodate constraints and address multiple objectives.

Robust MPC (RMPC) techniques have been used in the literature to actively dampen tower motion in the fore-aft direction, as demonstrated in [24]. Nonlinear MPC (NMPC) is another approach that assumes prior knowledge of future wind speeds through a Light Detection and Ranging (LIDAR) system, as discussed in [25] and a related work [26]. This approach can mitigate tower fore-aft motion in response to abrupt wind speed changes, as outlined in [26]. However, the computational complexity of solving non-convex Non-Linear Programming (NLP) problems often makes NMPC impractical for real-time applications.

The research presented in [27] introduces an alternative method for obtaining a quasi-Linear Parameter-Varying (quasi-LPV) model of the wind turbine, using velocity-based linearization. By defining the wind turbine dynamics as a quasi-LPV system, it becomes possible to apply quasi-Linear Model

Predictive Control (qLMPC) techniques to control the system. The study showed that qLMPC can significantly reduce fore-aft acceleration and power variance by nearly thirty times, compared to a baseline PI controller, when applied to below and above rated winds.

1-3-4 Advanced predictive passive control methods

While the MPC implementations aforementioned are active control strategies, there is a more passive MPC approach that has been recently explored in [15]. This passive MPC approach incorporates frequency-skipping capabilities to achieve an optimal balance between minimizing loads and maximizing energy production over a prediction horizon. The proposed method is designed to prevent side-side periodic excitation through a combination of Model Demodulation Transformation (MDT) and qLMPC techniques. Notably, the optimization problem associated with this approach is convex, enabling real-time computation.

The following section presents the research questions discovered during the literature review phase of this thesis, as well as the contributions made throughout the thesis work.

1-4 Research questions and thesis contribution

The current methods for mitigating tower side-side periodic loads predominantly rely on passive, frequency-skipping approaches. It is important to note that the technique described in [15] effectively prevents extended operation at critical rotational speeds by using frequency-skipping. However, it lacks the ability to actively counteract or reject periodic loads in practical situations due to its inability to estimate the unmeasurable and unknown periodic side-side loading. To address this limitation, [28] introduces a Kalman filter method for online estimation of the periodic loading as an extension of the work presented in [15].

As previously mentioned, the work presented in [27] explores an alternative approach to obtaining a quasi-Linear Parameter-Varying (quasi-LPV) model of the wind turbine. This involves using velocity-based linearization instead of the Model Demodulation Transformation (MDT) technique followed by Jacobian linearization as employed in [15]. This way, a velocity-based quasi-Linear Model Predictive Control (qLMPC) scheme can be used to control the system. However, the focus here is on mitigating the fore-aft acceleration of the wind turbine tower rather than the side-side periodic loading. Nevertheless, this implementation eliminates the need to store equilibrium input and state vectors, significantly reducing memory usage compared to the use of Jacobian linearization as in [15].

Upon a comprehensive analysis of these topics and their application to mitigating tower side-side periodic loads, certain aspects were identified as missing from the existing literature. Particularly, to the author's knowledge, a strategy that merges the MDT technique with velocity-based linearization to obtain a single quasi-LPV model of the wind turbine, allowing the qLMPC algorithm to actively mitigate the effects of the side-side periodic disturbance, has not yet been described in the literature. Therefore, it was concluded that the following areas deserve further investigation. One research direction is to explore the application of a qLMPC control scheme combined with a Kalman filter for estimating and mitigating periodic loads. This method could enable the rejection of periodic loads instead of merely avoiding operation at critical frequencies. Another research direction is to evaluate the efficiency of velocity-based linearization of the wind turbine's nonlinear model after

performing MDT, compared to MDT followed by Jacobian linearization. Velocity-based algorithms naturally possess disturbance rejection capabilities, potentially eliminating the need for disturbance estimators. This approach may lead to faster computations and offset-free control while rejecting periodic loads. Additionally, using velocity-based qLMPC removes the need to store equilibrium input and state vectors, significantly reducing memory usage while effectively capturing the system's nonlinear behaviour.

Addressing these research areas can contribute to advancing the understanding of wind turbine control and optimization, ultimately leading to more efficient and reliable wind energy generation systems. Hence, the following main research question is formulated:

Main research question: Can the tower side-side periodic load be completely mitigated through the implementation of an efficient velocity-based qLMPC scheme with as little influence on power production as possible?

Finally, this thesis presents two significant contributions aimed at addressing the main research question posed earlier. Firstly, the integration of MDT techniques with velocity-based linearization to create a unified quasi-LPV model of the wind turbine has not been explored in existing literature. This method effectively captures the system's nonlinear behaviour without the need to define or store operating points, as these naturally become the origin when mapped to velocity space. Hence, this constitutes the first contribution of the thesis.

Contribution 1: Derivation of a quasi-LPV model of the tower and wind turbine aerodynamics using MDT techniques and velocity-based linearization.

Secondly, conventional methods for mitigating periodic side-side tower loads, crucial for preventing tower resonance, have been limited to passive, frequency-skipping approaches. By developing a strategy that allows the qLMPC algorithm to actively mitigate the effects of side-side periodic disturbances, this thesis aims to fill a gap in the literature regarding the active cancellation of side-side tower periodic loads in wind turbines. Therefore, the final contribution of this thesis is:

Contribution 2: Design of a velocity-based qLMPC control scheme for actively cancelling periodic loads, specifically focusing on side-side tower periodic load mitigation.

1-5 Thesis outline

The thesis is divided into seven chapters. The first chapter is the introduction. The next three chapters cover the literature research conducted by the author, explaining fundamental concepts essential for understanding the control methods used in the thesis. The fourth chapter also presents results obtained using an illustrative example to enhance comprehension of certain concepts. The last three chapters focus on the algorithm design, implementation, simulation results and discussion, conclusions, and future work to be developed.

Chapter 2 introduces the concept of Linear Parameter-Varying (LPV) systems and highlights the benefits of using this framework to control nonlinear systems or those affected by external signals whose future values are unknown and unpredictable.

Chapter 3 extends to the predictive control of nonlinear systems via quasi-LPV representation.

Chapter 4 presents a mathematical framework of the velocity-based linearization technique to better define and understand its relevant characteristics.

Chapter 5 proposes the side-side controller design for reducing undesirable side-side periodic loads in wind turbine towers while minimizing the impact on power output.

Chapter 6 presents and discusses the results of five simulation cases designed to evaluate the performance and efficiency of the proposed framework.

Chapter 7 draws the main conclusions of this thesis and provides recommendations for future research.

Chapter 2

Linear and quasi-linear parameter-varying models

This chapter introduces the concept of Linear Parameter-Varying (LPV) systems and highlights the benefits of using this framework to control nonlinear systems or those affected by external signals whose future values are unknown and unpredictable. The notions of LPV embedding and quasi-LPV systems are also presented as these are two ways of representing nonlinear dynamics in the LPV framework. To better understand these concepts, the general nonlinear state-space representation is first introduced, followed by the definition of Linear Time-Invariant (LTI) systems and their well-established control methods.

2-1 Nonlinear systems

Before diving into the details of Linear Parameter-Varying (LPV) systems, which are the main subject of this chapter, the concept of nonlinear dynamical systems and their general mathematical definition are established. Firstly, let it be defined that in a nonlinear dynamical system, the relation between the input and output of the system is, as the name indicates, nonlinear. This relation is often shown in the following state-space representation, in continuous time:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t), u(t)), \end{aligned} \tag{2-1}$$

where $f(x(t), u(t)) : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_x}$ and $h(x(t), u(t)) : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R}^{n_y}$ are some nonlinear functions that can depend on the input $u \in \mathbb{R}^{n_u}$ and state $x \in \mathbb{R}^{n_x}$, and $y \in \mathbb{R}^{n_y}$ is the output.

Noteworthy is that Linear Time-Invariant (LTI) and LPV systems can be derived through linearization of (2-1). For this, there exist several linearization techniques, such as for instance, Jacobian and velocity-based linearization. Hence, it is important to note that nonlinear dynamics can be expressed either by an LTI model in the neighbourhood of an equilibrium point or by a single LPV system that depends continuously on measurable time-varying parameters [29]. These linear models, explored

in upcoming sections, are notably more manageable than the original nonlinear system. This allows the application of linear control and design methodologies, in contrast to the considerably more complex nonlinear approaches.

2-2 LTI systems

Linear Time-Invariant (LTI) systems are a widely studied class of dynamical systems in the literature. These systems exhibit a constant linear relationship between the input and output, which remains unchanged over time. This behaviour is commonly represented by a state-space model, in continuous time, as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2-2}$$

where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are continuous functions, $u \in \mathbb{R}^{n_u}$ is the input, $y \in \mathbb{R}^{n_y}$ is the output, and $x \in \mathbb{R}^{n_x}$ is the state variable. Note that in the case of LTI systems, the continuous functions A, B, C, and D are constant matrices independent of time due to the time-invariance property.

As might be expected, the LTI state-space model presented in (2-2) can be directly derived from (2-1), without resorting to linearization techniques, in case $f(x(t), u(t))$ and $h(x(t), u(t))$ are linear functions. However, most physical systems are of nonlinear nature when studied across a wide range of operations. In fact, nonlinear dynamics can be found in most engineering applications. Fortunately, in practice, even complex nonlinear systems are known to behave approximately linearly within some range around an operating or equilibrium point. Thus, it is reasonable to describe the system's behaviour using an LTI model when considering just a sufficiently small enough region [30]. Jacobian linearization of the nonlinear system is the most common approach used to derive the LTI model, as it will be explained in the following. Let $(\bar{x}, \bar{u})$ be an operating point of the original system with A, B, C, and D being constant matrices that define the LTI state-space model as in (2-2). These matrices can be obtained from the partial derivatives of the nonlinear functions $f(x(t), u(t))$ and $h(x(t), u(t))$ from (2-1) for when $(x(t), u(t)) = (\bar{x}, \bar{u})$ as shown below:

$$\begin{aligned} A &:= \left.\frac{\partial f(x(t),u(t))}{\partial x}\right|_{(x(t),u(t))=(\bar{x},\bar{u})} \quad , \quad B := \left.\frac{\partial f(x(t),u(t))}{\partial u}\right|_{(x(t),u(t))=(\bar{x},\bar{u})} \\ C &:= \left.\frac{\partial h(x(t),u(t))}{\partial x}\right|_{(x(t),u(t))=(\bar{x},\bar{u})} \quad , \quad D := \left.\frac{\partial h(x(t),u(t))}{\partial u}\right|_{(x(t),u(t))=(\bar{x},\bar{u})} \end{aligned} \tag{2-3}$$

Hence, it can be concluded that the behaviour of nonlinear dynamical systems around a neighbourhood of an equilibrium point can be approximated by an LTI model. This is an important note since most physical systems, for example, those of mechanical or electrical nature, are nonlinear. When linearizing these systems and obtaining the respective LTI model, the previously complex dynamics become rather simple to analyze and control, at least in a small range around an equilibrium. Furthermore, it is possible to use well-known efficient linear control techniques for LTI systems to control the nonlinear original system in that neighbourhood. In the following subsection, a few popular control methods for LTI systems are presented.

2-2-1 Control methods for LTI systems

Control methods for LTI systems are well-established and widely used due to their mathematical tractability and practical applicability. In the following, commonly used methods to control LTI systems are briefly presented, and references to popular literature are also provided.

The classic Proportional-Integral-Derivative (PID) control stands out as a straightforward yet effective technique for regulating systems with uncomplicated dynamics and known parameters [31, 32]. Another well-known technique is the Linear Quadratic Regulator (LQR) considered to be an optimal control strategy, aiming to minimize a quadratic cost function that includes both state deviations and control efforts [31, 32, 33]. When it comes to systems with uncertain or noisy dynamics, the Linear Quadratic Gaussian (LQG) control integrates LQR with a Kalman filter, composing an approach well-suited for this kind of system [32, 34, 35]. Meanwhile, Model Predictive Control (MPC) is a popular technique for its ability to handle complex dynamical systems subject to constraints, which seems to be an absent property in the other discussed control methods. Notably, MPC not only addresses complexity but also optimizes the system's response [36]. Lastly, $\mathcal{H}_\infty$ control is a robust control strategy particularly effective for LTI systems influenced by uncertainties or disturbances of considerable magnitude [37].

2-3 LPV systems

As previously mentioned in Section 2-2, in many cases, it is possible to describe the dynamics of a nonlinear system around an equilibrium or operating point by a Linear Time-Invariant (LTI) model through the Jacobian linearization of the original nonlinear system. This allows for the use of the same linear control design techniques mentioned earlier in Subsection 2-2-1. However, when applying the Jacobian linearization technique, the nonlinear system can only be controlled over a small region close to the operating point. If the nonlinear system needs to be controlled across a wider operating range or if the system's dynamic response is dependent on an external parameter that cannot be influenced by the designed controller, a more complex mathematical model framework is necessary [38].

Such a framework is referred to in the literature as a Linear Parameter-Varying (LPV) model, first introduced by Shamma in 1988 to model gain-scheduling in his PhD thesis [39], and also by Shamma and Athans later in 1990 [40]. In these types of systems, there is still a linear relationship between the input and output. However, this relationship may now depend on exogenous non-stationary parameters. This dependence is shown in the following continuous-time state-space representation:

$$\begin{aligned} \dot{x}(t) &= A(\theta(t))x(t) + B(\theta(t))u(t) \\ y(t) &= C(\theta(t))x(t) + D(\theta(t))u(t), \end{aligned} \tag{2-4}$$

where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are continuous maps, $u \in \mathbb{R}^{n_u}$ is the input, $y \in \mathbb{R}^{n_y}$ is the output, $x \in \mathbb{R}^{n_x}$ is the state variable, and $\theta \in \mathbb{R}^{n_\theta}$ is an external parameter that can depend on time [41]. The latter is also referred to in the literature as external *scheduling variable* and its current value is assumed to be measurable, for implementation purposes [38]. The parameter vector $\theta(t)$ belongs to a compact set of admissible values, i.e. $\theta(t) \in \Theta \subseteq \mathbb{R}^{n_\theta}\ \forall t \geq 0$ [42]. Note that the time dependence of the scheduling parameter will be omitted to shorten notation.

It can be realized that while LTI models of the form (2-2) are stationary, LPV models are not due to the exogenous time-varying parameter θ. Nevertheless, LTI models can be considered as a subset of the LPV models when taking the external parameter θ as a constant value.

2-3-1 LPV embedding

As previously stated, the LPV framework was instigated to model gain-scheduling. Gain-scheduling is a widely used technique to control complex nonlinear systems or processes across a wide range of operations. This approach emerged as a solution to address the limitations inherent in linearizing a nonlinear system around a single operating or equilibrium point. Such linearization restricts the application of LTI control techniques to a narrow vicinity around that specific point.

In contrast, gain-scheduling adopts a more versatile strategy by linearizing the original nonlinear system across multiple operating points. This results in a collection of local LTI models, each of which can be effectively controlled using LTI controllers within the neighbourhood of their respective operating point. In real-time, the dynamic switch, interpolation, or scheduling, between these diverse LTI controllers is facilitated through the use of available online measurements, often referred to as the scheduling variable. This enables the construction of a global controller for the original nonlinear process [38, 41].

Regardless of being a good tool to control complex nonlinear dynamical systems, the classical gain-scheduling method comes with no guarantees on the robustness, performance, or even nominal stability of the overall gain-scheduled system as referred in [40]. It is important to realize that as the local designs rely on LTI approximations of the plant, the designer can ensure that the feedback system possesses essential feedback requirements, like robust stability, robust performance, and stability, at each operating point. However, given the inherent nonlinearity of the actual system, the overall gain-scheduled system might not exhibit any of these traits. Hence, while the individual subsystems remain stable in a closed-loop for the respective operating points, the overall system's stability is not guaranteed. This is influenced by the selected operating points and the timing of interpolation, which in turn is determined by the defined scheduling variables.

To solve this issue, a global modelling technique known as *embedding* can be used [38, 43, 44]. In this case, instead of interpolating several local LTI models to define the global controller, a single LPV model can be determined. Noteworthy is that LPV models can approximate nonlinear dynamical systems effectively by abstracting away the nonlinearities of the original system through the introduction of an exogenous parameter θ, resulting instead in a linear, but nonstationary, system [41]. This is done through the so-called LPV embedding of the nonlinear system. Subsequently, stability and performance guarantees are provided within the context of the LPV embedding and can be readily extended to the original nonlinear system, contrary to the classic gain-scheduling design previously mentioned [42].

The following explanation outlines the process of constructing an LPV embedding for a nonlinear system. The visualization of these steps is presented in Figure 2-1. Let $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ denote a scheduling variable designed in a manner that enables the representation of a nonlinear system in the form of (2-1) (Figure 2-1a) to be equivalently transposed into an LPV model similar to (2-4):

$$f(x(t), u(t)) = A(\theta)x(t) + B(\theta)u(t). \tag{2-5}$$

Notably, θ could be a function of both states and inputs, allowing for a mapping $(x(t), u(t)) \rightarrow \theta$ to exist (Figure 2-1b). This is also referred to as *quasi-LPV system* in the literature, as it will be seen

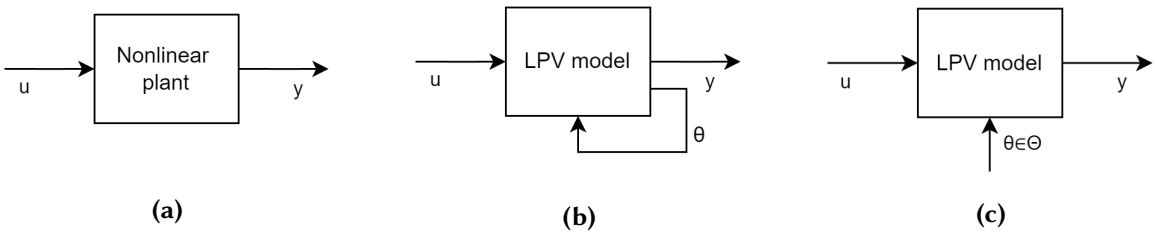

Figure 2-1: Construction of an LPV embedding of a nonlinear system [38, 44]. (a) Original nonlinear system. (b) Characterization of θ. (c) LPV embedding of the nonlinear system.

in Subsection 2-3-2. Now, consider $X \subseteq \mathbb{R}^{n_x}$, an open, bounded, and invariant set aligning with an operational range of the initial nonlinear system. Consequently, the parameter θ can be transformed into an external signal, effectively allocated within an appropriate bounded scheduling set denoted as $\Theta \subseteq \mathbb{R}^{n_\theta}$. This shift allows the mapping $(x(t), u(t)) \rightarrow \theta$ to be substituted with the inclusion constraint $\theta \in \Theta$ (Figure 2-1c) [38, 44].

Understanding the admissible trajectories of the LPV embedding is crucial. It is important to note that the LPV embedding has a much broader range of acceptable trajectories and that these are actually a superset of the set of trajectories of the initial nonlinear system. Therefore, if a controller can stabilize the LPV embedding, it is also capable of stabilizing the nonlinear system. To design controllers for LPV systems, LTI control methods like sensitivity shaping and modelling tools can be of use as these techniques have been extended to and are accessible for the LPV framework, as mentioned in [45]. The beauty of this is that these methods will still function effectively on the original nonlinear system due to the superset property, eliminating the need for much more complex nonlinear techniques [38], which tend to also be computationally heavier.

2-3-2 Quasi-LPV systems

It is crucial to understand that the categorization of LPV systems hinges on the nature of the scheduling variable θ into two distinct definitions: *general* or *pure* LPV systems and *quasi*-LPV systems. When θ is defined as an external signal characterized by unknown and unpredictable future values, it falls under the category of a general LPV system. In such instances, the scheduling parameter is considered uncertain.

In contrast, when the scheduling parameter θ assumes the role of modelling nonlinear dynamics, effectively becoming a function of the system's states and inputs, it becomes possible to predict its future values. This variant is referred to as a quasi-LPV system, as elaborated upon in existing literature [46, 47].

A quasi-LPV system conforms to the definition of an LPV system outlined in (2-4), where the time-varying parameters are determined by functions of inherent signals (i.e, states and inputs), expressed as:

$$\theta = \vartheta(x(t), u(t)). \tag{2-6}$$

Here, $\vartheta : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_\theta}$ denotes a continuous function on the compact set of admissible values Θ. It is worth emphasizing that within quasi-LPV systems, the presence of $\theta \in \Theta$ inherently outlines

a subset of the state space X, establishing a relationship where if $x \in X$, it implies that $\theta \in \Theta$. From this, it can be concluded that any stability result derived through a quasi-LPV methodology can only be defined locally, due to the fact that no stability-related conclusions can be drawn when $\theta \notin \Theta \Rightarrow x \notin X$ [42].

Noteworthy is that quasi-LPV parameterizations are inherently non-unique, and thus deriving a quasi-LPV model from a nonlinear system can be challenging and worth special attention. Notably, two prominent methodologies stand out: the ad-hoc approach and the Jacobian linearization-based parameterization [42, 46]. Nevertheless, there are other parameterization techniques such as, for instance, velocity-based linearization. This approach leads to a precise representation of the nonlinear dynamics of the original system, facilitating the assurance of stability along with offset-free control [48].

As an important remark, the following chapter will predominantly concentrate on the examination of quasi-LPV models for nonlinear systems, covering their control methodologies, particularly Model Predictive Control (MPC), and real-world applications.

2-4 Summary

The main concepts covered in this chapter are outlined as follows:

- In the vicinity of an operational or equilibrium point, nonlinear systems can be approximated as Linear Time-Invariant (LTI) models using a process called linearization. Alternatively, nonlinear systems can be represented as a Linear Parameter-Varying (LPV) model using a global modelling technique called embedding;
- While LTI models are considered stationary, LPV models are not due to the measurable exogenous time-varying parameter θ. Nevertheless, LTI models can be considered as a subset of the LPV models when taking the external parameter θ as a constant value;
- When modelling a system, both external and internal signals, as well as any nonlinearities affecting the original system, can be represented as a scheduling parameter θ. The choice between a pure LPV or quasi-LPV representation depends on whether θ models exogenous signals or nonlinear dynamics, respectively. In the former case, future values of θ are unknown and their prediction is not possible. Conversely, in the latter case, θ can be expressed as a function of states and inputs, making its future values predictable;
- LPV models can be used to embed nonlinear dynamics in a linear model, which makes it possible to extend LTI stability results to nonlinear systems. Additionally, this also allows the use of powerful LTI control and design techniques that have been proven to effectively work with LPV models to control the original nonlinear system, without resorting to the use of complex nonlinear control techniques.

Chapter 3

Quasi-linear model predictive control

In Chapter 2, the foundations for understanding the concepts of general Linear Parameter-Varying (LPV) and quasi-LPV models were defined, as well as the relevance of these model types to model nonlinear dynamical systems or systems affected by exogenous signals. It was especially introduced that LPV models can be used as a tool to embed nonlinear dynamics in a linear model, allowing for the extension of Linear Time-Invariant (LTI) control methods and stability results in nonlinear systems.

This chapter extends to the predictive control of nonlinear systems via quasi-LPV representation. Hence, it aims to present the fundamentals of Model Predictive Control (MPC) and, mainly, focuses on the state of the art in MPC via quasi-LPV representation. The state prediction formulation and control algorithm definition of quasi-Linear MPC (qLMPC) are also provided.

3-1 Model predictive control

This section provides a brief overview of the concept of Model Predictive Control (MPC). In 1987, the fundamentals for the MPC framework were first presented with Generalized Predictive Control (GPC) [49, 50]. MPC is a powerful and widely used control strategy for its ability to consider multiple objectives and constraints such as the physical limitations of the system's actuators, and also for the easy extension to multiple input-multiple output systems [51]. This control technique is also capable of predicting the system's future behaviour and choosing the right control actions that yield an optimal response for the system. These control actions are obtained via an optimization problem and the optimality is defined concerning a specified performance index [42]. However, it is important to note that an accurate mathematical model of the system is needed when using this control scheme to make a proper prediction of the system's behaviour.

When using MPC, the model of the system is often provided in discrete-time, describing the evolution of the states and output of the system for each time step k:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= h(x(k), u(k)), \end{aligned} \tag{3-1}$$

where $f(x(k), u(k)) : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_x}$ and $h(x(k), u(k)) : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_y}$ are functions that can depend on the input $u \in \mathbb{R}^{n_u}$ and state $x \in \mathbb{R}^{n_x}$, and $y \in \mathbb{R}^{n_y}$ is the output. Given the accurate dynamics of the system, its future behaviour can be obtained through a prediction model. To define how far into the future the output y is predicted, a *prediction horizon* $N_\mathrm{p} \in \mathbb{Z}^+$ is established. Additionally, the *control horizon* $N_\mathrm{c} \in \mathbb{Z}$ is given to define how far into the future the input u is predicted. Note that $N_\mathrm{c} \leq N_\mathrm{p}$, although it is often assumed that $N_\mathrm{c} = N_\mathrm{p} = N \in \mathbb{Z}^+$. Nevertheless, there are several reasons why a control horizon smaller than a prediction horizon may be chosen. For example, selecting a smaller N_c means that fewer variables need to be computed in the optimization problem at every control interval, resulting in faster computations. A smaller N_c can also promote the stability of the controller, although this is not guaranteed. Additionally, if the plant includes delays, it is essential to have $N_\mathrm{c} < N_\mathrm{p}$. Otherwise, some manipulated variables (in this case, control inputs) may not have an effect on any of the plant outputs before the end of the prediction horizon [52].

It is relevant to mention that in any real system, the actuators have limitations which lead to the saturation of the control input values u. Moreover, many systems are also constrained in their states x. Thus, these limitations need to be considered for the prediction of future behaviour to be precise. For this, the said limitations are presented as constraints usually defined as $u \in \mathcal{U}$ and $x \in \mathcal{X}$, respectively, where $\mathcal{U} \subset \mathbb{R}^{n_u}$ and $\mathcal{X} \subset \mathbb{R}^{n_x}$ are sets of admissible values that define feasibility [42].

The sequence of actions U_k inside the prediction horizon N is defined as the following column vector:

$$U_k = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix} \in \mathbb{R}^{Nn_u}, \tag{3-2}$$

which is determined through the following optimization problem that is solved online:

$$\begin{aligned} \min_{U_k} \quad & J_k(U_k) = \min_{U_k} \sum_{i=1}^{N} l(x(k+i), u(k+i-1)) + V_f(x(k+N)) \\ \text{subject to } & x(k+i) = f(x(k+i-1), u(k+i-1)) \quad \forall i \in [1 \quad N], \\ & u(k+i-1) \in \mathcal{U} \quad \forall i \in [1 \quad N], \\ & x(k+i-1) \in \mathcal{X} \quad \forall i \in [1 \quad N], \\ & x(k+N) \in \mathcal{X}_f, \end{aligned} \tag{3-3}$$

where J_k is the *finite horizon cost* composed by $l(x, u)$ - the *stage cost* - and $V_f(x)$ - the *terminal cost.*

The stage cost $l(x, u)$ is often defined as the following:

$$l(x, u) = x^\top Q x + u^\top R u, \tag{3-4}$$

where $Q = Q^\top \succeq 0 \in \mathbb{R}^{n_x \times n_x}$, $R = R^\top \succ 0 \in \mathbb{R}^{n_u \times n_u}$ are weighting matrices that can be tuned. In Equation (3-4), the term $x^\top Q x$ represents a penalty on the deviation of the state x from the origin and the term $u^\top R u$ represents the cost of control. Notice that from this, it can be concluded that the desired state is the origin.

Since the optimal control problem defining the control strategy operates within a finite horizon, it does not guarantee stability or optimality for the cost function through the use of a receding horizon

or MPC. However, stability can be achieved by introducing both a suitable terminal cost function and a terminal constraint into the optimal control problem.

Incorporating a terminal cost function into the optimal control problem typically introduces minimal or negligible additional computational complexity for online solving while often leading to enhanced performance. This terminal cost function must adhere to the criteria of being a local Control Lyapunov Function (CLF) [36]. The terminal cost, denoted as $V_f(x)$, is typically defined as:

$$V_f(x) = x^\top P x. \tag{3-5}$$

Here, it is essential to note that the matrix P is positive definite ($P > 0$), and $V_f(x)$ satisfies the conditions of $V_f(x) > 0$ for all non-zero states ($\forall x \neq 0$) and $V_f(0) = 0$.

It is worth mentioning that the set $\mathcal{X}_f$, referred to as the terminal set, is carefully designed to ensure closed-loop stability. In many cases, the inclusion of the terminal constraint $x(k+N) \in \mathcal{X}_f$ is imperative precisely because the terminal cost generally serves as a local CLF defined within the set $\mathcal{X}_f$, instead of a global CLF. Thus, to harness the benefits of the terminal cost, the terminal state x_N needs to be confined within $\mathcal{X}_f$. Additionally, the terminal set must be control invariant, ensuring that once the terminal state resides within this set, there exists at least a control action such that the next state will remain in the terminal set. Nevertheless, it is important to realize that introducing a terminal constraint may significantly increase the complexity of the optimal control problem [36].

Finally, the optimal input trajectory U_k^* is obtained as the solution to the optimization problem in (3-3) and its first element $u(k)^*$ is applied to the system. This concept is known as *receding horizon*. In the next time instant $k+1$, the same process is repeated and the optimization problem that minimizes J_{k+1} is solved for U_{k+1}, providing the new control law $u(k+1)$. The primary reason for implementing the controller in a receding horizon fashion is to ensure robustness against unexpected disturbances caused by external factors or discrepancies in the model. Essentially, the receding horizon approach serves as a means to connect the plant and controller in a closed loop during each time step, as discussed in [42].

Note that there exist other formulations for the optimization problem and that the cost function can be defined differently depending on the purpose of the MPC controller, i.e. for the tracking problem, the reference is also included in the cost function and the latter is defined as the error between the actual value and the desired one. The formulation presented in this section is based on the one found in [53] and used in the regulator problem where the objective is to stabilize the origin of the state-space.

3-2 Efficient quasi-linear model predictive control

As mentioned before, Model Predictive Control (MPC) is a technique used to solve an optimization problem in real-time while taking into account multiple objectives and constraints. Although MPC can be applied to systems that are not Linear Time-Invariant (LTI), it is commonly used in systems that match the form presented in Equation (2-2). This makes the computations fast and efficient, as the optimization procedure is actually a Quadratic Problem (QP). However, when dealing with complex nonlinear systems, the computations become more complicated. In these cases, MPC needs to solve a non-convex Non-Linear Program (NLP) problem, which is not as straightforward as in the linear case [51, 54] and often not suitable for systems with fast dynamics. It is also important to note that solving an NLP does not guarantee finding the global optimum [54].

To overcome this computational issue, various techniques have been developed and studied in the literature. Nonlinear Model Predictive Control (NMPC) was one of the first methods developed, but it was computationally challenging for real-time embedded applications due to the added complexity of the nonlinear components [54], as mentioned above. To address this, sub-optimal NMPC strategies were developed that approximately solve the problem, leading to near-optimal control. These sub-optimal NMPC methods were found to be a suitable solution to tackle the computational challenges [55]. Additionally, several other numerical methods for efficient and fast NMPC were studied to speed up the operation of NMPC [56]. These techniques are relevant and especially important when controlling systems with fast dynamics for which optimization procedure needs to be quick enough to provide the next control action.

In Chapter 2, it was mentioned that many nonlinear systems can be defined in the Linear Parameter Varying (LPV) framework as either pure LPV systems or quasi-LPV systems. To control these systems, several MPC approaches have been developed over the years. It is important to remember that pure LPV models have unpredictable scheduling parameters in the future but are measurable online. This uncertainty makes it difficult to predict the model for MPC. On the other hand, quasi-LPV systems have scheduling variables determined by the state and/or input, making them predictable. However, the resulting prediction equations are nonlinear [29, 42].

A recent study by Cisneros et al. introduced a new algorithm known as quasi-Linear Model Predictive Control (qLMPC) [53]. This strategy involves modelling a nonlinear system as a quasi-LPV system and using an MPC approach to control it. Due to the linearity property between inputs and outputs provided by this LPV framework, the drawbacks of the NMPC algorithms can be bypassed. Unlike certain fast NMPC methods that depend on a moving linearization, the LPV framework allows for an exact representation of the nonlinear system without the need for time-consuming linearization. As concluded in [53], this method is appealing because it is similar to the implementation of the standard LTI MPC and has relatively fast computations. This makes it possible to extend LTI MPC literature results to quasi-LPV, making them readily available for nonlinear systems [42]. Furthermore, if strict stability guarantees are not critical, the predictions can be made without the terminal constraint, which simplifies the online optimization problem into a QP that is easily solved using standard solvers, making it suitable for application in fast real-time systems. In the following subsection, a more concrete formulation of the qLMPC framework is provided.

3-2-1 State prediction formulation

This subsection presents the mathematical formulation of the qLMPC framework developed in [53]. For this, let the following discrete quasi-LPV model in state-space form be considered:

$$x(k+1) = A(\theta(k))x(k) + B(\theta(k))u(k), \tag{3-6}$$

where $u \in \mathbb{R}^{n_u}$ is the input, $x \in \mathbb{R}^{n_x}$ is the state variable, $\theta(k) = \vartheta(x(k), u(k)) \in \mathbb{R}^{n_\theta}$ is the scheduling variable, and $\vartheta \in \mathbb{R}^{n_\theta}$, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are continuous maps. Let it be assumed that $\theta(k) \in \Theta \forall k \geq 0$ with Θ being a compact set. Furthermore, it is considered that the pair $(A(\theta), B(\theta))$ is stabilizable $\forall \theta \in \Theta$ [53].

From this point on, the concepts regarding the formulation of MPC presented in Section 3-1 will be used. Consider the optimization problem presented in (3-3), with the sequence of control actions U_k over the prediction horizon N defined in (3-2). The control law, as previously mentioned in Section 3-1, is implemented using a receding horizon. Keep in mind that, for this case, the dynamics of the

model, which are included as an equality constraint in the mentioned optimization problem, are the ones given in Equation (3-6). Therefore, the equality constraint in (3-3) becomes:

$$x(k+i) = A(\theta(k+i-1))x(k+i-1) + B(\theta(k+i-1))u(k+i-1) \quad \forall i \in [1 \quad N]. \tag{3-7}$$

Another assumption is made that $\vartheta(\mathcal{X}_f \times \mathcal{U}) \subset \Theta$ [53]. Additionally, let the following be defined:

$$X_k = \begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N) \end{bmatrix} \in \mathbb{R}^{Nn_x} \quad , \quad T_k = \vartheta([x(k)^\top \quad X_k^\top]^\top, U_k) = \begin{bmatrix} \theta(k) \\ \theta(k+1) \\ \vdots \\ \theta(k+N-1) \end{bmatrix} \in \mathbb{R}^{Nn_\theta}, \tag{3-8}$$

where X_k is a column vector of states and T_k is the collection of scheduling variables at each time step over the prediction horizon N.

Now using definitions (3-2) and (3-8), the following prediction model can be found:

$$X_k = H(T_k)x(k) + S(T_k)U_k, \tag{3-9}$$

where the matrices $\{H, S\}$ are defined as

$$H(T_k) = \begin{bmatrix} A(\theta(k)) \\ A(\theta(k+1))A(\theta(k)) \\ \vdots \\ A(\theta(k+N-1)) \dots A(\theta(k)) \end{bmatrix}, \tag{3-10}$$

and

$$S(T_k) = \begin{bmatrix} B(\theta(k)) & 0 & \dots & 0 \\ A(\theta(k+1))B(\theta(k)) & B(\theta(k+1)) & \dots & 0 \\ \vdots & & \ddots & \vdots \\ A(\theta(k+N-1)) \dots A(\theta(k+1))B(\theta(k)) & \dots & & B(\theta(k+N-1)) \end{bmatrix}. \tag{3-11}$$

Using this, the cost function $J_k(U_k)$ can be re-written as

$$\begin{aligned} J_k(U_k) = &(H(T_k)x(k) + S(T_k)U_k)^\top \tilde{Q}(H(T_k)x(k) + S(T_k)U_k) \\ &+ U_k^\top \tilde{R} U_k + V_f(x(k+N)), \end{aligned} \tag{3-12}$$

with $\tilde{Q} = \text{diag}_N(Q) \in \mathbb{R}^{N \times N}$ and $\tilde{R} = \text{diag}_N(R) \in \mathbb{R}^{N \times N}$.

3-2-2 Control algorithm definition

In the following explanation, a description of the algorithm employed in qLMPC is provided. For this, the following notation is used for the sequences of inputs, states and scheduling parameters for each time step k and iteration step l: U_k^l, X_k^l and T_k^l, respectively.

At the start of the process at time step $k = 0$, only the initial state is considered to be known. The scheduling parameters are chosen to remain unchanged throughout the prediction horizon for each iteration step l. As a result, the scheduling trajectories are static, in other words time-invariant. For example, initially, for $l = 0$ and at time step k the following sequence of scheduling parameters is defined:

$$T_k^0 = \mathbf{1}_N \otimes \vartheta(x(k), u(k-1)), \tag{3-13}$$

with $\mathbf{1}_N \in \mathbb{R}^N$ being a vector consisting of ones. This approach results in the initial predicted state sequence, denoted as X_k^0, becoming linear with respect to the initial control input sequence, referred to as U_k^0 [53, 15]. Consequently, the quasi-LPV model can be effectively substituted with an LTI model for each iteration step l.

The scheduling sequence T_k^l is gradually adjusted through iterative steps l to reach its optimal value, denoted as $T_k^* = \vartheta(X_k^*, U_k^*)$. In each iteration step l, an optimization problem resembling (3-3) is solved with T_k replaced by T_k^l. For the next iteration step $l + 1$, a new scheduling sequence $T_k^{l+1} = \vartheta(X_k^l, U_k^l)$ is generated using the resulting optimal state and input sequences from the previous iteration l, namely, X_k^l and U_k^l.

Furthermore, as previously mentioned and as discussed in [53], if strict stability guarantees are not mandatory, it is possible to omit the terminal state constraint. This omission enables the optimization problem to be efficiently solved as a QP as demonstrated in literature [29, 53, 15], for which various standard solvers are readily available. In case this terminal constraint is desired, the optimization problem becomes a Second Order Cone Program (SOCP) problem, for which solvers also already exist [53, 57]. Note that solving a SOCP problem is still considerably more efficient than solving a nonlinear optimization problem as referred in [53].

The stopping criteria chosen in [53] is based on the norm of the predicted trajectory change. In other words, the iteration continues until the condition $X_k^l - X_k^{l-1} < \epsilon$ is met, where ϵ is a predefined error threshold. Once this condition is achieved, the input sequence U_k^l is used as an approximation of the optimal solution U_k^* to (3-3). Consequently, X_k^* can also be derived, allowing the determination of T_k^*. The first element of the optimal input sequence is then applied to the system (receding horizon). It is worth noting that alternative stopping criteria, such as setting a maximum number of iterations $l_{\max}$, can also be chosen.

Once an approximation of the optimal solution is reached, the obtained X_k^l and U_k^l are applied in the subsequent time step $k + 1$ to compute $T_{k+1}^0 = \vartheta(X_k^l, U_k^l)$ while the iteration count l is reset to zero. This approach, known as *warm-start*, aids in achieving faster convergence as concluded in [53, 15].

For a clearer understanding, the pseudocode for the qLMPC algorithm described above can be seen in Algorithm 1.

3-2-3 Concluding remarks

In summary, the core concept behind this algorithm involves solving a series of optimization problems in which, by iteratively predicting the future scheduling parameter trajectory T_k, the quasi-LPV model (3-6) can be actually replaced by a Linear Time-Varying (LTV) model [29, 42, 53]. This allows the underlying optimization problem to be solved efficiently as a sequence of QPs [29], which consequently leads to faster computations compared to other NMPC techniques. As reported in [53], this

Algorithm 1 Pseudocode for the qLMPC algorithm [53]

$k \leftarrow 0$
Define Q, R, N
Initialize reference trajectory, plant model, constants $\tilde{Q}$, $\tilde{R}$
Define $T_k^0 = \mathbf{1}_N \otimes \vartheta(x(k), u(k-1))$, $x(0) = x_0$
repeat
    $l \leftarrow 0$
    repeat
        Solve (3-3) with T_k^l for U_k^l
        Predict X_k^l through (3-9) using T_k^l and U_k^l
        Define $T_k^{l+1} = f(X_k^l, U_k^l)$
        $l \leftarrow l + 1$
    until stop criterion
    Apply u_k to the plant of the system ▷ Receding horizon
    Define $T_{k+1}^0 = f(X_k^l, U_k^l)$
    $k \leftarrow k + 1$
until end

algorithm is relatively easy to implement and its complexity is similar to that of LTI MPC. Further, the qLMPC algorithm's computations were proven to be just about 3 times slower than for LTI MPC.

It is relevant to mention that in [57], the qLMPC algorithm was applied for reference tracking for nonlinear systems in the presence of nonlinear input and state constraints. However, it must be noted that there were no stability guarantees in this study. By implementing the quasi-LPV framework, it was discovered that the nonlinear optimization problem can be replaced by a standard QP, even when nonlinear state constraints are present. A simulation example demonstrated that this approach is a viable alternative to complex trajectory planners for a robotic manipulator when dealing with obstacles, as long as nonlinear constraints are properly encoded as quasi-LPV constraints. Additionally, the algorithm proved to be computationally efficient, with a speed of 2 orders of magnitude faster than nonlinear optimization-based NMPC.

3-3 Frameworks and applications

Throughout the years, advancements have been made in extending the quasi-Linear Model Predictive Control (qLMPC) algorithm to nonlinear systems in Input-Output quasi-Linear Parameter-Varying (IO-qLPV) form [58]. This form might be preferred in certain cases for practical applicability due to its simplicity and similar accuracy in identification techniques when compared to the ones used for the State-Space (SS) form and the lack of uncertainty brought by observers. The nonlinear optimization problem is solved as a sequence of Quadratic Problems (QPs) to keep computational complexity low. However, it can be harder to obtain IO-qLPV models analytically and has fewer tuning parameters, which can bring less flexibility [42].

Additionally, a velocity algorithm has been developed for Nonlinear Model Predictive Control (NMPC) [59], where the model can be readily expressed as a quasi-LPV model through velocity-based linearization [60]. Integral action for offset-free control is intrinsic to the velocity algorithm, without requiring disturbance estimators. It is important to mention that in the original qLMPC framework,

the parameterization of all equilibrium points is required. However, in the velocity space, all equilibrium points are mapped to the origin [42] removing the need for parameterization and even for the terminal constraint in the formulation of the optimization problem, making computations simpler. Also, stability can be guaranteed without the need for complex offline computations.

Furthermore, data-driven quasi-LPV MPC has been explored using Koopman operator techniques [51], which assumes little a priori knowledge of the underlying system in the form of basis functions. This method has been proven to be effective and efficient in uncovering the dynamics of an unknown system while using the resulting model for data-driven control. However, establishing stability in this approach is rather difficult because the basis functions may not completely correspond to the system, as these are design parameters.

The above-mentioned qLMPC frameworks have been utilized in addressing problems related to the control of nonlinear dynamical systems. Initially, the effectiveness of the algorithm has been validated through simulations and experiments with robotic manipulators, as reported in [57] and [48]. Thereupon, wind turbine tower vibration control has been achieved through the use of these frameworks as described in [15] and [27]. Autonomous vehicles have also been able to solve trajectory-tracking problems using qLMPC, as outlined in [61]. Furthermore, robotic manipulators have benefited from the velocity-based qLMPC algorithm, which enables offset-free tracking, according to [59]. Additionally, [62] and [29] used qLMPC to control fixed-wing Unmanned Aerial Vehicles (UAVs) for meteorological purposes and to design an efficient and fast flight controller for flexible aircraft G-Flights Dimona, respectively.

In conclusion, the qLMPC algorithm has been demonstrated to be a reliable method for controlling nonlinear dynamical systems in several areas of application. It was shown in the literature that qLMPC can generate solutions that are very similar to those produced by more complex NMPC methods, while greatly reducing computational expenses and enabling real-time operation.

3-4 Summary

Below are the main concepts covered in this chapter presented for reference:

- Model Predictive Control (MPC) is a technique that solves an optimization problem in real-time while considering multiple objectives and constraints. However, when dealing with nonlinear systems, the computations for MPC can become complex. This led to the development and study of other algorithms such as Nonlinear MPC (NMPC), which proved to be computationally challenging for real-time embedded applications. To address this issue, sub-optimal NMPC or fast NMPC methods were also developed. However, the optimization problem in these cases is still a non-convex Non-Linear Programming (NLP) problem;
- Recently, a new method called quasi-Linear MPC (qLMPC) has been researched and found to be a reliable way to control nonlinear dynamical systems across various applications. This algorithm combines the quasi-Linear Parameter-Varying (quasi-LPV) framework and MPC, allowing the optimization problem to become a convex Quadratic Programming (QP) problem. Studies have shown that qLMPC can generate solutions similar to those produced by traditional NMPC methods while considerably reducing computational expenses and enabling real-time operation.

Chapter 4

Velocity-based quasi-linear model predictive control

In Chapter 2, it was discussed that quasi-Linear Parameter-Varying (quasi-LPV) parameterizations are inherently non-unique. Therefore, deriving a quasi-LPV model from a nonlinear system can be challenging and requires special attention. Nonetheless, it was also mentioned that velocity-based linearization can be used to obtain directly a quasi-LPV model, providing a precise representation of the nonlinear dynamics of the original system. According to the literature, this parameterization technique facilitates the assurance of stability. Moreover, it was noted in Chapter 3 that velocity algorithms intrinsically have integral action for offset-free control, without requiring disturbance estimators.

In this chapter, a mathematical framework of the velocity-based linearization technique is presented to better define and understand its relevant characteristics. The control of a quasi-LPV model obtained through velocity-based linearization using a Model Predictive Control (MPC) scheme is referred to as velocity-based qLPMC throughout the rest of this thesis and is also given in this chapter. Later in the chapter, the value of the velocity-based qLPMC algorithm is illustrated through the results of a simulation of an illustrative example, specifically the forced Van der Poll oscillator, when controlled in closed-loop by the velocity-based qLPMC.

4-1 Velocity-based linearization towards a quasi-LPV model

This section discusses the mathematical formulation of the velocity-based linearization technique. This method is used to obtain quasi-Linear Parameter-Varying (quasi-LPV) models directly from the original nonlinear system. The formulation presented here is based on the work of [60].

When it comes to obtaining a quasi-LPV model, it is important to note that velocity-based linearization offers some advantages over the ad-hoc LPV parameterization discussed in Chapter 2. For instance, it is a simple calculation that does not require the designer's expertise or intuition. On the other hand, Jacobian linearization-based methods may also be used, as previously noted. However,

Jacobian linearization may lead to approximation errors and inconvenient affine state-space representations. Additionally, Jacobian linearization is only valid in the vicinity of an equilibrium or operating point, and the size of this neighbourhood is difficult to represent and may be quite small. In contrast, velocity-based linearization produces an exact and linear state-dependent (or quasi-LPV) model, as opposed to an affine one. This eliminates the need to store equilibrium or operating points, which is necessary when using Jacobian linearization.

To better understand how the velocity-based linearization approach works, let the nonlinear system dynamics be given as

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t), u(t)), \end{aligned} \tag{4-1}$$

where $f(x(t), u(t)) : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_x}$ and $h(x(t), u(t)) : \mathbb{R}^{n_x \times n_u} \to \mathbb{R}^{n_y}$ are some nonlinear functions that can depend on the input $u \in \mathbb{R}^{n_u}$ and state $x \in \mathbb{R}^{n_x}$, and $y \in \mathbb{R}^{n_y}$ is the output. For simplicity, let it be assumed that the output is only influenced by the states. These dynamics can be described by the relationship between the state and input derivatives:

$$\begin{aligned} \ddot{x}(t) &= \underbrace{\nabla_x f(x(t), u(t))}_{A_{\mathrm{c}}(x(t),u(t))} \dot{x}(t) + \underbrace{\nabla_u f(x(t), u(t))}_{B_{\mathrm{c}}(x(t),u(t))} \dot{u}(t), \\ \dot{y}(t) &= \underbrace{\nabla_x h(x(t))}_{C_{\mathrm{c}}(x(t),u(t))} \dot{x}(t). \end{aligned} \tag{4-2}$$

Note that the subscript c in the state, input, and output matrices, $A_{\mathrm{c}}(x(t), u(t))$, $B_{\mathrm{c}}(x(t), u(t))$, and $C_{\mathrm{c}}(x(t), u(t))$, respectively, stands for continuous-time. Also, it is interesting to realize that the model in (4-2) is linear in the derivative of the states, and the state-space matrices are now state-dependent [59]. Another important remark is that the steady-state information is lost during the process of differentiation when using velocity-based linearization [42]. Nevertheless, the steady-state information on the output can be recovered by state augmentation, giving a velocity-form state-dependent linear model such as follows:

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{y}(t) \\ \ddot{x}(t) \end{bmatrix}}_{\dot{x}_{\mathrm{ext}}(t)} &= \underbrace{\begin{bmatrix} 0 & C_{\mathrm{c}}(x(t), u(t)) \\ 0 & A_{\mathrm{c}}(x(t), u(t)) \end{bmatrix}}_{A_{\mathrm{c,ext}}(x(t),u(t))} \underbrace{\begin{bmatrix} y(t) \\ \dot{x}(t) \end{bmatrix}}_{x_{\mathrm{ext}}(t)} + \underbrace{\begin{bmatrix} 0 \\ B_{\mathrm{c}}(x(t), u(t)) \end{bmatrix}}_{B_{\mathrm{c,ext}}(x(t),u(t))} \dot{u}(t), \\ y(t) &= \underbrace{\begin{bmatrix} I & 0 \end{bmatrix}}_{C_{\mathrm{c,ext}}} \underbrace{\begin{bmatrix} y(t) \\ \dot{x}(t) \end{bmatrix}}_{x_{\mathrm{ext}}(t)}. \end{aligned} \tag{4-3}$$

It can be understood that this can be written as a quasi-LPV model by setting:

$$\theta = H[x(t)^\top, u(t)^\top]^\top \in \mathbb{R}^{n_\theta}, \tag{4-4}$$

where H is a selector matrix. Discretizing (4-3) gives:

$$\begin{aligned} x_{\mathrm{ext}}(k+1) &= A_{\mathrm{d,ext}}(\theta(k)) x_{\mathrm{ext}}(k) + B_{\mathrm{d,ext}}(\theta(k)) \Delta u(k) \\ y(k) &= C_{\mathrm{d,ext}} x_{\mathrm{ext}}(k), \end{aligned} \tag{4-5}$$

with $x_{\text{ext}}(k) = \begin{bmatrix} y(k) \\ \Delta x(k), \end{bmatrix}$ and the subscript d in the state, input, and output matrices, $A_{\text{d,ext}}(\theta(k))$, $B_{\text{d,ext}}(\theta(k))$, and $C_{\text{d,ext}}$, respectively, standing for discrete-time. The discretized velocity model can be obtained through a variety of methods, for instance, a fourth-order Runge-Kutta discretization method as in [15]. This model is then used in the quasi-Linear Model Predictive Control (qLMPC) formulation presented in Chapter 3.

It is important to realize that the augmented velocity-form state-dependent linear discrete-time model in (4-5) has built-in integral action, which is a result of adding the output variable $y(k)$ to the state vector. On the same note, it is interesting that integral action can handle modelling errors and disturbances effectively without the need for other disturbance estimators. This comes from the fact that, with integral action, the control input u is designed to actively counteract any constant (or very slowly-varying) disturbance or any other deviation from the reference value such that the error between the output and its reference converges asymptotically to zero. Consequently, integral action can help achieve offset-free tracking of constant reference signals [63].

However, even though disturbance estimators are not needed when using the model in (4-5), observers are still often necessary to estimate the unknown or unmeasurable states.

The following section relates the velocity-based linearization method introduced here with the qLMPC framework provided in Chapter 3.

4-2 Velocity-based quasi-linear model predictive control

In Chapter 3, the concept of quasi-Linear Model Predictive Control (qLMPC) was introduced. However, this framework has a limitation when it comes to tracking problems. As per the work of [42], deriving the terminal set for the optimization problem in (3-3) requires a partition of the state-space. This means that Linear Matrix Inequalities (LMIs) would have to be solved offline to get a terminal set, but this approach is only suitable for low-order systems. For higher-order systems, the LMIs become intractable.

Moreover, ignoring stability guarantees makes the approach more flexible, but it requires a parametrization of all equilibria or operating points to achieve offset-free control. An alternative approach is to solve a target selector problem. This method computes the optimal corresponding steady-state input and state vectors for a desired set point. These vectors are then used as references for the inputs and states, respectively.

However, when working in the velocity space, all equilibrium or operating points are mapped to the origin, which makes computations simpler and less time-consuming. This means that the parameterization of equilibrium or operating points and the terminal constraint in the optimization problem are not required when using the velocity-based qLMPC framework instead of the original one. Furthermore, stability can be guaranteed without the need for complex offline computations as it will be shown, since there is no need to solve LMIs to get a terminal set. This approach will be further elaborated in this section.

First, the optimization problem formulation for the velocity-based qLMPC is provided:

$$\begin{aligned}
\min_{\Delta u} \quad & J_k(\Delta u) \\
\text{s.t.} \quad & \begin{bmatrix} y(k+i) \\ \Delta x(k+i) \end{bmatrix} = A_d(\theta(k+i-1)) \begin{bmatrix} y(k+i-1) \\ \Delta x(k+i-1) \end{bmatrix} \\
& + B_d(\theta(k+1-i))\Delta u(k+i-1) \quad \forall i \in [1 \quad N], \\
& u(k+i-1) = u(k-1) + \textstyle\sum_{j=1}^{i} \Delta u(k+j-1) \in \mathcal{U} \quad \forall i \in [1 \quad N], \\
& x(k+i-1) \in \mathcal{X} \quad \forall i \in [1 \quad N], \\
& \Delta x(k+N) = 0,
\end{aligned} \tag{4-6}$$

with the cost function being defined as

$$\begin{aligned}
J_k(\Delta u) = \textstyle\sum_{i=1}^{N} (&||y(k+i-1) - y_{\text{ref}}||^2_{Q_1} + ||\Delta x(k+i-1)||^2_{Q_2} \\
&+ ||\Delta u(k+i-1)||^2_R) + ||y(k+N) - y_{\text{ref}}||^2_P
\end{aligned} \tag{4-7}$$

Note that the cost function in (4-7) includes a penalization on the extended state vector x_{ext} with the weighting matrix $Q_1 \in \mathbb{R}^{n_y \times n_y}$ acting on y and $Q_2 \in \mathbb{R}^{n_{\Delta x} \times n_{\Delta x}}$ on Δx. There is also a penalty for the control input Δu given by the weighting matrix $R \in \mathbb{R}^{n_{\Delta u} \times n_{\Delta u}}$, and a terminal cost that defines that, at the end of the horizon, the output should go to its reference value y_{ref}.

An important remark about the optimization problem in (4-6) is that the constraint defined as $\Delta x(k+N) = 0$ guarantees that the state corresponding to $y(k+N)$ is a steady state. This is explained by the fact that setting $\Delta x(k+N) = 0$ at the end of the horizon means that this is a forced equilibrium, therefore keeping the same input keeps the system in a steady state and is feasible. With this, the recursive feasibility of the optimization problem can be achieved for a set of feasible initial conditions. Another conclusion that can be readily drawn is that the cost function is a decaying sequence due to the influence of the terminal cost, and thus the closed-loop is stable. In other words, $J_{k+1} \leq J_k$, which implies convergence of the cost function.

The following section illustrates the concepts discussed in this chapter through the results of a simulation of an illustrative example, specifically the forced Van der Poll oscillator, when controlled in closed-loop by the velocity-based qLPMC.

4-3 Illustrative example

To demonstrate how the velocity-based quasi-Linear Model Predictive Control (qLMPC) algorithm can be employed along with its benefits, an illustrative example such as the forced Van der Poll oscillator was used.

The dynamics of the forced Van der Poll oscillator can be described as:

$$\begin{aligned}
\ddot{x} &= \mu(1 - x^2)\dot{x} - x + u \\
y &= x,
\end{aligned} \tag{4-8}$$

with $\mu > 0$. This can be re-written as:

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} x_2 \\ -x_1 + \mu(1 - x_1^2)x_2 + u \end{bmatrix} \\
y &= x_1,
\end{aligned} \tag{4-9}$$

with $x = x_1$ and $\dot{x} = x_2$.

The following subsections present the results obtained from two different simulations of the forced Van der Poll oscillator. The first simulation uses the linearized model of the system around one equilibrium point. This model was obtained through Jacobian linearization and is then controlled using a simple state-feedback controller such as the Linear Quadratic Regulator (LQR). This first simulation aims to show the downsides of this approach such as the storage of the equilibrium or operating points. Also, the control of the linearized system is limited to the vicinity of the equilibrium point around which linearization was performed.

As for the second simulation, a quasi-LPV model of the forced Van der Poll oscillator is used which was obtained straightforwardly through velocity-based linearization. This system is then controlled using the velocity-based qLMPC framework discussed in this chapter.

4-3-1 State-feedback control with Jacobian linearization

In this simulation, the dynamics presented in (4-8) are linearized using Jacobian linearization around the equilibrium $x_{\text{eq}} = [x_{1,\text{eq}}, x_{2,\text{eq}}] = [0.5 \quad 0]^{\text{T}}$ and $u_{\text{eq}} = 0.5$. The linearized plant is then controlled using a Linear Quadratic Regulator (LQR). Note that the equilibrium points of this system can be obtained using the following:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 & 1 \\ -1 & \mu(1-x_1^2) \end{bmatrix} \begin{bmatrix} x_{1,\text{eq}} \\ x_{2,\text{eq}} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_{\text{eq}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} x_{1,\text{eq}} \\ x_{2,\text{eq}} \end{bmatrix} = \begin{bmatrix} u_{\text{eq}} \\ 0 \end{bmatrix} \tag{4-10}$$

When performing linearization around an equilibrium or operating point using Jacobian linearization, an affine representation of the nonlinear system is obtained as follows:

$$\begin{aligned} \begin{bmatrix} \dot{\delta}x_1 \\ \dot{\delta}x_2 \end{bmatrix} &= \left.\begin{bmatrix} 0 & 1 \\ -1-2x_1x_2\mu & \mu(1-x_1^2) \end{bmatrix}\right|_{\substack{(x_1,x_2)=(x_{1,\text{eq}},x_{2,\text{eq}}) \\ u=u_{\text{eq}}}} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} + \left.\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right|_{\substack{(x_1,x_2)=(x_{1,\text{eq}},x_{2,\text{eq}}) \\ u=u_{\text{eq}}}} \delta u \\ &= \begin{bmatrix} 0 & 1 \\ -1 & \mu(1-0.5^2) \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \delta u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \end{aligned} \tag{4-11}$$

with $\delta x_{\text{i}} = x_{\text{i}} - x_{\text{i,eq}}$ for $\text{i} = 1, 2$, and $\delta u = u - u_{\text{eq}}$.

It is important to realize the need to subtract the equilibrium point from the linearized dynamics of the plant. To better illustrate this concept, a simulation was run for both the case when the equilibrium point was not subtracted from the linearized dynamics of the plant and for when the model in (4-11) is used instead. The plant for both cases is discretized using the Fourth Order-Runge Kutta Method with a sampling time of $t_{\text{s}} = 0.1$ seconds. The reference for the output is set to $y_{\text{ref}} = 0.5$. Both plants are controlled in closed-loop using an LQR with the weighting matrices being defined as $Q = \text{diag}(80, 0)$ and $R = 0.1$. The initial condition at discrete-time $k = 0$ for the states is defined as $(x_1(0), x_2(0)) = (1.5, 0)$ and the simulation is performed for a total of 25 seconds.

As it can be noticed from the time-domain output response of the linearized model of the forced Van der Poll oscillator illustrated in Figure 4-1, when the equilibrium point is not subtracted from the

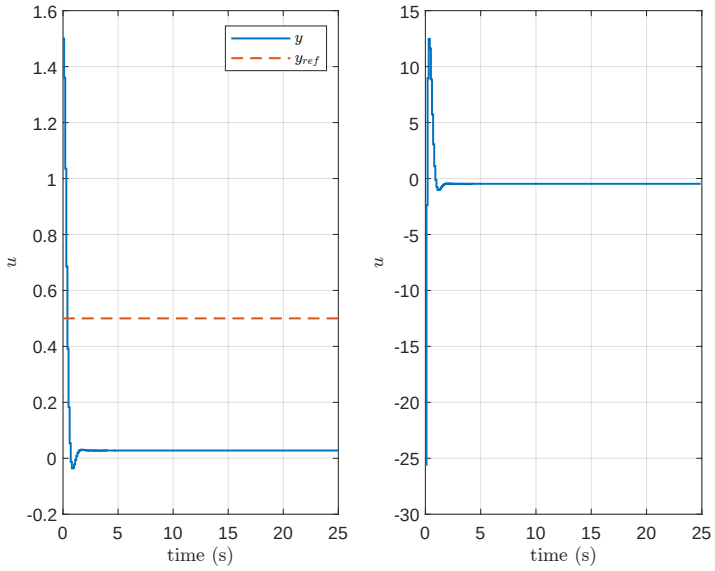

Figure 4-1: Left plot: Time-domain output response of the linearized model of the forced Van der Poll oscillator around the equilibrium $x_{\text{eq}} = [x_{1,\text{eq}}, x_{2,\text{eq}}] = [0.5 \quad 0]^{\text{T}}$ and $u_{\text{eq}} = 0.5$ using Jacobian linearization (blue solid line —) and respective reference (red dashed line - - -). In this case, the equilibrium point is not subtracted from the dynamics of the linearized mode. **Right plot**: Control input provided by the state-feedback LQR controller.

dynamics, the system is not capable of following the set reference, presenting a quite large steady-state error. On the other hand, when the model in (4-11) is used, the system follows its reference perfectly without any steady-state error - see Figure 4-2.

As a remark, it is relevant to understand that this was performed for only one equilibrium point. However, the same could be done for a set of equilibrium points which would provide several different Linear Time-Invariant (LTI) models of the forced Van der Poll oscillator. In this case, all the equilibrium points would have to be stored such that these could be subtracted from the respective LTI model obtained through Jacobian linearization.

It is worth noting that many real-life systems have actuation constraints that limit the control action. In such cases, the controller may not be able to make the system follow a reference or even stabilize the linearized system around an equilibrium point. This behavior is demonstrated in the time-domain output response of the linearized model of the forced Van der Poll oscillator, as seen in Figure Figure 4-3. When the control input is saturated to $u \in [-1.3, 1.3]$, the closed-loop system becomes unstable as the plant moves too far from the equilibrium point. Due to the actuation saturation, it cannot recover to the neighborhood of the equilibrium.

4-3-2 Velocity-based quasi-linear model predictive control

For this simulation, a quasi-Linear Parameter Varying (quasi-LPV) model of the forced Van der Poll oscillator is used. By performing velocity-based linearization as described in Section 4-1, the follow-

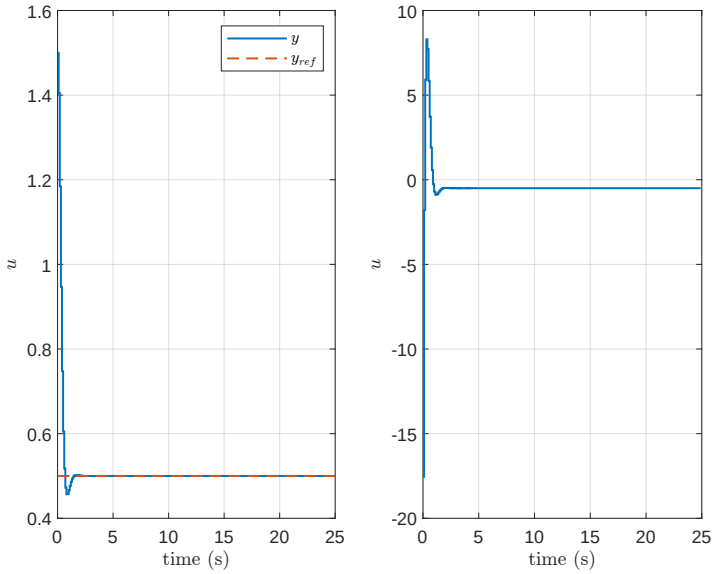

Figure 4-2: Left plot: Time-domain output response of the linearized model of the forced Van der Poll oscillator around the equilibrium $x_{\text{eq}} = [x_{1,\text{eq}}, x_{2,\text{eq}}] = [0.5 \quad 0]^{\mathsf{T}}$ and $u_{\text{eq}} = 0.5$ using Jacobian linearization (blue solid line —) and respective reference (red dashed line - - -). In this case, the equilibrium point is subtracted from the dynamics of the linearized mode. **Right plot**: Control input provided by the state-feedback LQR controller.

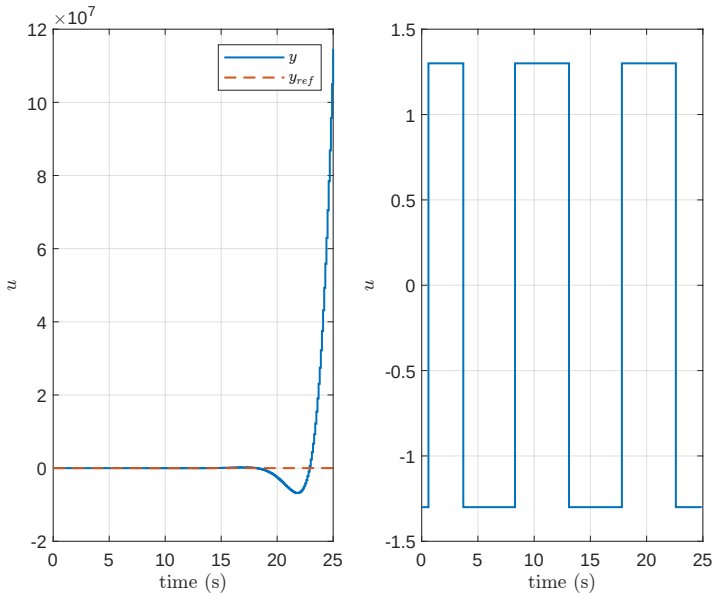

Figure 4-3: Left plot: Time-domain output response of the linearized model of the forced Van der Poll oscillator around the equilibrium $x_{\text{eq}} = [x_{1,\text{eq}}, x_{2,\text{eq}}] = [0.5 \quad 0]^{\mathsf{T}}$ and $u_{\text{eq}} = 0.5$ using Jacobian linearization (blue solid line —) and respective reference (red dashed line - - -). In this case, the equilibrium point is subtracted from the dynamics of the linearized mode and the time-domain output response of the linearized model of the forced Van der Poll oscillator becomes unstable when saturating the control input to $u \in [-1.3, 1.3]$. **Right plot**: Control input provided by the state-feedback LQR controller.

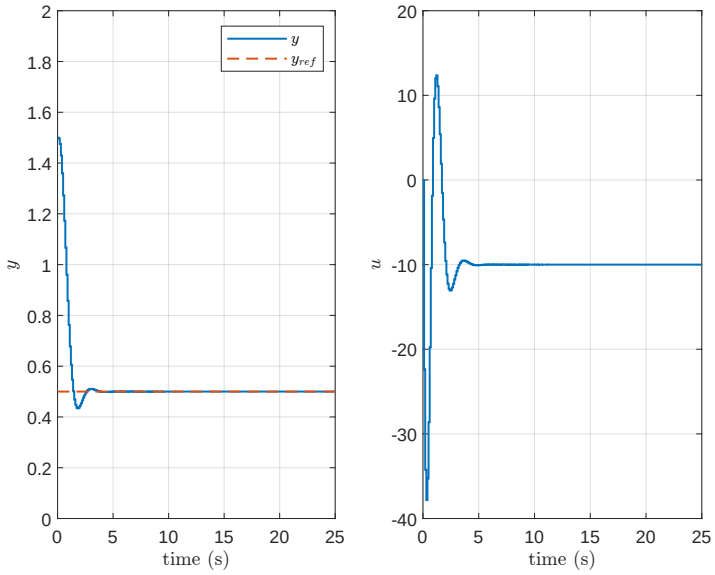

Figure 4-4: Left plot: Time-domain output response of the quasi-LPV model of the forced Van der Poll oscillator (blue solid line —) and respective reference (red dashed line - - -). **Right plot**: Control input provided by the velocity-based qLPMC algorithm.

ing quasi-LPV model can be defined:

$$\begin{bmatrix} \dot{y} \\ \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1-2\mu x_1 x_2 & \mu(1-x_1^2) \end{bmatrix} \begin{bmatrix} y \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{u}$$
$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + 0\dot{u}, \tag{4-12}$$

with the scheduling parameter $\theta = [x_1 \quad x_2]^{\mathrm{T}}$. Note that $x_1 = y$ and $x_2 = \dot{x_1}$, which are the first two elements of the extended state vector. This makes these variables recoverable from the states of the new quasi-LPV system presented in (4-12).

The state-space representation above is then discretized using the Fourth Order-Runge Kutta Method with a sampling time of $t_s = 0.1$s. The optimization problem used for the velocity-based qLMPC was the same as presented in (4-6). The reference for the output is set to $y_{\text{ref}} = 0.5$. Also, the weighting matrices are defined as $Q_1 = 80$, $Q_2 = \text{diag}(0, 0)$, $R = 0.1$ and $P = 10Q_1$, and the prediction horizon is set to $N = 20$. The simulation is performed for a total of 25 seconds. The results for the reference tracking problem can be seen in Figure 4-4, more precisely the system's output response and control action provided by the controller. Note that the control action presented here is the original u and not the increments Δu provided by the velocity-based algorithm. Furthermore, Figure 4-5 shows the number of iterations used in each run of the velocity-based quasi-Linear Model Predictive Control (qLMPC) algorithm and the time taken by each run to find the optimal scheduling sequence.

It can be understood that the system follows the set reference for the output with a settling time close to 3 seconds. As for the number of iterations for each run of the algorithm, it can be noticed that this decreases over time until only one iteration is needed to find the optimal value for the scheduling

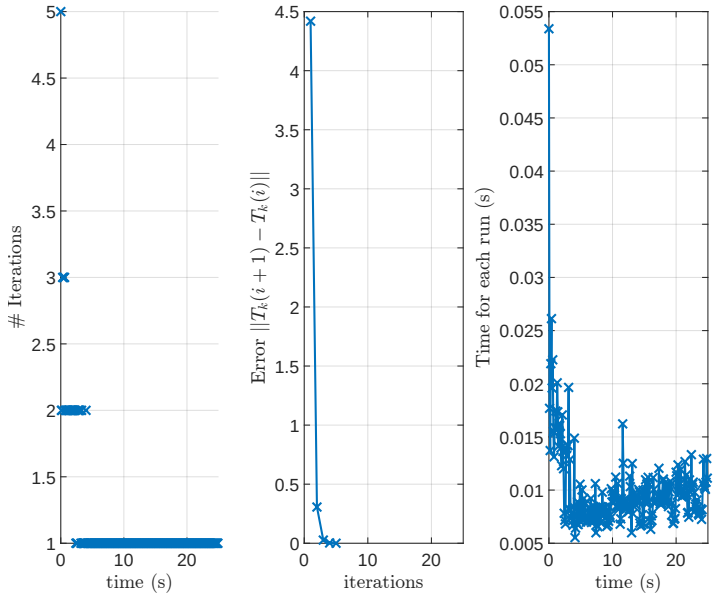

Figure 4-5: Left plot: Number of iterations l needed for the velocity-based qLMPC algorithm to converge. **Middle plot**: Number of iterations l needed for the velocity-based qLMPC algorithm to converge to an error of 10^{-3} for the first run $k = 0$. **Right plot**: Time needed for each run k of the velocity-based qLMPC algorithm. The computational time decreases with the number of iterations needed for the algorithm to converge.

Table 4-1: Average run time for a total number of 100 runs when using the velocity-based qLMPC algorithm to control the quasi-LPV model of the forced Van der Poll oscillator.

Algorithm	# of runs	Average run time (s)
Velocity-based qLMPC	100	4.5703

sequence. Accordingly, the time needed for each run also decreases. This is explained by the use of the so-called *warm-up* technique.

Finally, Table 4-1 shows the average run time for the entire 25-second simulation. It can be seen that for a total of 100 runs, the average run time was found to be around 4.57 seconds using an AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz processor, which makes this algorithm viable for online applications.

Another simulation was run with the same controller, however this time the control action is again saturated such that $u \in [-1.3, 1.3]$. The results for the tracking problem can be seen in Figure 4-6. As it can be noticed, the system does not reach the set reference as the control action is not "strong" enough to do so. However, the velocity-based qLMPC is still capable of stabilizing the system, contrary to what was seen previously with the linearized model around a single equilibrium point controlled by a simple state-feedback LQR.

As a final note, it should be noted that unlike the findings presented in [53], this method does not require an offline solution of a set of LMI conditions to ensure the feasibility and stability of the MPC problem for quasi-LPV systems. Instead, the original states' derivatives are controlled to converge to zero at the end of the horizon, leading the system to a steady state, as mentioned earlier. Furthermore, since the model obtained after performing velocity-based linearization has intrinsic integral action

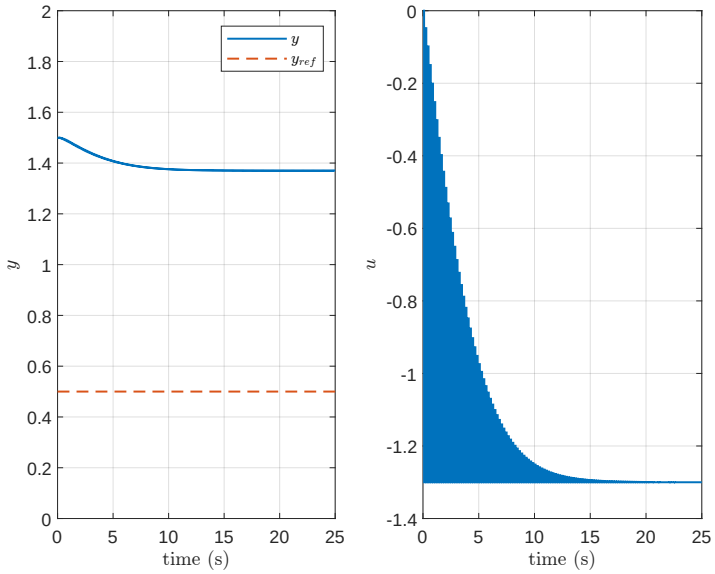

Figure 4-6: Left plot: Time-domain output response of the quasi-LPV model of the forced Van der Poll oscillator (blue solid line —) and respective reference (red dashed line - - -). The time-domain output response of the quasi-LPV model of the forced Van der Poll oscillator cannot reach the set reference when saturating the control input to $u \in [-1.3, 1.3]$. However, the velocity-based qLPMC algorithm is capable of stabilizing the closed-loop system, reaching a steady state. **Right plot**: Control input provided by the velocity-based qLPMC algorithm.

as discussed in Section 4-1, the system can reach the reference with zero steady-state error without the need to save or consider equilibrium or operating points if the control input is unrestricted. However, if the control input saturates, the system may not reach the set reference since the control action may not be "strong" enough to achieve it.

The following chapter illustrates how the velocity-based qLMPC algorithm can be used in wind turbine control when subject to side-side periodic loads.

4-4 Summary

The main concepts covered in this chapter are listed below:

- This chapter discusses the advantages of velocity-based linearization over ad-hoc Linear Parameter-Varying (LPV) parameterization and Jacobian linearization in obtaining quasi-LPV models. Velocity-based linearization simplifies calculations without requiring expert intuition. Also, contrary to velocity-based linearization, Jacobian linearization can lead to errors and inconvenient affine representations. Furthermore, velocity-based linearization produces an exact and linear quasi-LPV model, eliminating the need to store equilibrium points;

- The augmented velocity-form state-dependent linear discrete-time model presented in this chapter has built-in integral action. Integral action facilitates offset-free tracking of constant reference signals, contributing to improved control performance;

- However, even though disturbance estimators are not needed when using the augmented velocity-form state-dependent linear discrete-time model, observers are still often necessary to estimate the unknown or unmeasurable states;
- In velocity space, equilibrium points are mapped to the origin, simplifying computations and eliminating the need for parameterization of the equilibria or operating points and offline computation of terminal sets. Therefore, stability can be guaranteed without complex offline computations since there is no need to solve Linear Matrix Inequalities (LMIs) to get a terminal set.

Chapter 5

Application in wind turbine control

This chapter presents an interesting development for mitigating undesirable side-side periodic loads while minimizing the impact on the power output. Previously, a velocity quasi-Linear Model Predictive Control (qLMPC) algorithm was introduced in [27] for reducing tower fore-aft acceleration. The quasi-Linear Parameter-Varying (quasi-LPV) framework used in this study accommodates modelling faults and uncertainties, such as blade or rotor weight imbalances that cause side-side motions in soft tower configurations, as explored in [15]. However, unlike the algorithm in [15], this algorithm linearizes the nonlinear model of the system using a velocity-based linearization technique. The authors suggest that this eliminates the need for computing and storing input and state operating point vectors, in contrast to the method used in [15], which requires several operating points for which Jacobian linearization is performed.

It should be noted that velocity-based linearization, discussed in Section 4-1, leads to a precise representation of the nonlinear dynamics of the original system, facilitating stability and offset-free control. Additionally, velocity algorithms have integral action for offset-free control, without requiring disturbance estimators, as previously mentioned.

The proposed framework presented in this chapter combines the approaches presented in [15] and [27]. The wind turbine tower dynamics are demodulated using a Model Demodulation Transformation (MDT) technique, and the real and imaginary parts of the tower top displacement signal are used as outputs of the system. In an ideal scenario, the aim is to bring these outputs to zero, meaning that the tower top would stand still and negate the side-side periodic force. The model also takes into account the wind turbine aerodynamics. Here, the nonlinear model is linearized using velocity-based linearization and controlled using a velocity-based qLMPC controller. The cost function is defined to balance minimum impact in the power output with minimizing disturbances originating from the side-side periodic loads.

5-1 Nominal tower and wind turbine aerodynamic models

This section introduces the nominal tower and wind turbine aerodynamic models used as the basis for the framework developed in this thesis.

5-1-1 Tower nominal model

The nominal tower dynamics are given in [15] as the following second-order model:

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}-d/m & -k/m\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}1/m\\ 0\end{bmatrix} F_{\text{sd}}(t), \tag{5-1}$$

with $\{m, d, k\} \in \mathbb{R}^+$ being the constant first modal mass, modal damping, and modal stiffness. The states $x_1(t)$ and $x_2(t)$ represent the tower's top velocity and displacement, respectively. Here, $F_{\text{sd}}(t)$ is the side-side force modelled as:

$$F_{\text{sd}}(t) = a_{\text{u}}(\tau)\cos(\psi(t)) + cT_{\text{g}}, \tag{5-2}$$

with $a_{\text{u}}(\tau) \in \mathbb{R}^+$ being the amplitude of the excitation, cT_{g} the contribution of T_{g} in the form of a static contribution to the tower dynamics, $c = 3/2H$ the ratio between the angular and translational displacement of the tower motion [64], H the tower height, and $\psi(t) \in [0, 2\pi)$ the rotor azimuth angle. Note that $F_{\text{sd}}(t)$ is considered to be a generic force in the work of [28].

5-1-2 Wind turbine aerodynamic model

Another definition that will be needed later in this section is the representation of the complete wind turbine aerodynamics:

$$\dot{\omega}_{\text{r}} = \tfrac{T_{\text{a}} - GT_{\text{g}}}{J_{\text{r}}}, \tag{5-3}$$

$$T_{\text{a}} = \tfrac{1}{2}\rho\pi R^3 V^2 C_{\text{q}}(\lambda, \beta), \tag{5-4}$$

$$T_{\text{g}} = \tfrac{\pi\rho R^5 C_{\text{p}}(\lambda,\beta)\omega_{\text{r}}^2}{2G\lambda^3} = \tfrac{K}{G}\omega_{\text{r}}^2, \tag{5-5}$$

with $T_{\text{a}} \in \mathbb{R}$ being the aerodynamic rotor torque, $T_{\text{g}} \in \mathbb{R}^+$ the generator torque, $G \geq 1$ the gearbox ratio, and $J_{\text{r}} \in \mathbb{R}^+$ the total rotor inertia. The optimal mode gain K is defined as:

$$K = \tfrac{\pi\rho R^5 C_{\text{p}}(\lambda,\beta)}{2\lambda^3} \in \mathbb{R}^+. \tag{5-6}$$

Furthermore, $\rho \in \mathbb{R}^+$ is the air density, $R \in \mathbb{R}^+$ the rotor radius, $V \in \mathbb{R}^+$ the wind speed, ω_{r} the rotor speed, $C_{\text{q}} \in \mathbb{R}$ the aerodynamic torque coefficient, and C_{p} the power coefficient as a function of the tip-speed ratio $\lambda = \omega_{\text{r}}R/V$ and the pitch angle β.

5-2 Tower demodulated model

This section presents the concept of Model Demodulation Transformation (MDT). The MDT technique offers the advantage of separating the fast- and slow-varying components of states when representing a system in the standard state-space form. This separation proves particularly valuable in the design of model-based controllers, as it effectively isolates the transient and steady-state aspects of the system's behaviour. Notably, this technique was initially employed in [65] to address challenges in deriving a transient dynamic model for cantilevers in tapping mode atomic force microscopy.

In the wind energy sector, the MTD technique has demonstrated significant utility in modelling the aerodynamics of wind turbines and the tower's response to side-side periodic loads as a quasi-Linear Parameter-Varying model (quasi-LPV). This approach generates a quasi-steady signal for the tower top displacement rather than a periodic one [15]. Hence, this transformation enables the formulation of a convex optimization problem to mitigate side-side periodic loading using predictive controllers such as (velocity-based) quasi-Linear Model Predictive Control (qLMPC) addressed in Chapters 3 and 4.

5-2-1 Model demodulation transformation

As mentioned in [15], when a linear system is subjected to a periodic input, it responds with a periodic output at the same frequency but with a certain phase shift and magnitude relative to the input. Let the periodic signal be represented as a cosine wave as follows:

$$x_{\mathrm{i}}(t) = a_{\mathrm{i}}(\tau)\cos(\omega_{\mathrm{r}} t + \phi(\tau)). \tag{5-7}$$

Given that the amplitude and phase exhibit a significantly slower rate of change, a new time coordinate is introduced, denoted as τ. This parameter will serve as the descriptor for any functions that change slowly. As explained in [15, 65], a function $f(\tau)$ is called slow when comparing to a rapidly varying periodic function $g(t)$ with period T if the following property holds:

$$\int_0^{\top} f(\tau)g(t)dt \approx f(\tau)\int_0^{\top} g(t)dt. \tag{5-8}$$

It is relevant to mention that τ is not different from t, as it is merely introduced to distinguish slowly evolving functions from those that exhibit rapid changes. Then, the signal is re-written using Euler's formula:

$$x_{\mathrm{i}}(t) = \mathfrak{Re}\{a_{\mathrm{i}}(\tau)e^{j(\omega_{\mathrm{r}} t + \phi(\tau))}\}, \tag{5-9}$$

with $j = \sqrt{-1}$ being the imaginary unit, and $\mathfrak{Re}\{\cdot\}$ and $\mathfrak{Im}\{\cdot\}$ the real and imaginary components, respectively. The slow-varying term can be separated from the fast-varying term as follows:

$$x_{\mathrm{i}}(t) = \mathfrak{Re}\{\underbrace{a_{\mathrm{i}}(\tau)e^{j\phi(\tau)}}_{X_{\mathrm{i}}(\tau)} e^{j\omega_{\mathrm{r}} t}\}, \tag{5-10}$$

with $x_{\mathrm{i}}(\tau) \in \mathbb{C}$. The amplitude and phase can be reconstructed using the following relations:

$$\begin{aligned} a_{\mathrm{i}}(\tau) &= \sqrt{\mathfrak{Re}\{X_{\mathrm{i}}(\tau)\}^2 + \mathfrak{Im}\{X_{\mathrm{i}}(\tau)\}^2}, \\ \phi_i(\tau) &= \arctan\left(\frac{\mathfrak{Im}\{X_{\mathrm{i}}(\tau)\}}{\mathfrak{Re}\{X_{\mathrm{i}}(\tau)\}}\right). \end{aligned} \tag{5-11}$$

Taking the first-time derivative gives

$$\dot{x}_{\mathrm{i}}(t) = \mathfrak{Re}\left\{(\dot{X}_{\mathrm{i}}(\tau) + j\omega_{\mathrm{r}} X_{\mathrm{i}}(\tau))e^{j\omega_{\mathrm{r}} t}\right\}. \tag{5-12}$$

Having these insights into the MDT technique, the following subsection provides the steps to obtain the wind turbine aerodynamics and tower demodulated model.

5-2-2 Wind turbine aerodynamics and tower demodulated model

Assuming a constant rotor's angular velocity over one revolution, the following holds:

$$\psi(t) = \int \omega_{\mathrm{r}}(t)dt \approx \omega_{\mathrm{r}} t, \tag{5-13}$$

where $\omega_{\mathrm{r}}(t)$ is the rotor's angular velocity, which determines the frequency of the periodic excitation [28]. The side-side force is considered a semi-periodic signal, which means a harmonic signal with an amplitude and phase that slowly varies. This signal can be broken down into its semi-harmonic components and rewritten as follows:

$$F_{\mathrm{sd}}(t) = \mathfrak{Re}\left\{a_{\mathrm{u}}(\tau)e^{j\omega_{\mathrm{r}}t} + c\sum_{n=0}^{\infty} T_{\mathrm{g}}^{(n)}(\tau)e^{jn\omega_{\mathrm{r}}t}\right\} \tag{5-14}$$

with $T_{\mathrm{g}}^{(n)}(\tau) \in \mathbb{C}$ being the amplitude and phase of the n-th harmonic component of the generator torque signal. Applying (5-10), (5-12) and (5-14) to the state-space in (5-1) provides the following:

$$\begin{aligned}
&\mathfrak{Re}\left\{\left(\dot{X}_1(\tau) + j\omega_{\mathrm{r}}X_1(\tau) + \tfrac{d}{m}X_1(\tau) + \tfrac{k}{m}X_2(\tau) - \tfrac{1}{m}a_{\mathrm{u}}(\tau)\right)e^{j\omega_{\mathrm{r}}t}\right. \\
&\left.-\tfrac{c}{m}\sum_{n=0}^{\infty} T_{\mathrm{g}}^{(n)}(\tau)e^{jn\omega_{\mathrm{r}}t}\right\} = 0, \qquad (5\text{-}15)\\
&\mathfrak{Re}\left\{\left(\dot{X}_2(\tau) + j\omega_{\mathrm{r}}X_2(\tau) - X_1(\tau)\right)e^{j\omega_{\mathrm{r}}t}\right\} = 0. \qquad (5\text{-}16)
\end{aligned}$$

By multiplying Equations (5-15) and (5-16) by $e^{j\omega_{\mathrm{r}}t}$ following by integration over an oscillation period gives:

$$\begin{aligned}
&\int_0^{T_r} \mathfrak{Re}\left\{\left(\dot{X}_1(\tau) + j\omega_{\mathrm{r}}X_1(\tau) + \tfrac{d}{m}X_1(\tau) + \tfrac{k}{m}X_2(\tau) - \tfrac{1}{m}a_{\mathrm{u}}(\tau)\right)e^{j\omega_{\mathrm{r}}t}\right. \\
&\left.-\tfrac{c}{m}\sum_{n=0}^{\infty} T_{\mathrm{g}}^{(n)}(\tau)e^{jn\omega_{\mathrm{r}}t}\right\} = 0, \qquad (5\text{-}17)\\
&\int_0^{T_r} \mathfrak{Re}\left\{\left(\dot{X}_2(\tau) + j\omega_{\mathrm{r}}X_2(\tau) - X_1(\tau)\right)e^{j\omega_{\mathrm{r}}t}\right\} e^{j\omega_{\mathrm{r}}t}dt = 0. \qquad (5\text{-}18)
\end{aligned}$$

Note that by applying the result from (5-8), the following is given:

$$\begin{bmatrix}\dot{X}_1(\tau)\\ \dot{X}_2(\tau)\end{bmatrix} = \begin{bmatrix}-j\omega_{\mathrm{r}} - \frac{d}{m} & -\frac{k}{m}\\ 1 & -j\omega_{\mathrm{r}}\end{bmatrix}\begin{bmatrix}X_1(\tau)\\ X_2(\tau)\end{bmatrix} + \begin{bmatrix}\frac{c}{m}\\ 0\end{bmatrix}T_{\mathrm{g}}^{(1)} + \begin{bmatrix}\frac{1}{m}\\ 0\end{bmatrix}a_{\mathrm{u}}(\tau). \tag{5-19}$$

As a remark, it is important to realize that only the first harmonic of $T_{\mathrm{g}}^{(n)}$ is taken into account, denoted as $T_{\mathrm{g}}^{(1)}$. This is due to the following orthogonality property of harmonic functions:

$$\int_0^{2\pi} \mathfrak{Re}\left\{\gamma e^{jn\theta}\right\} e^{j\theta}d\theta = 0, \forall n \neq 1, \tag{5-20}$$

which leads to the harmonics corresponding to zeroth and higher harmonics being cancelled, as mentioned in [28].

Defining the state vector $q = [q_1, q_2, q_3, q_4]^\top = [\mathfrak{Re}\{X_1\}, \mathfrak{Im}\{X_1\}, \mathfrak{Re}\{X_2\}, \mathfrak{Im}\{X_2\}]^\top \in \mathbb{R}^{4\times 1}$, the input vector $u = [\mathfrak{Re}\{T_{\mathrm{g}}^{(1)}\}, \mathfrak{Im}\{T_{\mathrm{g}}^{(1)}\}]$, and adding the representation of the complete aerodynamics $\dot{\omega}_{\mathrm{r}}$ as a step to obtain a quasi-LPV model, the system can be re-defined as the following set

of differential equations:

$$\dot{q}_1 = f_1 = -\tfrac{d}{m}q_1 + \omega_{\mathrm{r}}q_2 - \tfrac{k}{m}q_3 + \tfrac{c}{m}\mathfrak{Re}\{T_{\mathrm{g}}^{(1)}\} + \tfrac{a_{\mathrm{u}}(\tau)}{m}, \tag{5-21}$$

$$\dot{q}_2 = f_2 = -\omega_{\mathrm{r}}q_1 - \tfrac{d}{m}q_2 - \tfrac{k}{m}q_4 + \tfrac{c}{m}\mathfrak{Im}\{T_{\mathrm{g}}^{(1)}\}, \tag{5-22}$$

$$\dot{q}_3 = f_3 = q_1 + \omega_{\mathrm{r}}q_4, \tag{5-23}$$

$$\dot{q}_4 = f_4 = q_2 - \omega_{\mathrm{r}}q_3, \tag{5-24}$$

$$\dot{\omega}_{\mathrm{r}} = f_5 = \tfrac{T_{\mathrm{a}} - G(T_{\mathrm{g}} + T_{\mathrm{g}}^{(1)})}{J_{\mathrm{r}}}. \tag{5-25}$$

Note that

$$\begin{aligned} T_{\mathrm{g}}^{(1)} &= \sqrt{\mathfrak{Re}\{T_{\mathrm{g}}^{(1)}\}^2 + \mathfrak{Im}\{T_{\mathrm{g}}^{(1)}\}^2}\cos\left(\omega_{\mathrm{r}}t + \arctan\left(\tfrac{\mathfrak{Im}\{T_{\mathrm{g}}^{(1)}\}}{\mathfrak{Re}\{T_{\mathrm{g}}^{(1)}\}}\right)\right) \\ &\approx \mathfrak{Re}\{T_{\mathrm{g}}^{(1)}\}\cos(\omega_{\mathrm{r}}t) + \mathfrak{Im}\{T_{\mathrm{g}}^{(1)}\}\sin(\omega_{\mathrm{r}}t). \end{aligned} \tag{5-26}$$

The instantaneous amplitude and phase of the dynamic system response at frequency ω_{r} are given as follows:

$$a_2 = \sqrt{q_3^2 + q_4^2}, \tag{5-27}$$

$$\phi_2 = \arctan\left(\tfrac{q_4}{q_3}\right). \tag{5-28}$$

The next subsection presents an illustrative example to better understand the working principles and characteristics of the derived MDT model.

5-2-3 Illustrative example

Figure 5-1 is useful to illustrate the working principles and characteristics of the derived MDT model. A chirp signal is set as the input for the nominal model of the wind turbine tower with frequency varying linearly from 0 Hz to 0.191 Hz, which is equivalent to a linearly varying rotational speed between 0 rad/s and 1.2 rad/s, as also presented in Figure 5-1. The amplitude of the chirp signal is 1 N and is used to define the tower input signal corresponding to the side-side periodic force F_{sd}. The results of the simulation are displayed in the last plot of Figure 5-1 in red. In addition, the nonlinear dynamics of the demodulated model were also simulated for a linearly varying rotational speed, which can be seen as the blue line in the same plot of Figure 5-1. It can be understood that the demodulated model accurately tracks the amplitude of the signal obtained when using the nominal model, leading to a quasi-steady state signal. This is relevant for the use of qLMPC as a control algorithm, as the optimization problem needs to be convex.

Note that the model represented by (5-21)-(5-25) is not yet linear. Additionally, the output is a non-linear combination of the state vector elements. To use the qLMPC algorithm to obtain the optimal torque control input, the model must first be linearized to obtain the quasi-Linear Parameter-Varying (quasi-LPV) model. In [15], the Jacobian linearization technique is used by performing linearization around an operating point. However, there are alternative linearization techniques available, such as the velocity-based linearization method presented in [27].

Finally, after linearizing the model to obtain a quasi-LPV representation, the system is discretized using the forward Euler method. The linearized and discretized model is then used in the velocity-based qLMPC algorithm.

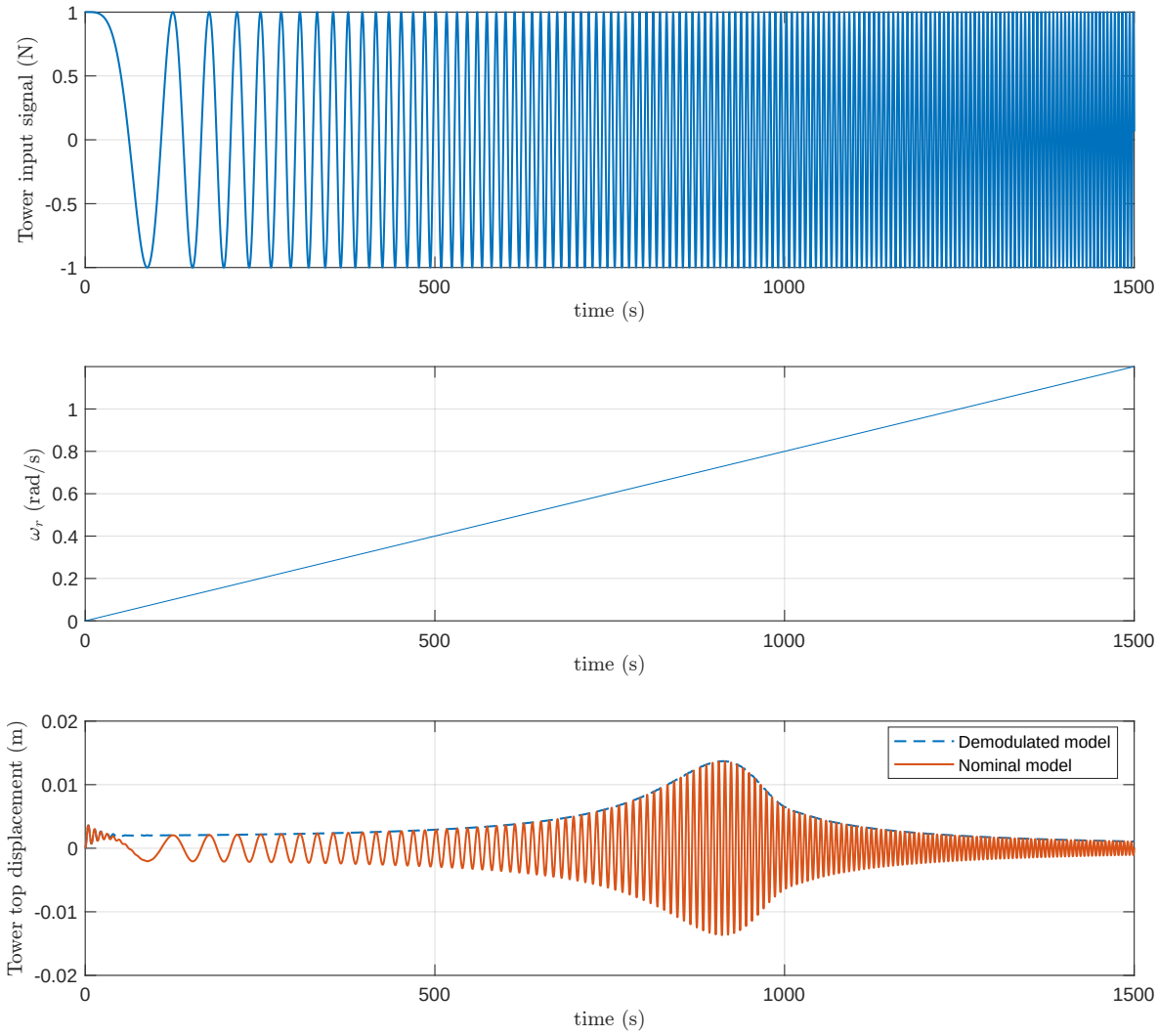

Figure 5-1: Top plot: Chirp signal set as the input for the nominal model of the wind turbine tower with frequency varying linearly from 0 Hz to 0.191 Hz. The amplitude of the chirp signal is 1 N and is used to define the tower input signal corresponding to the side-side periodic force F_{sd}. **Bottom left plot**: Rotational speed linearly ramping up from 0 rad/s to 1.2 rad/s. **Bottom right plot**: Time-domain response of the nominal model of the wind turbine tower subject to the defined chirp signal as input (red solid line —). Time-domain response of the demodulated wind turbine tower model (blue dashed line - - -). It can be understood that the demodulated model accurately tracks the amplitude of the signal obtained when using the nominal model, leading to a quasi-steady state signal.

5-3 Velocity-based qLMPC for side-side periodic load mitigation

This section mathematically formalizes the velocity-based quasi-Linear Model Predictive Control (qLMPC) used for side-side periodic load mitigation. As previously mentioned in Chapter 4, velocity-based linearization is employed to derive quasi-Linear Parameter-Varying (quasi-LPV) models directly from the original nonlinear system. Hence, this approach enables the linearization of wind turbine aerodynamics and the tower demodulated model into a quasi-LPV model. This quasi-LPV model can subsequently be used in the velocity-based qLMPC algorithm, which is designed to actively cancel or mitigate the periodic side-side loads on the wind turbine tower.

5-3-1 Velocity-based linearization of the wind turbine aerodynamics and tower demodulated model

In this subsection, a quasi-LPV model of the wind turbine demodulated tower model and aerodynamics is derived through the use of velocity-based linearization.

First, the model presented in (5-21)-(5-25) is linearized using the velocity-based linearization technique described in (4-2). Also, it is assumed that the wind speed and the disturbance change only at the beginning of each optimization calculation. Hence, by defining:

$$x = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ \omega_\mathrm{r} \\ V \\ a_\mathrm{u} \end{bmatrix} \in \mathbb{R}^{7\times 1}, u = \begin{bmatrix} \mathfrak{Re}\{{T_\mathrm{g}}^{(1)}\} \\ \mathfrak{Im}\{{T_\mathrm{g}}^{(1)}\} \end{bmatrix} \in \mathbb{R}^{2\times 1}, \dot{x} = f(x,u) = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{\omega}_\mathrm{r} \\ \dot{V} \\ \dot{a}_\mathrm{u} \end{bmatrix} \in \mathbb{R}^{7\times 1}, y = h(x,u) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ \omega_\mathrm{r} \end{bmatrix} \in \mathbb{R}^{5\times 1}, \tag{5-29}$$

the following can be derived:

$$\begin{aligned} \ddot{x} &= \nabla_x f(x,u)\dot{x} + \nabla_u f(x,u)\dot{u} \\ &= \underbrace{\begin{bmatrix} -\frac{d}{m} & \omega_\mathrm{r} & -\frac{k}{m} & 0 & q_2 & 0 & \frac{1}{m} \\ -\omega_\mathrm{r} & -\frac{d}{m} & 0 & -\frac{k}{m} & -q_1 & 0 & 0 \\ 1 & 0 & 0 & \omega_\mathrm{r} & q_4 & 0 & 0 \\ 0 & 1 & -\omega_\mathrm{r} & 0 & -q_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial \dot{\omega}_\mathrm{r}}{\partial \omega_\mathrm{r}} & \frac{\partial \dot{\omega}_\mathrm{r}}{\partial V} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_\mathrm{c}(x,t)} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{\omega}_\mathrm{r} \\ \dot{V} \\ \dot{a}_\mathrm{u} \end{bmatrix}}_{\dot{x}} + \underbrace{\begin{bmatrix} \frac{c}{m} & 0 \\ 0 & \frac{c}{m} \\ 0 & 0 \\ 0 & 0 \\ \frac{\partial \dot{\omega}_\mathrm{r}}{\partial \mathfrak{Re}\{T_\mathrm{g}^{(1)}\}} & \frac{\partial \dot{\omega}_\mathrm{r}}{\partial \mathfrak{Im}\{T_\mathrm{g}^{(1)}\}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{B_\mathrm{c}(x,t)} \underbrace{\begin{bmatrix} \mathfrak{Re}\{\dot{T}_\mathrm{g}^{(1)}\} \\ \mathfrak{Im}\{\dot{T}_\mathrm{g}^{(1)}\} \end{bmatrix}}_{\dot{u}}, \end{aligned}$$

$$\dot{y} = \nabla_x h(x,u)\dot{x}$$
$$= \underbrace{\begin{bmatrix} I_{5\times5} & 0_{5\times2} \end{bmatrix}}_{C_\mathrm{c}} \underbrace{\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{\omega}_\mathrm{r} \\ \dot{V} \\ \dot{a}_\mathrm{u} \end{bmatrix}}_{\dot{x}}, \tag{5-30}$$

with

$$\frac{\partial \dot{\omega}_\mathrm{r}}{\partial \omega_\mathrm{r}} = \frac{\rho\pi R^4}{2J_\mathrm{r}} \frac{\partial C_\mathrm{q}(\lambda,\beta)}{\partial \lambda} V - \frac{2K}{J_\mathrm{r}}\omega_\mathrm{r} - \frac{G}{J_\mathrm{r}}(-\mathfrak{Re}\{T_\mathrm{g}^{(1)}\}\sin(\omega_\mathrm{r}t)t + \mathfrak{Im}\{T_\mathrm{g}^{(1)}\}\cos(\omega_\mathrm{r}t)t), \tag{5-31}$$

$$\frac{\partial \dot{\omega}_\mathrm{r}}{\partial V} = \frac{\rho\pi R^3 C_\mathrm{q}(\lambda,\beta)}{J_\mathrm{r}} V - \frac{\rho\pi R^4}{2J_\mathrm{r}} \frac{\partial C_\mathrm{q}(\lambda,\beta)}{\partial \lambda}\omega_\mathrm{r}, \tag{5-32}$$

$$\frac{\partial \dot{\omega}_\mathrm{r}}{\partial \mathfrak{Re}\{T_\mathrm{g}^{(1)}\}} = -\frac{G}{J_\mathrm{r}}\cos(\omega_\mathrm{r}t), \tag{5-33}$$

$$\frac{\partial \dot{\omega}_\mathrm{r}}{\partial \mathfrak{Im}\{T_\mathrm{g}^{(1)}\}} = -\frac{G}{J_\mathrm{r}}\sin(\omega_\mathrm{r}t), \tag{5-34}$$

and where $A_\mathrm{c}(x,u) \in \mathbb{R}^{7\times7}$ and $B_\mathrm{c}(x,u) \in \mathbb{R}^{7\times2}$ are the state and input matrices, respectively, and $C_\mathrm{c} \in \mathbb{R}^{5\times7}$ is the output matrix.

Having this, a velocity-form state-dependent linear model can be defined as:

$$\underbrace{\begin{bmatrix} \dot{y} \\ \ddot{x} \end{bmatrix}}_{\dot{x}_\mathrm{ext}} = \underbrace{\begin{bmatrix} 0 & C_\mathrm{c} \\ 0 & A_\mathrm{c}(x,u) \end{bmatrix}}_{A_\mathrm{c,ext}(x,u)} \underbrace{\begin{bmatrix} y \\ \dot{x} \end{bmatrix}}_{x_\mathrm{ext}} + \underbrace{\begin{bmatrix} 0 \\ B_\mathrm{c}(x,u) \end{bmatrix}}_{B_\mathrm{c,ext}(x,u)} \dot{u},$$
$$y = \underbrace{\begin{bmatrix} I_{5\times5} & 0_{5\times7} \end{bmatrix}}_{C_\mathrm{c,ext}} \underbrace{\begin{bmatrix} y \\ \dot{x} \end{bmatrix}}_{x_\mathrm{ext}}. \tag{5-35}$$

Note that here the output $y \in \mathbb{R}^{5\times1}$ is chosen to be $\{q_1, q_2, q_3, q_4, \omega_\mathrm{r}\}$. Also, the state and input matrices depend on those parameters and the wind speed. Therefore, it can be understood that this can be written as a quasi-LPV model by setting:

$$\theta = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad \omega_\mathrm{r} \quad V]^\top \in \mathbb{R}^{6\times1}. \tag{5-36}$$

The continuous state-space presented in (5-35) is then discretized using the forward Euler method. Thus, the discrete-time matrices $A_\mathrm{d,ext}(\theta)$, and $B_\mathrm{d,ext}(\theta)$, can be defined as:

$$A_\mathrm{d,ext}(\theta) = I + A_\mathrm{c,ext}(\theta)t_\mathrm{s} \in \mathbb{R}^{12\times12}, \tag{5-37}$$

$$B_\mathrm{d,ext}(\theta) = B_\mathrm{c,ext}(\theta)t_\mathrm{s} \in \mathbb{R}^{12\times2}. \tag{5-38}$$

The matrices from the output are equal in continuous- and discrete-time, thus:

$$C_\mathrm{d,ext} = C_\mathrm{c,ext} \in \mathbb{R}^{5\times12}, \tag{5-39}$$

$$D_\mathrm{d,ext} = D_\mathrm{c,ext} = 0_{5\times2}. \tag{5-40}$$

Also, it is relevant to realize that t can be defined in discrete-time as $t = kt_{\text{s}}$. Therefore, the following discrete-time state-space formulation is then used for the rest of this project, when assuming that the states and disturbances can be measured or known:

$$\begin{aligned} \mathbf{x}_{\text{ext}}(k+1) &= A_{\text{d,ext}}(\theta(k))\mathbf{x}_{\text{ext}}(k) + B_{\text{d,ext}}(\theta(k))\Delta\mathbf{u}(k) \\ \mathbf{y}(k) &= C_{\text{d,ext}}\mathbf{x}_{\text{ext}}(k), \end{aligned} \tag{5-41}$$

with $\mathbf{x}_{\text{ext}}(k) = \begin{bmatrix} y(k) & \Delta x(k) \end{bmatrix}^\top$, $\Delta\mathbf{u}(k) = u(k) - u(k-1)$, and the subscript d in the state, input, and output matrices, $A_{\text{d,ext}}(\theta(k))$, $B_{\text{d,ext}}(\theta(k))$, and $C_{\text{d,ext}}$, respectively, indicating discrete-time.

5-3-2 Velocity-based qLMPC algorithm

The optimization problem at hand is defined earlier in (4-6) minimizing the cost in (4-7), and applied to the derived model in the previous subsection. Here, the variable $\mathbf{y}$ represents the output, which in this case is $\{q_1, q_2.q_3, q_4, \omega_{\text{r}}\}$, while $\Delta\mathbf{u}$ represents the control input that corresponds to the real and imaginary components of the first harmonic of $T_{\text{g}}^{(n)}$. The weight matrices Q_1 and R are used to act on the predicted tower-top displacement and velocity amplitudes and deviation from the optimal control signal, respectively.

It is important to note that the $||\Delta\mathbf{u}||_R^2$ term of the cost function assumes optimal power production efficiency using the K-omega-squared torque control strategy. Furthermore, it can be deduced that $||\mathbf{y} - \mathbf{y}_{\text{ref}}||_{Q_1}^2$ aims to minimize periodic fatigue loads, whereas $||\Delta\mathbf{u}||_R^2$ is a combination of impact on energy production minimization and penalization on the control input. By formulating the objective in this way, it is possible to achieve a convenient trade-off between the influence in the power production and load reductions by varying the weight ratio of the matrices R and Q_1.

The following section focuses on the results and discussion of five simulation cases performed with the National Renewable Energy Laboratory (NREL) 5-MW reference wind turbine model.

5-4 Summary

In this chapter the following topics were addressed:

- The wind turbine tower dynamics can be demodulated using a Model Demodulation Transformation (MDT) technique. This way, the amplitude of the tower top displacement is used as the output of the system. The aim is to bring this output to zero, meaning that the tower top would stand still and mitigating the effect of the side-side periodic force;
- The model also takes into account the wind turbine aerodynamics. Here, the nonlinear model is linearized using velocity-based linearization and controlled using a velocity-based quasi-Linear Model Predictive Control (qLMPC) controller. The cost function is defined to balance minimum impact in the power output with minimizing disturbances originating from the side-side periodic loads.

Chapter 6

Simulation results

This chapter presents the results and respective discussion of five simulation cases designed to evaluate the performance and efficiency of the proposed framework. It provides general details of the simulations for each case, along with the parameters for linearization and simulation performed with the National Renewable Energy Laboratory (NREL) 5-MW reference wind turbine.

In Figure 6-1 the yellow block represents the quasi-Linear Parameter-Varying (quasi-LPV) model and the velocity-based quasi-Linear Model Predictive Control (qLMPC) controller used for mitigating the side-side periodic load on the tower. The focus of this thesis is primarily on the components within the yellow block. To demonstrate the working principles and effectiveness of the proposed scheme, four "artificial" simulation cases related to the yellow block are defined. The green block depicts the nonlinear wind turbine model, which includes rotor aerodynamics and a demodulated nominal tower model. Additionally, the generator torque controller is integrated into the control scheme along with the designed velocity-based qLMPC controller to reject the side-side tower periodic load. The green block extends the control strategy to a fully nonlinear torque-controlled wind turbine, showcasing its effectiveness in a realistic turbine simulation. The presented control strategy assumes that all states and disturbances, such as wind and tower side-side periodic load amplitude, are known and measurable. Consequently, a Kalman filter or any other observer is not included in this setup. Nevertheless, a Kalman state estimator [28] could be implemented as the final step to complete the control framework.

Throughout a period of 250 seconds, four different simulation cases were conducted using a closed-loop system depicted in the yellow block from the block diagram in Figure 6-1. For all cases, the amplitude of the tower's side-side periodic load is introduced as a disturbance to the system after 25 seconds. The controller is then enabled after 50 seconds. This setup allows for the observation of the controller's effectiveness in mitigating the side-side periodic loading, ultimately resulting in no displacement at the top of the tower.

For these four cases, the quasi-LPV continuous-time model is discretized using a sampling time of $t_{\mathrm{s}} = 0.05$ seconds. The four cases are briefly described below:

- **Case 1:** The wind speed is kept constant at 5.5 m/s. The side-side periodic load amplitude on

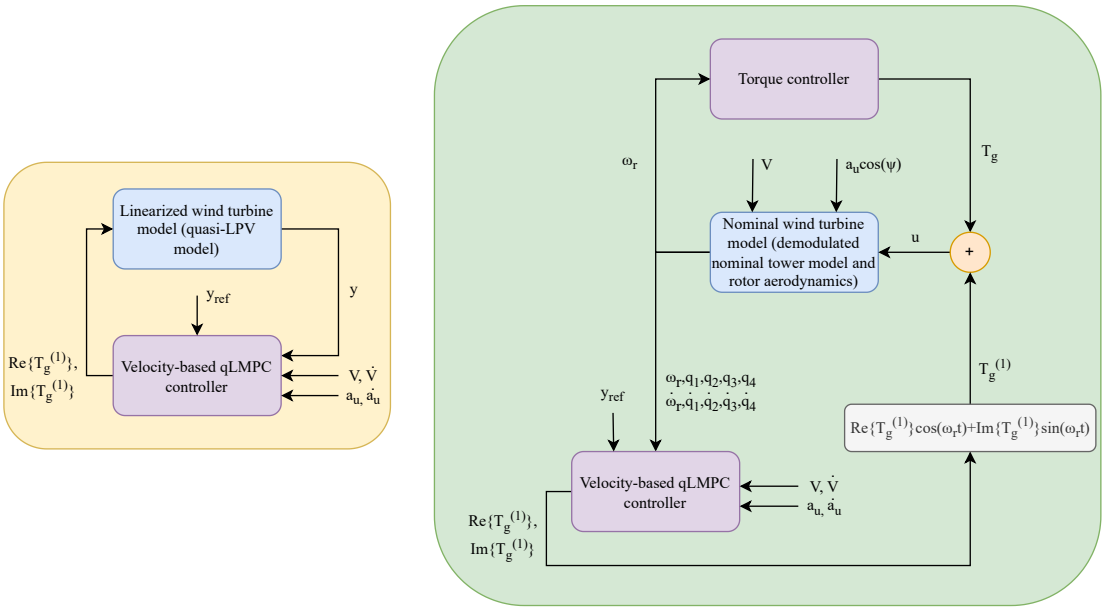

Figure 6-1: The closed-loop systems used for the five simulation cases are shown in the block diagrams. The yellow block represents the quasi-LPV model and velocity-based qLMPC controller designed to mitigate the tower's side-side periodic load. The green block depicts the nonlinear wind turbine model including the demodulated nominal tower model and rotor aerodynamics along with the generator torque controller and velocity-based qLMPC controller for side-side tower periodic load rejection. This control approach assumes that all states and disturbances, including wind and tower side-side periodic load amplitude, are known and measurable. In more realistic scenarios, a state and disturbance estimator would be required to estimate the unmeasurable states, wind speed, and side-side periodic load amplitude. It is important to highlight that this thesis primarily concentrates on the yellow block.

the tower is set to 0 N for the first 25 seconds and then increased to 150 N for the rest of the simulation (Section 6-1).

- **Case 2:** The wind speed is maintained at a constant 5.5 m/s. The side-side periodic load amplitude on the tower starts at 0 N for the first 25 seconds and then increases by 10 N at regular intervals for the remainder of the simulation (Section 6-2).
- **Case 3:** The wind speed increases linearly from 5.5 m/s to 8.5 m/s. Like in Simulation Case 2, the side-side periodic load amplitude on the tower is initially 0 N for the first 25 seconds and then increases by 10 N at regular intervals for the remainer of the simulation (Section 6-3).
- **Case 4:** The quasi-LPV wind turbine model experiences turbulent wind speeds and a side-side periodic load amplitude on the tower based on the centrifugal force generated by the rotor and, thus, dependent on the rotational speed and imbalances in the rotor and blade mass (Section 6-4).

A final simulation case, referred to as Case 5, is designed to validate the effectiveness of the qLMPC controller in stabilizing a nonlinear model of the wind turbine's rotor aerodynamics and tower, subjected to side-side periodic loads. For this case, the closed-loop system depicted in the green block in Figure 6-1 is utilized. Two simulations are conducted, each lasting 50 seconds. A disturbance is introduced after 5 seconds. In one simulation, the qLMPC controller is enabled from the beginning, while in the other, the controller remains disabled throughout. Here, the quasi-LPV continuous-time model is discretized using a sampling time of $t_{\mathrm{s}} = 0.1$ seconds. Below is a brief description of this simulation case:

- **Case 5:** The nominal wind turbine rotor model experiences turbulent wind speeds and a side-side periodic load amplitude on the demodulated nominal tower model, driven by the centrifugal force generated by the rotor. This force is influenced by the rotor's rotational speed and imbalances in the rotor and blade mass. In the simulation where the controller is enabled, the qLMPC utilizes the rotor speed from the nominal rotor model and the states from the demodulated nominal tower model as initial values for each run. The control action determined by the qLMPC is then applied to the nominal models of both the rotor and tower (Section 6-5).

The parameters used for linearization and simulation performed with the (modified) NREL 5-MW reference wind turbine in the below-rated operating region can be found in Table 6-1. Originally, the tower of this reference turbine was classified as soft-stiff. To reclassify it as a soft-soft tower, its wall thickness is reduced by a factor of 7.5 his reduction decreases the tower's mass, bringing its first natural frequency to approximately $\omega_{\mathrm{n}} = 0.7071$ rad/s, similar to that of the soft-soft tower in a simplified wind turbine model. The scaled tower's modal mass, damping, and stiffness are represented as m, d, and k in Table 6-1 and the tower's natural frequency is denoted ω_{n} [13].

As a remark, all the simulations are run using an AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz processor.

6-1 Case 1: Constant wind speed and constant disturbance

This section describes the first simulation setup, presents the results for the first case, and provides a discussion of the results. The results are presented for different scenarios and tuning parameters of the quasi-Linear Model Predictive Control (qLMPC) controller.

Table 6-1: Parameters for linearization and simulation performed with the (modified) NREL 5-MW reference wind turbine in the below-rated operating region [66, 13].

Parameter	Symbol	Value	Units
Gearbox ratio	G	97	-
LSS equivalent inertia	J_{r}	4.0802×10^7	kg m^2
Rotor radius	R	63	m
Tower height	H	90	m
Tower modal mass	m	3.6200×10^5	kg
Tower modal damping	d	2.4588×10^3	kg s^{-1}
Tower modal stiffness	k	1.7677×10^5	kg s^{-2}
Tower natural frequency	ω_{n}	0.7071	rad s^{-1}

Table 6-2: Weighting matrices used for the first simulation case.

Combination	Q_1	Q_2	R	P
1	diag(1,1,1,1,0)$\times 10^3$	diag(1,1,1,1,0,0,0)$\times 10^3$	diag(1,1)$\times 10^{-8}$	$2Q_1$
2	diag(1,1,1,1,0)$\times 10^4$	diag(1,1,1,1,0,0,0)$\times 10^4$	diag(1,1)$\times 10^{-8}$	$2Q_1$
3	diag(0.1,0.1,10,10,0)$\times 10^4$	diag(0.1,0.1,0.1,0.1,0,0,0)$\times 10^4$	diag(1,1)$\times 10^{-8}$	$2Q_1$
4	diag(1,1,10,10,0)$\times 10^4$	diag(1,1,1,1,0,0,0)$\times 10^4$	diag(1,1)$\times 10^{-8}$	$2Q_1$

For this simulation case, the wind speed is maintained at a constant value of 5.5 m/s. The side-side periodic load amplitude on the tower is set to 0 N for the first 25 seconds of the simulation and then adjusted to 150 N for the remaining simulation time - see Figure 6-2.

The states, previously defined in Chapter 5 as $\mathbf{x} = [q_1; q_2; q_3; q_4; \omega_{\mathrm{r}}; \dot{q}_1; \dot{q}_2; \dot{q}_3; \dot{q}_4; \dot{\omega}_{\mathrm{r}}; \dot{V}; \dot{a}_{\mathrm{u}}]$, are initialized with $\mathbf{x}(0) = [0; 0; 0; 0; 0.6113; 0; 0; 0; 0; 0; 0; 0]$. The rotor speed is initialized at 0.6113 rad/s since the wind speed is set to 5.5 m/s at the beginning of the simulation. For the qLMPC controller, the prediction horizon is set to $N = 25$. Additionally, for this simulation case, four different combinations of weighting matrices were defined in Table 6-2 to evaluate the trade-off between the control action and load reductions.

Results and discussion

To gain deeper insights into how the weighting matrices influence the system's behaviour, four different combinations of these matrices as outlined in Table 6-2 were used to conduct simulations. Analyzing the modulated signals for wind turbine tower top displacement and side-side periodic force, depicted in Figure 6-3, over a shorter duration of the simulation provides a window into transient behaviour.

Notably, as the penalty on tower top displacement increases, the system reaches its steady-state goal more quickly. Simultaneously, the peak control input within the side-side periodic force modulated signal decreases (refer to Figure 6-3). This makes sense intuitively: a higher penalty on the states emphasizes the importance of the system following the reference or reaching its goal. Thus, the greater the penalty, the faster the system achieves steady-state. Between Combination 3 and Combination 4, the main difference lies in the increased penalty on velocity and acceleration-related states. By prioritizing the rapid achievement of zero velocity and acceleration, the system can reach a steady

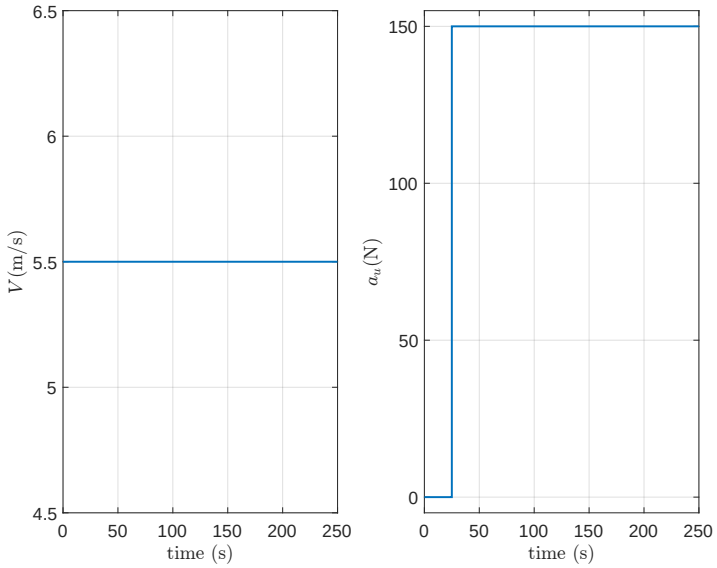

Figure 6-2: Left plot: Wind speed. **Right plot:** Tower side-side periodic load amplitude. The wind speed is maintained at a constant value of 5.5 m/s. The side-side periodic load on the tower is set to 0 N for the first 25 seconds of the simulation and then adjusted to 150 N for the remaining simulation time.

state more quickly and with smaller control actions. Therefore, the remaining results presented in this chapter were obtained using Combination 4.

It is crucial to note that these signals are derived from control inputs, rotational speed, and demodulated tower states obtained through closed-loop simulation, with the quasi-Linear Parameter-Varying (quasi-LPV) model of the wind turbine tower and aerodynamics serving as the plant, and the qLMPC acting as the controller.

The behaviour of the quasi-LPV model under constant wind speed, constant disturbance (or tower side-side periodic loading amplitude), and controlled by the velocity-based qLMPC algorithm, is analyzed in Figure 6-4. The figure shows the outputs $\{q_1, q_2, q_3, q_4, \omega_\mathrm{r}\}$ as well as the real and imaginary parts of the first harmonic of the torque signal (control inputs).

For the first 25 seconds, no disturbance or control input is applied, and therefore, all outputs remain at their initial conditions. After 25 seconds, the disturbance becomes active, and the outputs related to the real and imaginary parts of the tower's top velocity and displacement signals start oscillating with an increase in the oscillation's amplitude over time. The rotational speed remains constant as it is not affected by the tower's dynamics.

After 50 seconds, the controller is enabled, indicated by the outputs related to the real and imaginary parts of the tower top velocity and displacement signals going to zero as intended in the control objective. Moreover, the rotational speed exhibits a notable increase as the controller responds to the disturbance. During the transient phase, the control action shows a significant jump in both the imaginary and real parts of the first harmonic of the torque signal. Once in steady-state, the real part settles at approximately -9.35 kNm, while the imaginary part stabilizes around 0 kNm. Consequently, the rotational speed increases substantially during the transient, stabilizing at around 1.41 rad/s once the system reaches a steady state.

As expected, the imaginary part of the control input diminishes, since the disturbance is a simple

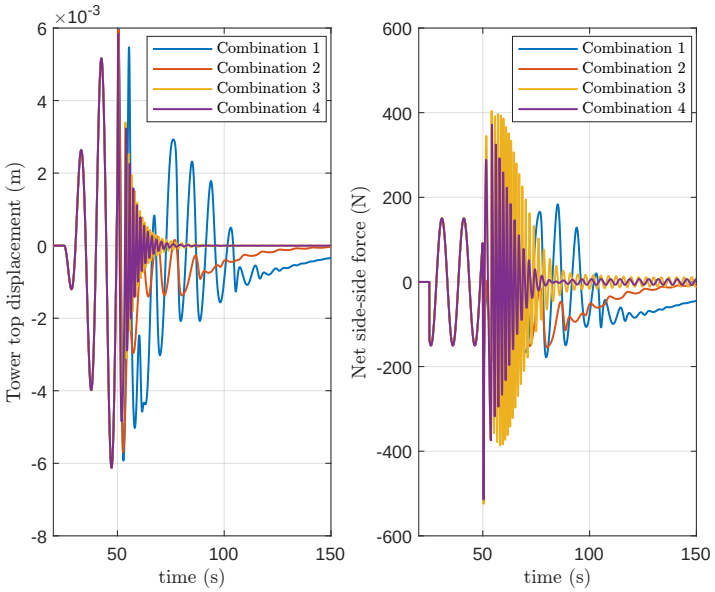

Figure 6-3: Left plot: Modulated tower top displacement signal. **Right plot:** Side-side periodic force. After conducting simulations using four different combinations of weighting matrices, it was observed that a higher penalty on the states emphasizes the importance of the system following the reference or reaching its goal. Thus, the greater the penalty, the faster the system achieves steady-state.

cosine wave with zero phase. This means that only the real part of the control input should be used to cancel this disturbance.

To analyze the impact of the control action derived from the qLMPC controller in mitigating the effects of the tower's side-side periodic load, please refer to Figure 6-5. The top left plot shows the side-side periodic component acting as a disturbance, which begins to influence the system with a constant amplitude of 150 N after 25 seconds. The top right plot displays the component related to the torque. In the bottom plots, we observe the sum of both components in the bottom left plot and a zoomed-in version in the bottom right plot.

It is evident that the torque component significantly mitigates the disturbance caused by the side-side periodic load on the tower, reducing it from an amplitude of 150 N to 7 N in steady-state. While the disturbance is not completely eliminated, it is considerably reduced by approximately 95.3%.

As for the modulated signals for the tower top displacement and velocity, it can be observed in Figure 6-6 that both signals asymptotically converge to zero as intended in the control objective.

The number of iterations needed for the velocity-based qLMPC algorithm to converge is an interesting result to consider - see Figure 6-7. It is noticeable that once a steady state is reached, only one iteration is needed for the algorithm to converge in each run. This shows that the warm-start technique, previously mentioned in Chapter 3, is effective in minimizing the number of required iterations for convergence of the algorithm. Additionally, the time needed for each run of the algorithm remains below 0.06 seconds in steady-state, suggesting that this controller could be used in online applications.

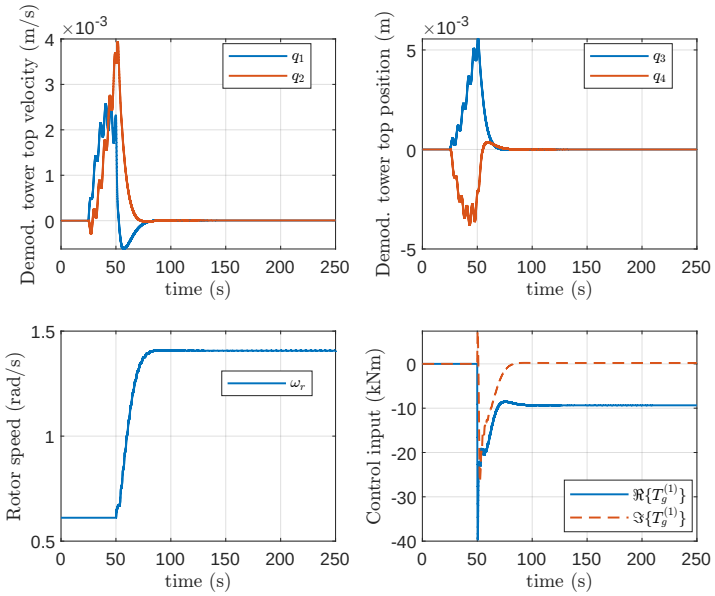

Figure 6-4: Top left plot: Outputs $\{q_1, q_2\}$. **Top right plot:** Outputs $\{q_3, q_4\}$. **Bottom left plot:** Output $\{\omega_r\}$. **Bottom right plot:** Real and imaginary parts of the first harmonic of the torque signal (control inputs). For the first 25 seconds, there is no disturbance or control input, so all outputs stay at their initial conditions. After 25 seconds, the disturbance activates, causing oscillations in the tower's top velocity and displacement signals. The rotational speed remains constant as it is unaffected by the tower's dynamics. At 50 seconds, the controller is enabled, stabilizing the velocity and displacement signals at zero as intended. The rotational speed increases due to the control inputs related to the torque signal's first harmonic. The imaginary part of the control input converges to zero, as the disturbance is a simple cosine wave with zero phase, requiring only the real part of the control input to cancel it.

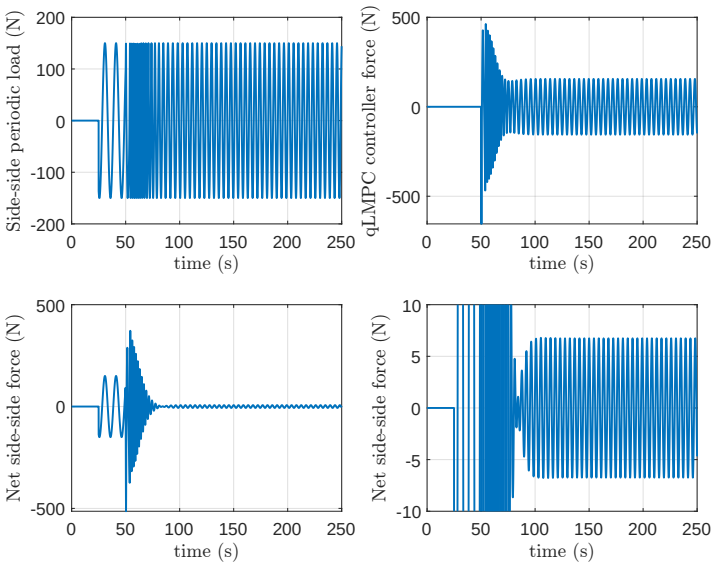

Figure 6-5: Top left plot: side-side periodic component acting as a disturbance, which begins to influence the system with a constant amplitude of 150 N after 25 seconds. **Top right plot:** Component related to the torque and obtained through the use of the velocity-based qLMPC algorithm. **Bottom left plot:** Sum of both components leading to the total side-side periodic force applied to the tower. **Bottom right plot:** Zoomed-in version of the bottom left plot to better analyse the steady-state behaviour of the total side-side periodic force. The torque component significantly reduces the disturbance caused by the side-side periodic load on the tower, decreasing it from an amplitude of 150 N to 7 N in steady-state.

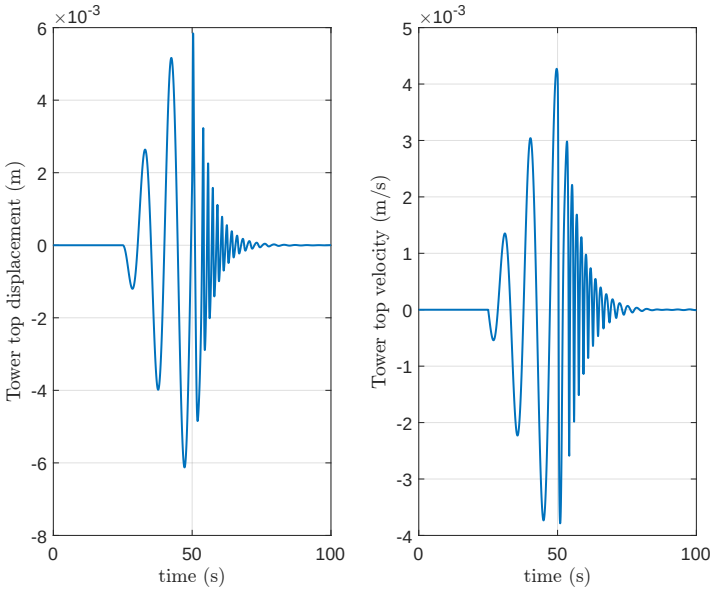

Figure 6-6: Left plot: Modulated signal for the tower top displacement. **Right plot:** Modulated signal for the tower top velocity. The plots are shown just for the initial 100 seconds as the behaviour of the system remains similar for the rest of the simulation time. Both signals asymptotically converge to zero as intended in the control objective.

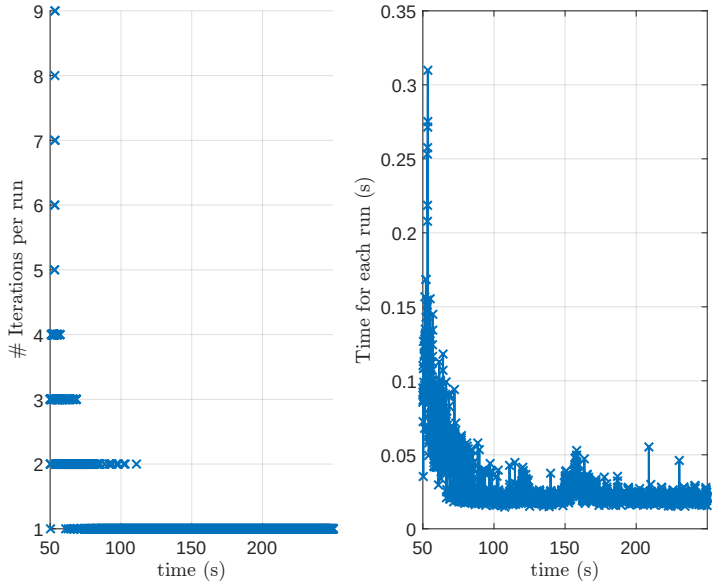

Figure 6-7: Left plot: Number of iterations needed for the velocity-based qLMPC algorithm to converge per run. **Right plot:** Time needed for each run of the algorithm. It is clear that once a steady state is reached, only one iteration is needed for the algorithm to converge in each run. The time needed for each run remains below 0.06 seconds in steady-state, suggesting that this controller could be used in online applications.

6-2 Case 2: Constant wind speed and stair disturbance

This section presents the setup, results, and discussion for the second case. For this simulation case, the wind speed is maintained at a constant value of 5.5 m/s. The side-side periodic load amplitude on the tower starts at 0 N for the first 25 seconds and then increases by 10 N at regular intervals for the remainder of the simulation - see Figure 6-8.

The states are initialized with $\mathbf{x}(0) = [0; 0; 0; 0; 0.6113; 0; 0; 0; 0; 0; 0; 0]$. For the quasi-Linear Model Predictive Control (qLMPC) controller, the prediction horizon is set to $N = 25$ and the weighting matrices are the same as the final ones defined in Section 6-1.

Results and discussion

The behaviour of the quasi-Linear Parameter-Varying (quasi-LPV) model under constant wind speed and stair disturbance, controlled by the velocity-based qLMPC algorithm, is analyzed in Figure 6-9. After 25 seconds, the disturbance becomes active, causing instability in the tower's top velocity and displacement signals. Enabling the controller at 50 seconds reduces oscillations and stabilizes the system. The imaginary part of the control input goes to zero, as the disturbance is a simple cosine wave with zero phase, so only the real part is used to cancel the disturbance. The real part of the control input follows the opposite trend of the disturbance to mitigate or cancel it.

To investigate the impact of the qLMPC controller in mitigating the tower's side-side periodic load, refer to Figure 6-10. The top left plot shows the disturbance with gradually increasing amplitude starting after 25 seconds. The top right plot displays the torque component. The bottom left plot shows the sum of both components, with a zoomed-in version in the bottom right plot.

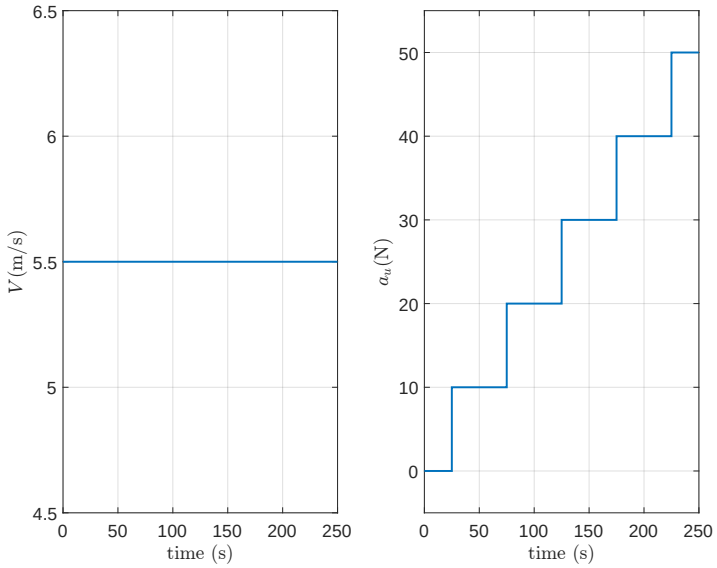

Figure 6-8: Left plot: Wind speed. **Right plot:** Tower side-side periodic load amplitude. The wind speed is maintained at a constant value of 5.5 m/s. The side-side periodic load amplitude on the tower starts at 0 N for the first 25 seconds and then increases by 10 N at regular intervals for the remainder of the simulation.

The torque component significantly reduces the disturbance caused by the side-side periodic load. When there is a change in the disturbance, the system is initially affected and the total side-side force amplitude increases slightly to compensate for the change in the disturbance. However, the controller quickly brings the total side-side force asymptotically back to a value very close to zero.

As for the modulated signals for the tower top displacement and velocity, it can be observed in Figure 6-11 that both signals asymptotically converge to zero as intended in the control objective.

The number of iterations needed for the velocity-based qLMPC algorithm to converge is shown in Figure 6-12. Once a steady state is reached, the algorithm converges in just one iteration. However, when there is an increase in the disturbance, the controller must compensate for the abrupt changes, requiring more iterations to converge. Despite this, the controller allows the system to rapidly re-establish a steady state where only one iteration is needed again for the velocity-based qLMPC algorithm to converge. Additionally, each run of the algorithm consistently takes less than 0.18 seconds throughout the simulation, indicating that this controller is suitable for online applications.

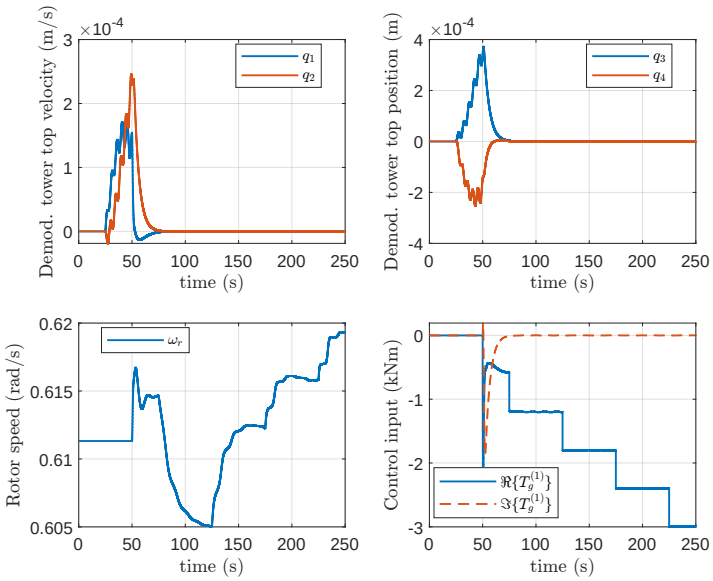

Figure 6-9: Top left plot: Outputs $\{q_1, q_2\}$. **Top right plot:** Outputs $\{q_3, q_4\}$. **Bottom left plot:** Output $\{\omega_r\}$. **Bottom right plot:** Real and imaginary parts of the first harmonic of the torque signal (control inputs). After 25 seconds, the disturbance becomes active, leading to unstable behaviour in the tower's top velocity and displacement signals. Adding the controller at 50 seconds reduces the oscillations and stabilizes the system. As expected, the imaginary part of the control input goes to zero, since the disturbance is a simple cosine wave with zero phase. On the other hand, the real part of the control input follows the opposite trend of the disturbance, as desired, to mitigate or cancel it.

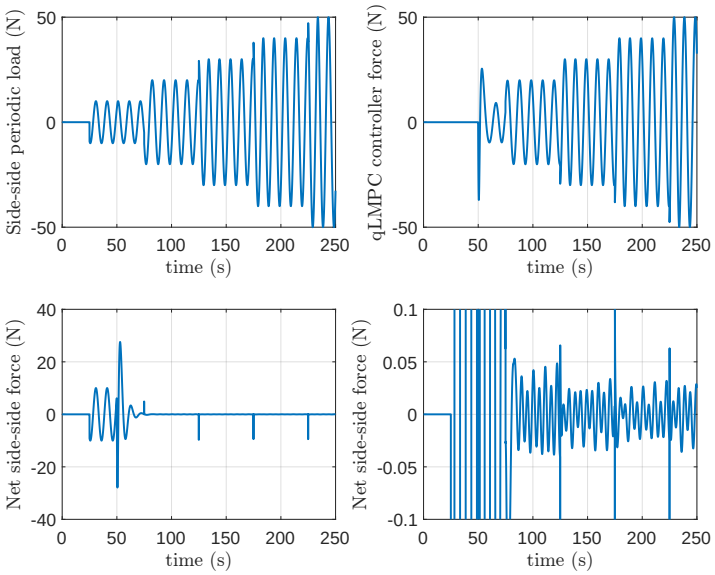

Figure 6-10: Top left plot: side-side periodic component acting as a disturbance with gradually increasing amplitude starting after 25 seconds. **Top right plot:** Component related to the torque and obtained through the use of the velocity-based qLMPC algorithm. **Bottom left plot:** Sum of both components leading to the total side-side periodic force applied to the tower. **Bottom right plot:** Zoomed-in version of the bottom left plot to better analyse the steady-state behaviour of the total side-side periodic force. The torque component significantly reduces the disturbance caused by the side-side periodic load. Although the system is initially affected by changes in the disturbance, the controller quickly brings the total side-side force back to nearly zero.

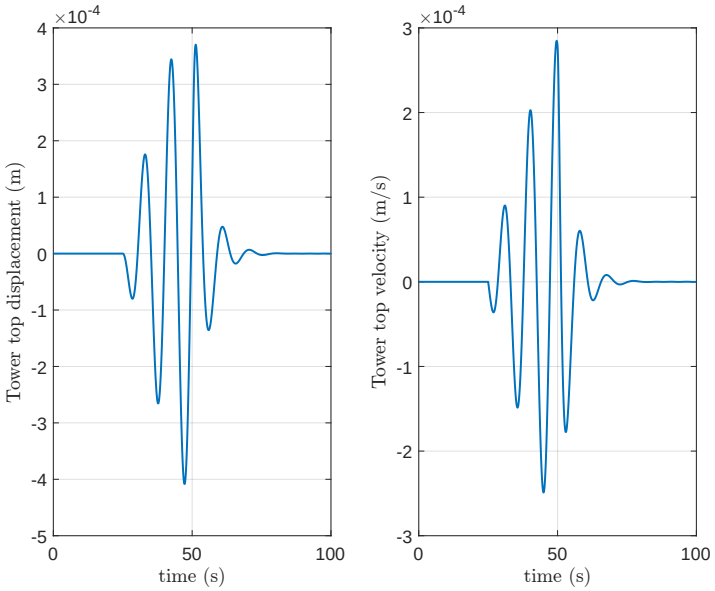

Figure 6-11: Left plot: Modulated signal for the tower top displacement. **Right plot:** Modulated signal for the tower top velocity. The plots are shown just for the initial 100 seconds as the behaviour of the system remains similar for the rest of the simulation time. Both signals asymptotically converge to zero as intended in the control objective.

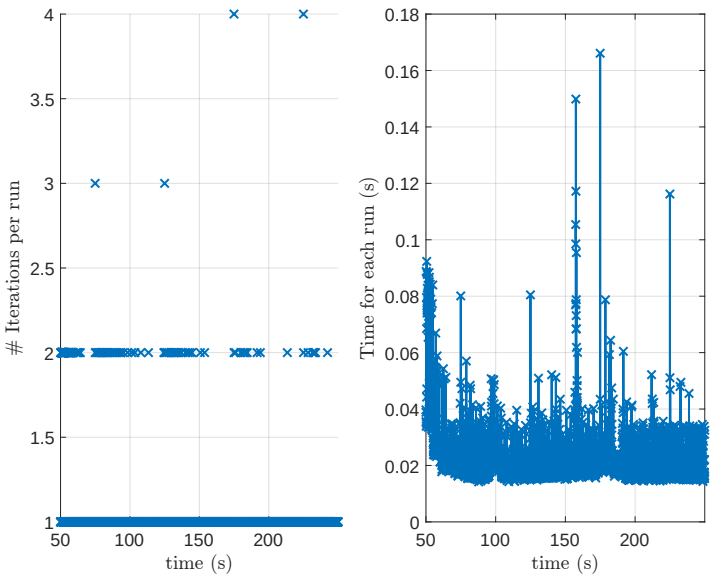

Figure 6-12: Left plot: Number of iterations needed for the velocity-based qLMPC algorithm to converge per run. **Right plot:** Time needed for each run of the algorithm. Once a steady state is achieved, the algorithm requires only one iteration for convergence. However, when there is an increase in the disturbance, the controller must compensate for the abrupt changes, requiring more iterations to converge. Despite this, the controller enables the system to quickly re-establish a steady state, where only one iteration is needed again for the velocity-based qLMPC algorithm to converge. Moreover, the time required for each run of the algorithm remains below 0.18 seconds throughout the simulation, indicating that this controller is suitable for online applications.

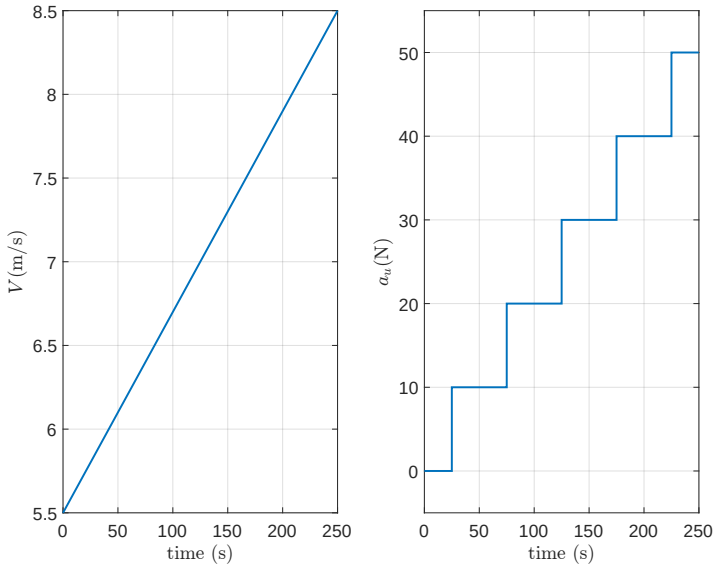

Figure 6-13: Left plot: Wind speed. **Right plot:** Tower side-side periodic load amplitude. The wind speed ramps linearly from 5.5 m/s to 8.5 m/s. The side-side periodic load amplitude on the tower starts at 0 N for the first 25 seconds and then increases by 10 N at regular intervals for the remainder of the simulation.

6-3 Case 3: Linearly increasing wind speed and stair disturbance

The simulation setup, results and discussion for Case 3 are provided in this section. In the third simulation setup, the wind speed ramps linearly from 5.5 m/s to 8.5 m/s. The side-side periodic force has the same behaviour described in Section 6-2 and is shown in Figure 6-13. The states are initialized with $\mathbf{x}(0) = [0; 0; 0; 0; 0.6113; 0; 0; 0; 0; 0; 0; 0]$. The parameters for the quasi-Linear Model Predictive Control (qLMPC) controller are the same as in the previous simulation cases.

Results and discussion

As in the previous cases, let's first analyze the behaviour of the quasi-Linear Paramter-Varying (quasi-LPV) model under linearly changing wind speed and stair disturbance, controlled by the velocity-based qLMPC algorithm (see Figure 6-14). This simulation shows behaviour similar to the previous case in Section 6-2, especially in the demodulated signals for tower velocity and position. The control inputs also exhibit a similar trend. However, this time the rotational speed increases from the start, which is expected since the wind speed is linearly increasing and rotational speed depends on it.

To examine the control action's impact on mitigating the tower's side-side periodic load, refer to Figure 6-15. The torque component significantly reduces the disturbance caused by the side-side load. Similar to the behaviour seen in the previous simulation, when the disturbance changes, the system is initially affected, increasing the total side-side force amplitude slightly, but the controller quickly brings the net side-side force back to nearly zero asymptotically leading the system to a steady state.

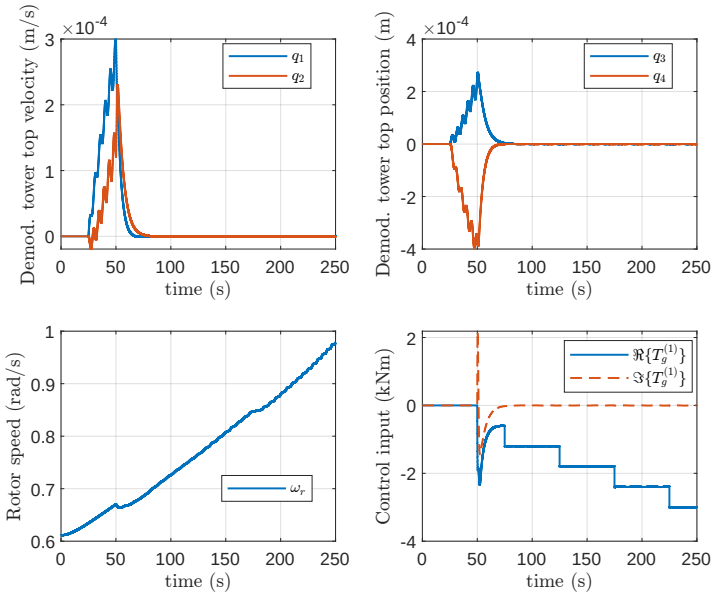

Figure 6-14: Top left plot: Outputs $\{q_1, q_2\}$. **Top right plot:** Outputs $\{q_3, q_4\}$. **Bottom left plot:** Output $\{\omega_r\}$. **Bottom right plot:** Real and imaginary parts of the first harmonic of the torque signal (control inputs). After 25 seconds, the disturbance becomes active, leading to unstable behaviour in the tower's top velocity and displacement signals. Adding the controller at 50 seconds reduces the oscillations and stabilizes the system. As expected, the imaginary part of the control input goes to zero, since the disturbance is a simple cosine wave with zero phase. On the other hand, the real part of the control input follows the opposite trend of the disturbance, as desired, to mitigate or cancel it.

For the modulated signals of tower top displacement and velocity (see Figure 6-16), both signals converge to zero as intended.

Regarding the number of iterations for the qLMPC algorithm to converge (see Figure 6-17), one or two iterations are needed once a steady state is reached. Still, more iterations are required whenever there is an abrupt increase in the disturbance. The algorithm's run time remains below 0.1 seconds for this simulation case, except for one outlier which took around 0.45 seconds.

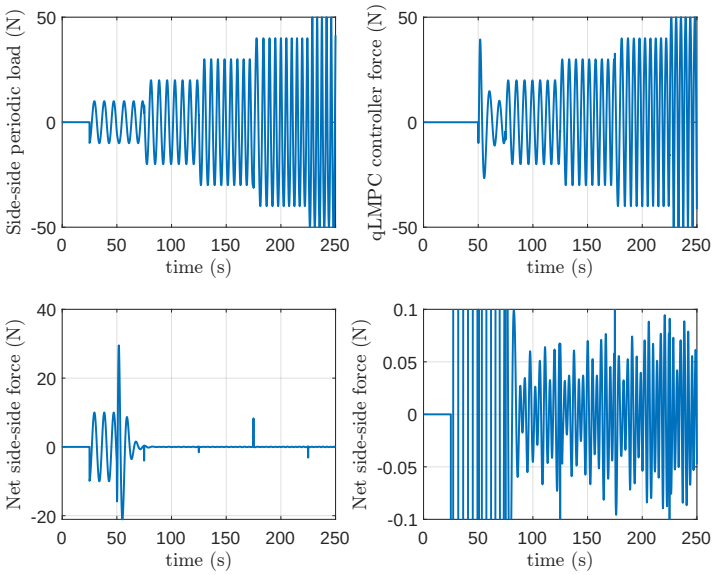

Figure 6-15: Top left plot: side-side periodic component acting as a disturbance with gradually increasing amplitude starting after 25 seconds. **Top right plot:** Component related to the torque and obtained through the use of the velocity-based qLMPC algorithm. **Bottom left plot:** Sum of both components leading to the total side-side periodic force applied to the tower. **Bottom right plot:** Zoomed-in version of the bottom left plot to better analyse the steady-state behaviour of the total side-side periodic force. The torque component significantly reduces the disturbance caused by the side-side load. When the disturbance changes, the system is initially affected, increasing the total side-side force slightly, but the controller quickly brings it back to zero asymptotically.

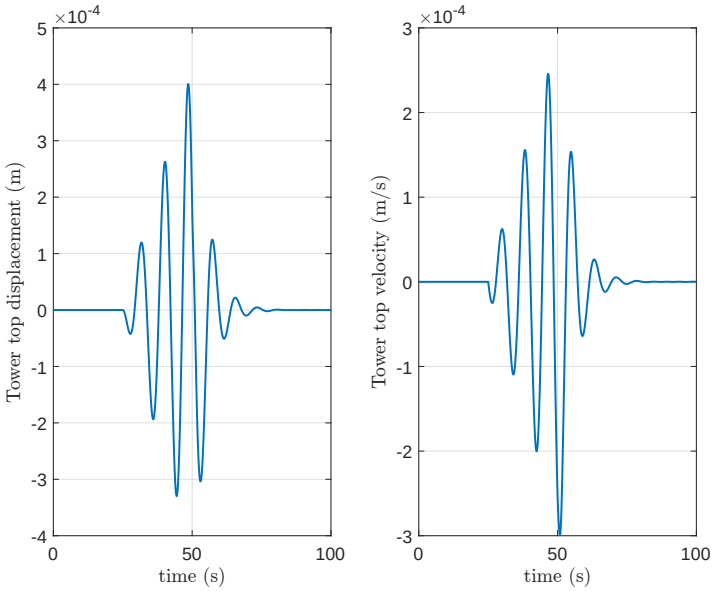

Figure 6-16: Left plot: Modulated signal for the tower top displacement. **Right plot:** Modulated signal for the tower top velocity. The plots are shown just for the initial 100 seconds as the behaviour of the system remains similar for the rest of the simulation time. It can be noticed that both the modulated signals for the tower top velocity and displacement converge to zero.

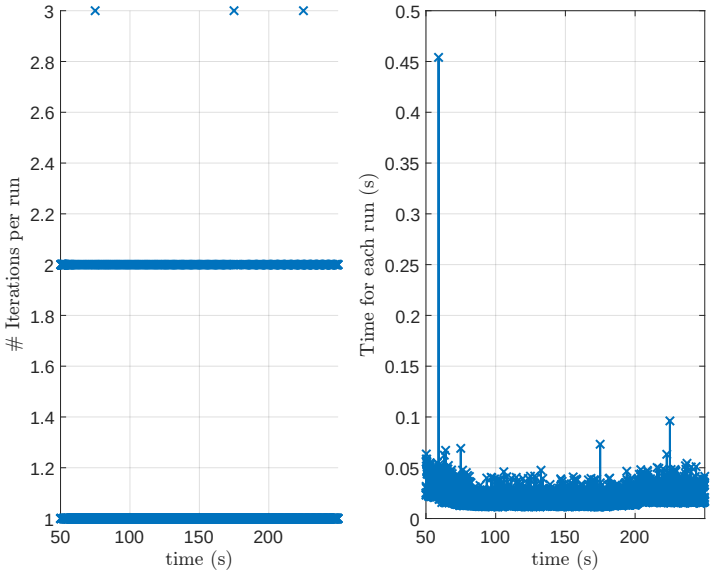

Figure 6-17: Left plot: Number of iterations needed for the velocity-based qLMPC algorithm to converge per run. **Right plot:** Time needed for each run of the algorithm. One or two iterations are needed once a steady state is reached. However, more iterations are required whenever there is an abrupt increase in the disturbance. The algorithm's run time remains below 0.1 seconds for this simulation case, except for one outlier which took around 0.45 seconds.

6-4 Case 4: Turbulent wind speed and disturbance based on the centrifugal force generated by the rotor

This section presents the simulation setup, results, and discussion for Case 4. For this case, the quasi-Linear Parameter-Varying (quasi-LPV) system of the wind turbine and tower is exposed to turbulent wind speeds generated using TurbSim, a turbulence simulator developed by the National Renewable Energy Laboratory (NREL) [67]. The system experiences a disturbance after 25 seconds based on the centrifugal force generated by the rotor and, thus, dependent on the rotational speed and imbalances in the rotor and blade mass, defined as:

$$a_{\mathrm{u}} = m\omega_{\mathrm{r}}^2 \frac{R}{2.5\times10^5}. \tag{6-1}$$

Note that a factor of 2.5×10^5 is chosen to reduce the amplitude of the side-side periodic force to values between 130 N and 225 N. This is illustrated in Figure 6-18.

The states are initialized with $\mathbf{x}(0) = [0; 0; 0; 0; 1.0315; 0; 0; 0; 0; 0; 0; 0]$. The parameters for the qLMPC controller are the same as in the previous simulation cases.

Results and discussion

In Figure 6-19 it is observed that the demodulated signals for the velocity and position of the wind turbine tower become quite unstable once the disturbance affects the system. However, when the controller is enabled, it counteracts the disturbance, stabilizing the system and causing the demodulated signals of velocity and position to converge to zero. The control inputs once again demonstrate

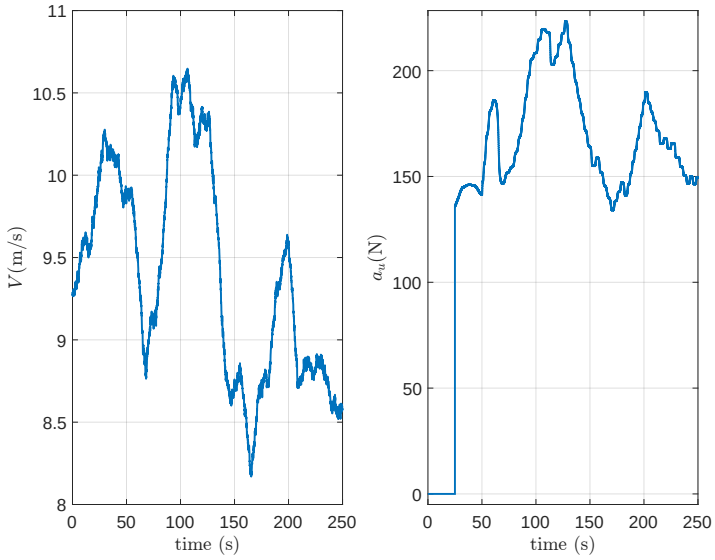

Figure 6-18: Left plot: Wind speed. **Right plot:** Tower side-side periodic load amplitude. The quasi-LPV system of the wind turbine and tower is exposed to turbulent wind speeds generated using TurbSim, a turbulence simulator developed by NREL. For further details on the workings of this simulator, refer to [67]. Additionally, the system experiences a disturbance after 25 seconds, with an amplitude that depends on the rotational speed and imbalances in the rotor and blade mass.

that the imaginary part is used only during the transient phase. Once steady-state is reached, the imaginary part is no longer needed as the disturbance has zero phase. The real part of the control input is then used to counteract the disturbance. The rotational speed exhibits a behaviour similar to that of the side-side force amplitude, as expected since it is dependent on the control input and this has a mirrored behaviour to that of the disturbance.

To examine the control action's impact on mitigating the tower's side-side periodic load, refer to Figure 6-20. The torque component significantly reduces the disturbance caused by the side-side load. However, this time the controller is less effective in reducing the influence of the disturbance, resulting in values between -2 N and 1.5 N for the side-side force. Nevertheless, these amplitudes are quite small considering the wind turbine's size and mass and represent about around 0.2% of the original disturbance amplitude.

For the modulated signals of tower top displacement and velocity (see Figure 6-21), both signals converge to zero as intended.

As for the number of iterations for the quasi-Linear Model Predictive Control (qLMPC) algorithm to converge presented in Figure 6-17), this can vary as the system is in the presence of turbulent wind speed and also time-varying disturbance. The algorithm's run time remains below 0.2 seconds for this simulation case, except for one outlier which took around 0.43 seconds.

As a final remark for Simulation Case 4, the simulation was run 100 times to calculate the average run time of the algorithm for the system under more real-life conditions. The average run time was around 254.51 seconds for the entire 250-second simulation - see Table 6-3. This result demonstrates that the algorithm is well-suited for real-time applications, providing an efficient qLMPC algorithm for wind turbine control subject to tower periodic loads.

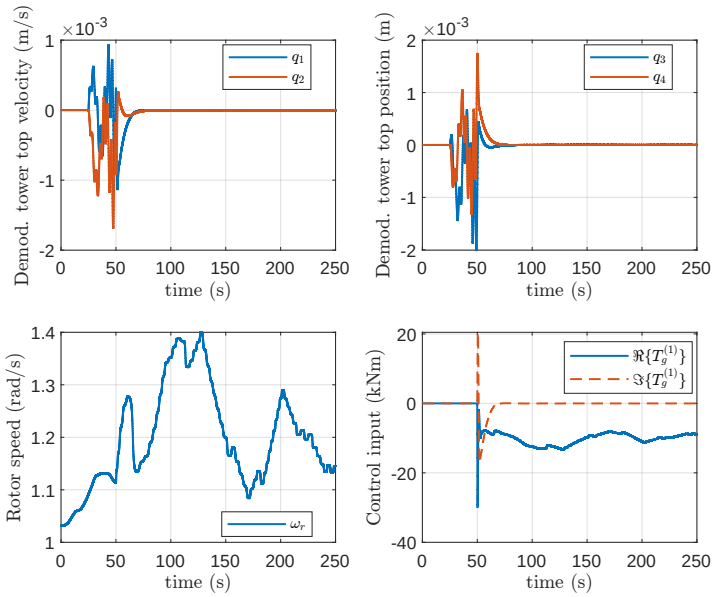

Figure 6-19: Top left plot: Outputs $\{q_1, q_2\}$. **Top right plot:** Outputs $\{q_3, q_4\}$. **Bottom left plot:** Output $\{\omega_r\}$. **Bottom right plot:** Real and imaginary parts of the first harmonic of the torque signal (control inputs). After 25 seconds, the disturbance becomes active, leading to unstable behaviour in the tower's top velocity and displacement signals. Adding the controller at 50 seconds reduces the oscillations and stabilizes the system. As expected, the imaginary part of the control input goes to zero, since the disturbance is a simple cosine wave with zero phase. On the other hand, the real part of the control input follows the opposite trend of the disturbance, as desired, to mitigate or cancel it. The rotational speed exhibits a behaviour similar to that of the side-side force amplitude, as expected since it is dependent on the control input and this has a mirrored behaviour to that of the disturbance.

Table 6-3: Average run time for a total number of 100 runs when using the velocity-based qLMPC algorithm to control the quasi-LPV model of the wind turbine and tower dynamics under the conditions defined for Simulation Case 4.

Algorithm	# of runs	Average run time (s)
Velocity-based qLMPC	100	254.5182

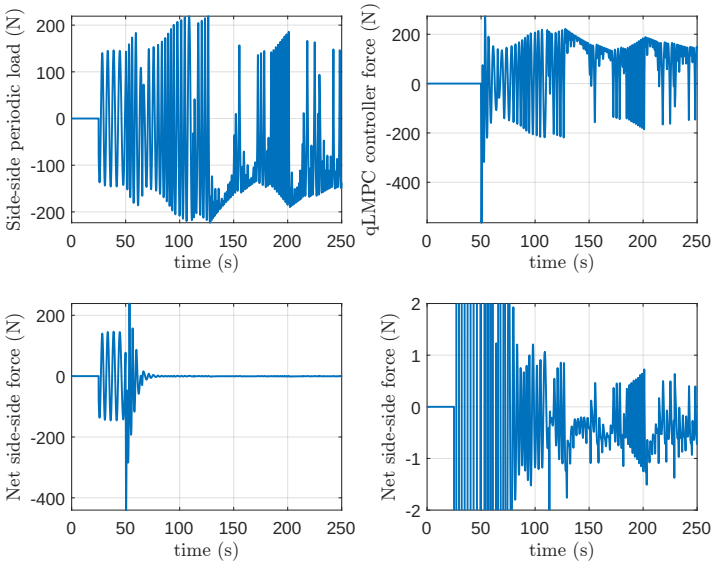

Figure 6-20: Top left plot: side-side periodic component acting as a disturbance after 25 seconds based on the centrifugal force generated by the rotor and, thus, dependent on the rotational speed and imbalances in the rotor and blade mass. **Top right plot:** Component related to the torque and obtained through the use of the velocity-based qLMPC algorithm. **Bottom left plot:** Sum of both components leading to the total side-side periodic force applied to the tower. **Bottom right plot:** Zoomed-in version of the bottom left plot to better analyse the steady-state behaviour of the total side-side periodic force. The torque component significantly reduces the disturbance caused by the side-side load. However, this time the controller is less effective in reducing the influence of the disturbance, resulting in values between -2 N and 1.5 N for the side-side force.

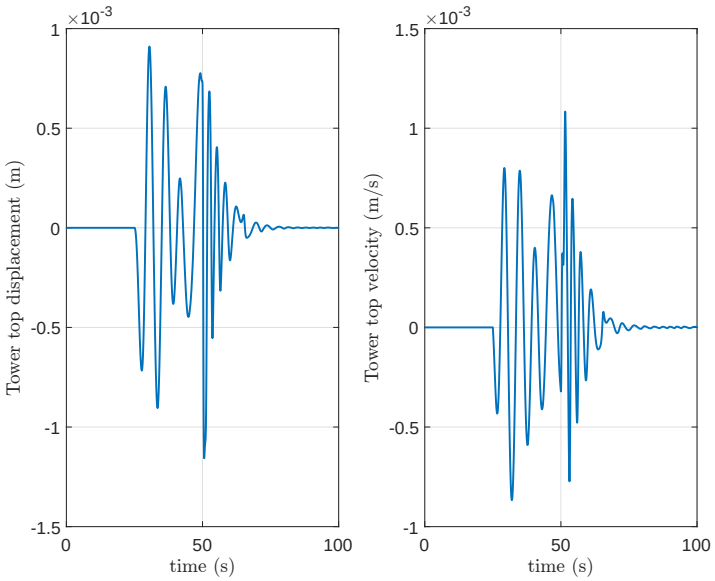

Figure 6-21: Left plot: Modulated signal for the tower top displacement. **Right plot:** Modulated signal for the tower top velocity. The plots are shown just for the initial 100 seconds as the behaviour of the system remains similar for the rest of the simulation time. Both signals asymptotically converge to zero as intended in the control objective.

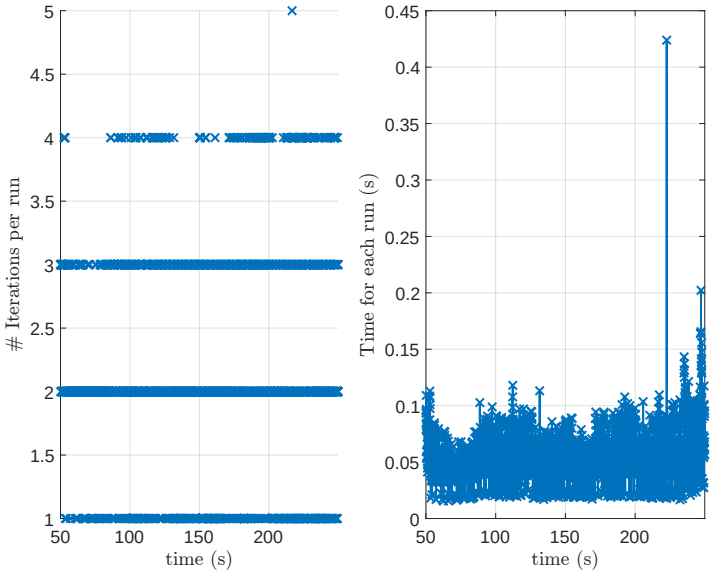

Figure 6-22: Left plot: Number of iterations needed for the velocity-based qLMPC algorithm to converge per run. **Right plot:** Time needed for each run of the algorithm. The number of iterations needed for the velocity-based qLMPC algorithm to converge per run can vary as the system is in the presence of turbulent wind speed and also time-varying disturbance. The algorithm's run time remains below 0.2 seconds for this simulation case, except for one outlier which took around 0.43 seconds.

6-5 Case 5: Side-side tower periodic load rejection in the nominal wind turbine model using a velocity-based qLMPC

This section presents the simulation setup, results, and discussion for Case 5. The fifth simulation setup is designed to validate the effectiveness of the quasi-Linear Model Predictive Control (qLMPC) controller in stabilizing a nonlinear model of the wind turbine's rotor aerodynamics and tower, which are subjected to side-side periodic loads. Two 50-second simulations are conducted: in one simulation, the qLMPC controller is enabled from the beginning, while in the other, the controller remains disabled throughout. These simulations are run using Simulink. For more information about the Simulink block diagram, the reader is referred to Appendix A. Note that a shorter simulation time was chosen in this case for efficiency, as each full 50-second simulation took approximately 30 minutes to run. This way, it was possible to promptly re-tune the controller and evaluate the final results effectively.

The nominal wind turbine rotor model experiences turbulent wind speeds and an artificially chosen side-side periodic load amplitude on the demodulated nominal tower model, driven by the centrifugal force generated by the rotor. The system encounters this force after 5 seconds, with an amplitude that depends on the rotational speed and imbalances in the rotor and blade mass, defined as in (6-1).

In the simulation where the controller is enabled, the qLMPC uses the rotor speed from the nominal rotor model and the states from the demodulated nominal tower model as initial values for each run. The control action determined by the qLMPC is then applied to the nominal models of both the rotor and tower.

The states are initialized with $\mathbf{x}(0) = [0; 0; 0; 0; 0.9040; 0; 0; 0; 0; 0; 0; 0]$. The parameters for the

qLMPC controller are consistent with those used in the previous simulation cases, with the exception of a change in the sampling time from $t_s = 0.05$ seconds to $t_s = 0.1$ seconds. The penalty on the states related to the output of the system is also adapted to $Q_1 = \text{diag}(1, 1, 1, 1, 0) \times 10^4$ (corresponding to Combination 2 from Table 6-2).

The modifications to the controller were necessary due to its inability to converge to the optimal scheduling sequence with the previous Q_1 matrix and sampling time t_s when disturbances were introduced. It was also observed that the optimization algorithm sometimes reached the maximum number of iterations without finding an optimal solution when using the same parameters as in the other four simulation cases. These adjustments were crucial since the control target shifted from a linearized model to the nominal wind turbine model. This shift led to a potential mismatch between the states in the qLMPC controller, which was based on the linearized model, and the actual states of the nonlinear system due to differing dynamics. Consequently, re-tuning the controller by relaxing the penalty on the states and sampling at a lower frequency was essential to ensure the algorithm's convergence when applied to the nominal wind turbine model. It is also important to realize that by increasing the sampling time the prediction horizon increases from 1.25 to 2.5 seconds. Thus, with the new sampling time, the controller "looks" further into the predicted future behaviour of the system. Usually, higher prediction horizons are favourable as more information is given to the controller to better evaluate the optimal action to take in the next time step.

Results and discussion

In Figure 6-23, the wind and rotor speed, side-side tower periodic load amplitude, and tower top displacement for simulations with and without the velocity-based qLMPC controller can be analysed. In the simulation where the velocity-based qLMPC controller is disabled, the rotor speed is smaller, and the amplitude of the side-side periodic force applied to the top of the tower is also smaller compared to the simulation with the controller enabled.

When examining the tower top displacement with the controller disabled, the disturbance increases the displacement amplitude. This displacement decreases due to the natural damping from the tower model but starts to increase again after a few seconds, each time with a lower amplitude. It is interesting to realize that this behaviour is related to the fact that the tower is modelled as a second-order mass-spring-damper system.

In the simulation where the controller is enabled, once the disturbance is applied, the tower top displacement amplitude drifts slightly from zero but quickly decreases and converges back asymptotically to zero (see zoomed-in plot of the tower top displacement in Figure 6-23). This demonstrates the efficiency and capability of the designed control algorithm in mitigating the influence of the side-side periodic load on the tower top displacement when applied to the nominal wind turbine model.

To illustrate the effectiveness of the velocity-based qLMPC controller in reducing the influence of the side-side periodic load on the tower of the wind turbine, Figure 6-24 is presented. Here it can be noticed that the force obtained from the designed controller practically mirrors the behaviour of the side-side periodic load, leading to a net force applied to the tower consisting of just 0.98% of the original side-side periodic load, on average.

As a final remark, the impact of the torque derived from the velocity-based qLMPC controller on the overall torque produced (combining the torque controller and the velocity-based qLMPC) is analyzed in Figure 6-25. The generator torque exhibits significant changes when the torque from the velocity-based qLMPC is added to that from the torque controller. Notably, a periodic signal is superimposed

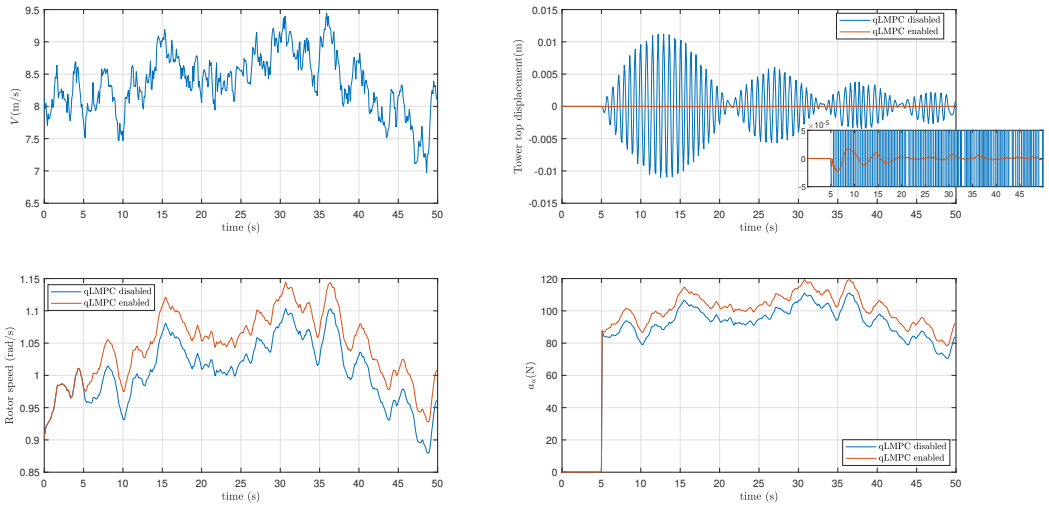

Figure 6-23: Wind and rotor speed, side-side tower periodic load amplitude, and tower top displacement for simulations with (red solid lines —) and without (blue solid lines —) the velocity-based qLMPC controller. In the simulation without the velocity-based qLMPC controller, the rotor speed and the amplitude of the side-side periodic force at the tower top are smaller than with the controller enabled. When the controller is disabled, the disturbance increases the tower top displacement amplitude. Though natural damping reduces the displacement, it increases again after a few seconds, with each peak smaller than the last. This behaviour is related to the fact that the tower is modelled as a second-order mass-spring-damper system. With the controller enabled, the disturbance causes a slight initial drift in the tower top displacement amplitude, but it quickly decreases and converges back to zero, demonstrating the controller's effectiveness in mitigating the side-side periodic load's impact on the tower top displacement (see zoomed-in plot of the tower top displacement).

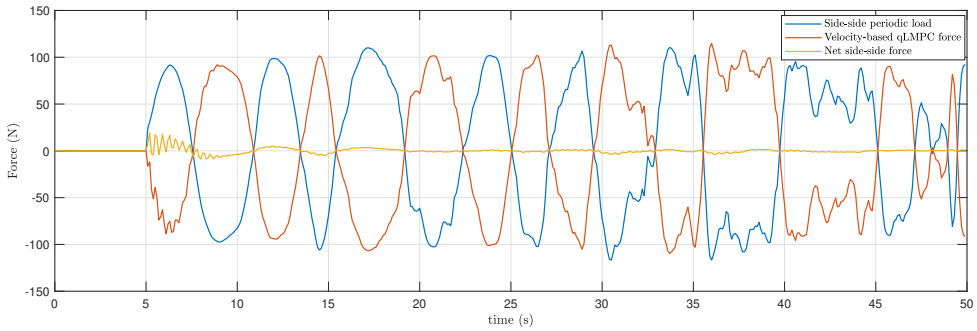

Figure 6-24: Side-side periodic load (blue solid line —), velocity-based qLMPC force (red solid line —), and net side-side force (yellow solid line —). The force generated by the designed controller closely mirrors the behaviour of the side-side periodic load. This results in a net force applied to the tower that is only 0.98% of the original side-side periodic load, on average.

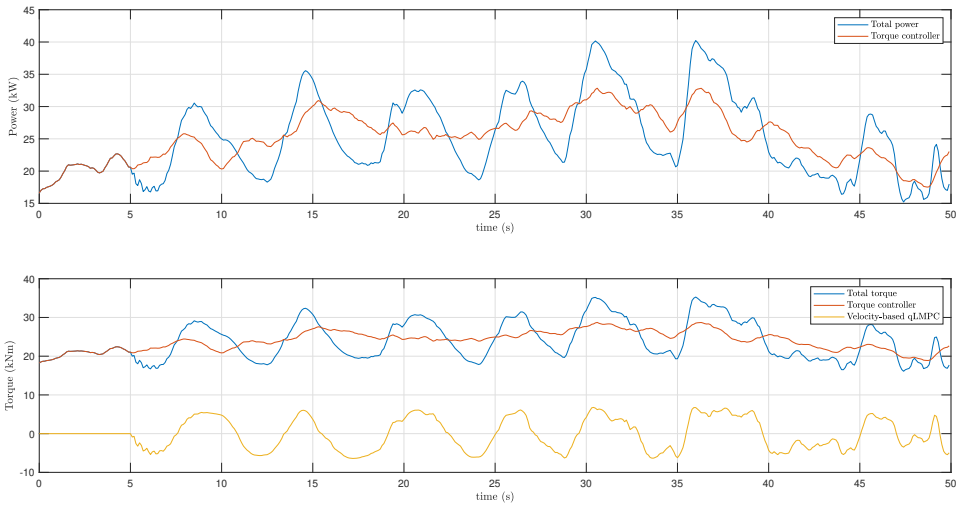

Figure 6-25: Top plot: Total generator power (blue solid line —), and power related to the torque obtained from the torque controller (red solid line —). **Bottom plot:** Total generator torque (blue solid line —), torque obtained from the torque controller (red solid line —), and torque obtained from the velocity-based qLMPC controller (yellow solid line —). The generator torque undergoes notable changes when the torque from the velocity-based qLMPC is combined with that from the torque controller, resulting in a periodic signal overlaying the original torque and, consequently, the power signal. However, drawing definitive conclusions about the control algorithm's impact on power production is premature, as the artificially set side-side tower periodic load amplitude may not be realistic.

on the original torque signal. The relationship between power and torque is defined by the following equation:

$$P = T\omega_{\mathrm{r}} \tag{6-2}$$

where P represents the power, and T is the net torque. For this specific simulation, the inclusion of the velocity-based qLMPC controller markedly altered the trend in generated power, as illustrated in Figure 6-25. Consequently, a periodic signal is also superimposed on the original power signal, reflecting its dependence on torque.

However, it is essential to recognize that drawing precise and definitive conclusions about the velocity-based qLMPC algorithm's effect on power production is premature. This caution arises as the side-side tower periodic load amplitude was artificially set and may not yet be sufficiently realistic.

Chapter 7

Conclusions and recommendations

This chapter draws the main conclusions of this thesis and provides recommendations for future research. It begins by summarizing the key findings of the study, highlighting how they address the research questions and objectives outlined in the introductory chapter. The conclusions reflect on the significance of these findings, emphasizing their potential contributions to advancing the understanding of wind turbine control and optimization.

7-1 Conclusions

This thesis introduces an innovative approach to wind turbine control, focusing on addressing tower side-side periodic loads using a quasi-Linear Model Predictive Control (qLMPC) framework. Tall slim towers possess inherent flexibility that can lead to resonant behavior, particularly in variable-speed wind turbines, posing risks such as material degradation and structural failure. While conventional control strategies have primarily targeted fore-aft tower movement, the need for advanced methods to actively cancel side-side periodic loads persists.

A key aspect of this approach is the utilization of two specific coordinate transformations. The Model Demodulation Transformation (MDT) technique extracts wind turbine tower dynamics, with the aim of minimizing tower top displacement caused by these periodic forces. Additionally, the model incorporates wind turbine aerodynamics, linearized using a velocity-based approach, and controlled with a qLMPC controller. This method balances the impact on power output (actuation effort) and disturbances from periodic loads. By mapping equilibrium points to the origin in velocity-space, velocity-based linearization simplifies computations and eliminates the need for offline computations of terminal sets and Linear Matrix Inequalities (LMIs). Furthermore, this approach omits the need for extensive equilibrium input and state vectors, reducing memory usage while effectively capturing the system's nonlinear behaviour. Hence, these advancements underscore the efficacy of strategic coordinate transformations in improving wind turbine control strategies.

Simulation parameters for the modified National Renewable Energy Laboratory (NREL) 5-MW reference wind turbine were presented and simulations across five distinct cases were summarized. The first four cases increased in complexity, demonstrating the mechanisms and robustness of the

developed control algorithm. To gain deeper insights into the influence of weighting matrices on the system's behaviour, simulations were conducted using four different combinations of these matrices. Notably, as the penalty on tower top displacement increased, the system reached its steady-state goal more quickly. Simultaneously, the peak control input within the side-side periodic force-modulated signal decreased. This is intuitive: a higher penalty on the states emphasizes the importance of the system following the reference or reaching its goal, leading to a faster achievement of a steady state. Additionally, prioritizing the rapid achievement of zero velocity and acceleration allowed the system to reach a steady state more quickly with smaller control actions.

The first four cases are defined to show the effectiveness, efficiency and working principles of the designed algorithm. In all four simulation cases, the average time required for a single control step remained below 0.2 seconds. The time needed for each run increased slightly with the complexity of the wind speed and disturbance signals encountered. However, even in the most complex scenario (Case 4), the computational time per run was low, demonstrating the algorithm's suitability for real-time applications. This efficiency highlights the velocity-based qLMPC algorithm's effectiveness in the control of wind turbines for side-side tower periodic load reductions.

In the final simulation case, which incorporated a nonlinear wind turbine aerodynamic and tower model, it was concluded that the designed controller efficiently rejects tower side-side periodic loads when controlling a nonlinear model of the wind turbine's rotor aerodynamics and the demodulated tower nominal model. This conclusion was supported by comparing it to a simulation where the system faced the same wind speeds, but the controller was disabled. This comparison underscores the developed control algorithm's effectiveness as demonstrated in this thesis. Furthermore, the impact of the torque from the velocity-based qLMPC controller on the overall torque (which combines torque from both the torque controller and velocity-based qLMPC) was analyzed. When the velocity-based qLMPC torque is added, the generator torque changes significantly, with a periodic signal superimposed on the original torque signal. This periodic signal is also reflected in the power.

In conclusion, this thesis aimed to design a velocity-based qLMPC controller to completely mitigate tower side-side periodic loads with minimal impact on power production, addressing the main research question posed in the introductory chapter. The main research question was divided into two parts, focusing specifically on designing an efficient velocity-based qLMPC scheme for side-side tower periodic load reductions. The results presented in this thesis demonstrate that the designed velocity-based qLMPC algorithm can considerably reduce the influence of side-side tower periodic loads in wind turbines. However, regarding the impact on power production, it is still not possible to draw accurate conclusions at this stage. The side-side tower periodic load amplitude was artificially set and may not yet be realistic. Therefore, further work is needed to obtain conclusive results on the proposed framework's impact on power production.

7-2 Recommendations

This thesis demonstrates the efforts to establish an efficient velocity-based quasi-Linear Model Predictive Control (qLMPC) framework to address tower side-side periodic loads. However, further improvements can be made, and the following recommendations are provided for future research:

1. An interesting future direction involves conducting high-fidelity simulations using the OpenFAST tool provided by the National Renewable Energy Laboratory (NREL). In this approach,

the control actions derived from the qLMPC, based on the quasi-Linear Parameter-Varying (quasi-LPV) model of the wind turbine, would be integrated into the nominal wind turbine model provided in the simulation tool. The states from the nominal model could then serve as initial states for each optimization algorithm run, similar to the method used in Case 5. This approach would allow for an assessment of the algorithm's performance in the nominal model through high-fidelity simulations, enabling more realistic conclusions regarding the control algorithm's impact on power production;

2. In the current work presented in this thesis, the states and disturbances are assumed to be measurable for simplicity. However, in practical scenarios, these loads are frequently unmeasurable and unknown. Therefore, a recommendation for future work would be to integrate the designed qLMPC control scheme with a Kalman filter for the online estimation of periodic loads. Such integration could significantly enhance the controller's performance by providing real-time load estimates, thus improving the overall effectiveness of the control system.

3. As discussed in Case 5, it is crucial to recognize that drawing precise and definitive conclusions about the velocity-based qLMPC algorithm's impact on power production is not yet feasible. This limitation arises as the side-side tower periodic load amplitude was artificially set and may not accurately reflect realistic conditions. Future research will need to address and refine this aspect to achieve more reliable results.

Appendix A

Simulink block diagram

In this appendix, the reader can find the Simulink block diagram used to perform the simulations in Case 5 from Section 6-5. It is important to mention that this Simulink implementation was mostly obtained from [68], and adapted to the needs of this thesis.

In Figure A-1, the aerodynamic nominal model of the wind turbine rotor is illustrated. The subsystem's output is the aerodynamic torque, calculated using the functions within the *Rotor* block, which are based on Equation (5-4) from Chapter 5. This aerodynamic torque, along with the torque from the torque controller and the velocity-based quasi-linear Model Predictive Control (qLMPC) algorithm, allows for the determination of rotor speed evolution over time using Equation (5-25) in Chapter 5. Within this scheme, one can also analyze the total torque applied to the system and the generated power, which is related to both the torque and rotor speed as per Equation (6-2) in Chapter 6.

As shown in Figure A-2, the demodulated tower nominal model is represented within the *Amplitude Transformation* block, which is subjected to a side-side periodic load. The model in this block is based on Equations (5-21)-(5-24) and is depicted in the bottom block diagram of Figure A-2. In this scheme, the modulated torque signal from the velocity-based qLMPC is defined using Equation (5-26). This torque signal is then multiplied by the constant c from Chapter 5 to convert torque to force, which is then combined with the side-side periodic load signal to determine the net side-side force applied to the wind turbine tower.

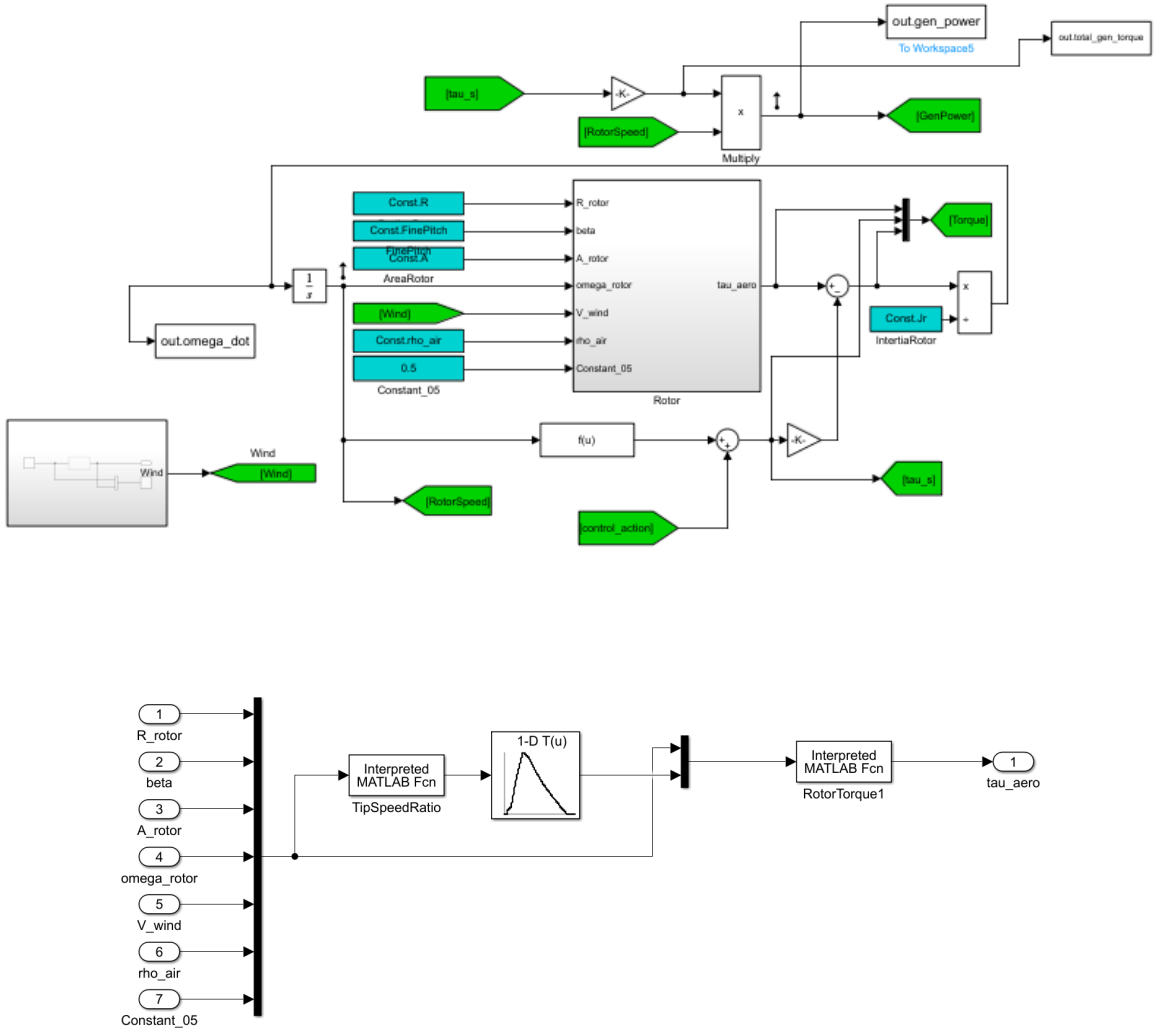

Figure A-1: Top block diagram: Wind turbine rotor's aerodynamic nominal model, with the torque from the torque controller and velocity-based qLMPC. The figure also shows the rotor speed evolution over time. **Bottom block diagram:** Functions inside the rotor block used to compute the aerodynamic torque based on Equation (5-4) from Chapter 5. The output of this subsystem is the aerodynamic torque. By combining this torque with the torque obtained from the torque controller and the velocity-based quasi-linear Model Predictive Control (qLMPC) algorithm, the evolution of the rotor speed over time can be defined using the relation presented in Equation (5-25) in Chapter 5.

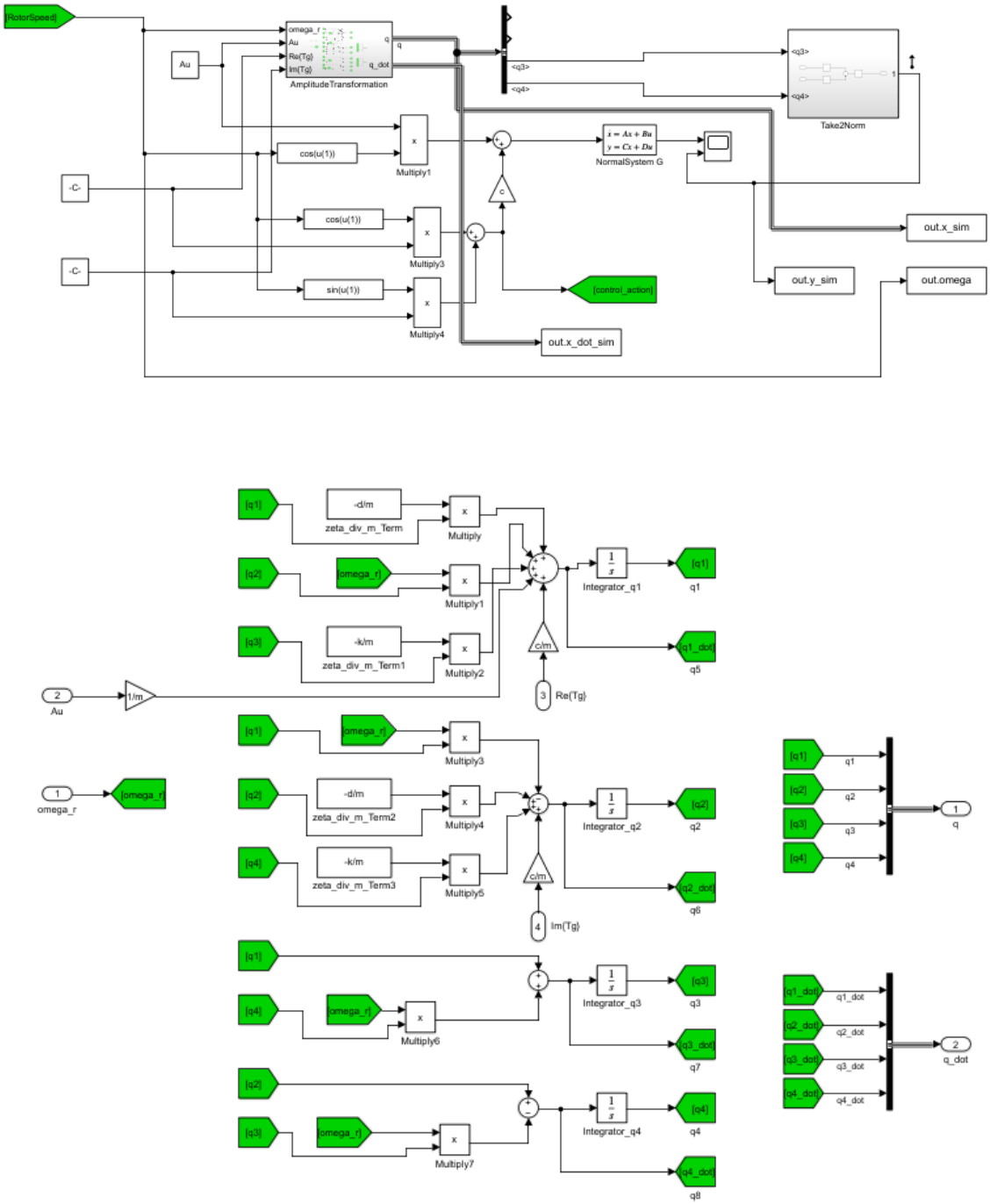

Figure A-2: Top block diagram: Demodulated tower nominal model subjected to the side-side periodic load. The velocity-based qLMPC controller is used to reduce the influence of the side-side periodic load. **Bottom block diagram:** Model within the *Amplitude Transformation* block based on Equations (5-21)-(5-24). In this scheme, the modulated torque signal from the velocity-based qLMPC is defined using Equation (5-26). Then, this torque signal is multiplied by the constant c discussed in Chapter 5 to convert torque to force. Subsequently, the resulting signal is combined with the side-side periodic load signal to calculate the overall side-side force applied to the wind turbine tower, which according to the control objectives should converge to zero.

Bibliography

[1] "Renewable Energy Sources and Climate Change Mitigation — IPCC," May 2024. [Online; accessed 11. May 2024].

[2] "EU renewable electricity has reduced environmental pressures; targeted actions help further reduce impacts," July 2023. [Online; accessed 11. May 2024].

[3] "Climate change mitigation: reducing emissions," May 2024. [Online; accessed 11. May 2024].

[4] Alex, "Global Wind Report 2024 - Record year for wind energy shows momentum but highlights need for policy-driven action - Global Wind Energy Council," *Global Wind Energy Council*, Apr. 2024.

[5] "Study: Levelized Cost of Electricity - Renewable Energy Technologies - Fraunhofer ISE," May 2024. [Online; accessed 11. May 2024].

[6] M. Bolinger, E. Lantz, R. Wiser, B. Hoen, J. Rand, and R. Hammond, "Opportunities for and challenges to further reductions in the "specific power" rating of wind turbines installed in the United States," *Wind Eng.*, vol. 45, pp. 351–368, Jan. 2020.

[7] R. Wiser, D. Millstein, M. Bolinger, S. Jeong, and A. Mills, "The hidden value of large-rotor, tall-tower wind turbines in the United States," *Wind Eng.*, vol. 45, pp. 857–871, July 2020.

[8] "Wind Turbine Lubricant Production | ExxonMobil Product Solutions," May 2024. [Online; accessed 11. May 2024].

[9] "Wind Turbines: the Bigger, the Better," May 2024. [Online; accessed 11. May 2024].

[10] K. Dykes, R. Damiani, O. Roberts, and E. Lantz, "Analysis of Ideal Towers for Tall Wind Applications," *ResearchGate*, Jan. 2018.

[11] G. Sieros, P. Chaviaropoulos, J. D. Sørensen, B. H. Bulder, and P. Jamieson, "Upscaling wind turbines: Theoretical and practical aspects and their impact on the cost of energy," *Wind Energy*, vol. 15, pp. 3–17, Jan. 2012.

[12] "Paper Listing | Wind Power Generation and Wind Turbine Design | WIT Transactions on State-of-the-art in Science and Engineering," May 2024. [Online; accessed 12. May 2024].

[13] A. K. Pamososuryo, "Wind Turbine Load Control and Estimation: Advancements by Coordinate Transformations," 2024. [Online; accessed 13. May 2024].

[14] W. E. Leithead, S. Dominguez, and C. Spruce, "Analysis of tower/blade interaction in the cancellation of the tower fore-aft mode via control," 2004. [Online; accessed 13. May 2024].

[15] S. P. Mulders, T. G. Hovgaard, J. D. Grunnet, and J.-W. van Wingerden, "Preventing wind turbine tower natural frequency excitation with a quasi-LPV model predictive control scheme," *Wind Energy*, vol. 23, pp. 627–644, Mar. 2020.

[16] E. A. Bossanyi, "Wind Turbine Control for Load Reduction," *Wind Energy*, vol. 6, pp. 229–244, July 2003.

[17] A. Gambier and Y. Y. Nazaruddin, "Collective Pitch Control with Active Tower Damping of a Wind Turbine by Using a Nonlinear PID Approach," *IFAC-PapersOnLine*, vol. 51, pp. 238–243, Jan. 2018.

[18] Z. Zhang, S. R. K. Nielsen, F. Blaabjerg, and D. Zhou, "Dynamics and Control of Lateral Tower Vibrations in Offshore Wind Turbines by Means of Active Generator Torque," *Energies*, vol. 7, pp. 7746–7772, Nov. 2014.

[19] D. Duckwitz and M. Geyler, "Active damping of the side-to-side oscillation of the tower," in *Proceedings of DEWEK*, 2010.

[20] B. Fischer and M. Shan, "A survey on control methods for the mitigation of tower loads," *Fraunhofer IWES Report*, vol. 655, 2013.

[21] P. Schaak, G. Corten, and E. Van der Hooft, "Crossing resonance rotor speeds of wind turbines," 2003.

[22] J. Licari, J. B. Ekanayake, and N. Jenkins, "Investigation of a Speed Exclusion Zone to Prevent Tower Resonance in Variable-Speed Wind Turbines," *Sustainable Energy, IEEE Transactions on*, vol. 4, pp. 977–984, Oct. 2013.

[23] E. Smilden and A. J. Sørensen, "Application of a Speed Exclusion Zone Algorithm on a Large 10MW Offshore Wind Turbine," *ASME Digital Collection*, Oct. 2016.

[24] M. A. Evans, M. Cannon, and B. Kouvaritakis, "Robust MPC Tower Damping for Variable Speed Wind Turbines," *IEEE Trans. Control Syst. Technol.*, vol. 23, pp. 290–296, Jan. 2015.

[25] D. Schlipf, D. J. Schlipf, and M. Kühn, "Nonlinear model predictive control of wind turbines using LIDAR," *Wind Energy*, vol. 16, pp. 1107–1129, Oct. 2013.

[26] E. Tofighi, D. Schlipf, and C. M. Kellett, "Nonlinear model predictive controller design for extreme load mitigation in transition operation region in wind turbines," in *2015 IEEE Conference on Control Applications (CCA)*, pp. 1167–1172, 2015.

[27] A. Dittmer, B. Sharan, and H. Werner, "A Velocity quasiLPV-MPC Algorithm for Wind Turbine Control," in *2021 European Control Conference (ECC)*, pp. 1550–1555, IEEE, June 2021.

[28] A. K. Pamososuryo, S. P. Mulders, R. Ferrari, and J.-W. van Wingerden, "Periodic Load Estimation of a Wind Turbine Tower using a Model Demodulation Transformation," in *2022 American Control Conference (ACC)*, pp. 08–10, IEEE.

[29] L. Rieck, B. Herrmann, F. Thielecke, and H. Werner, "Efficient Quasi-Linear Model Predictive Control of a Flexible Aircraft Based on Laguerre Functions," in *2023 American Control Conference (ACC)*, pp. 2855–2860, IEEE, May 2023.

[30] R. Dorf and R. Bishop, *Modern Control Systems*. London, England, UK: Pearson, Jan. 2016.

[31] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ, USA: Princeton University Press, Apr. 2010.

[32] G. F. F. e. A. E.-N. J. Powell, *Feedback Control Of Dynamic Systems, Global Edition*. London, England, UK: Pearson Education Limited, May 2019.

[33] B. Friedland, *Control System Design: An Introduction to State-Space Methods (Dover Books on Electrical Engineering)*. Mineola, NY, USA: Dover Publications, Mar. 2005.

[34] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods (Dover Books on Engineering)*. Mineola, NY, USA: Dover Publications, Feb. 2007.

[35] R. F. Stengel, *Optimal Control and Estimation (Dover Books on Mathematics)*. Mineola, NY, USA: Dover Publications, Sept. 1994.

[36] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, Oct. 2020.

[37] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*. Hoboken, NJ, USA: Wiley-Interscience, Nov. 2005.

[38] J. Hanema, "Anticipative model predictive control for linear parameter-varying systems," *Eindhoven University of Technology research portal*, Oct. 2018.

[39] J. S. Shamma, *Analysis and design of gain scheduled control systems*. PhD thesis, Massachusetts Institute of Technology, 1988.

[40] J. S. Shamma and M. Athans, "Analysis of gain scheduled control for nonlinear plants," *IEEE Trans. Autom. Control*, vol. 35, pp. 898–907, Aug. 1990.

[41] *Control of Linear Parameter Varying Systems with Applications*. New York, NY, USA: Springer.

[42] P. S. Gonzalez Cisneros, "Quasi-Linear Model Predictive Control: Stability, Modelling and Implementation," 2021. [Online; accessed 18. Aug. 2023].

[43] A. Kwiatkowski, M.-T. Boll, and H. Werner, "Automated generation and assessment of affine lpv models," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 6690–6695, 2006.

[44] R. Toth, J. C. Willems, P. S. C. Heuberger, and P. M. J. Van den Hof, "The behavioral approach to linear parameter-varying systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2499–2514, 2011.

[45] C. Hoffmann and H. Werner, "Complexity of Implementation and Synthesis in Linear Parameter-Varying Control," *IFAC Proceedings Volumes*, vol. 47, pp. 11749–11760, Jan. 2014.

[46] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[47] C. Hoffmann and H. Werner, "A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 416–433, 2015.

[48] P. S. G. Cisneros, A. Sridharan, and H. Werner, "Constrained Predictive Control of a Robotic Manipulator using quasi-LPV Representations," *IFAC-PapersOnLine*, vol. 51, pp. 118–123, Jan. 2018.

[49] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control—Part I. The basic algorithm," *Automatica*, vol. 23, pp. 137–148, Mar. 1987.

[50] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized Predictive Control—Part II Extensions and interpretations," *Automatica*, vol. 23, pp. 149–160, Mar. 1987.

[51] Pablo. S. G. Cisneros, A. Datar, P. Göttsch, and H. Werner, "Data-Driven quasi-LPV Model Predictive Control Using Koopman Operator Techniques," *IFAC-PapersOnLine*, vol. 53, pp. 6062–6068, Jan. 2020.

[52] "Choose Sample Time and Horizons - MATLAB & Simulink," Oct. 2023. [Online; accessed 4. Oct. 2023].

[53] P. S. G. Cisneros, S. Voss, and H. Werner, "Efficient Nonlinear Model Predictive Control via quasi-LPV representation," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 3216–3221, IEEE, Dec. 2016.

[54] E. F. Camacho and C. Bordons, "Nonlinear Model Predictive Control: An Introductory Review," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, pp. 1–16, Berlin, Germany: Springer, 2007.

[55] Y. Zhang, S. Li, and L. Liao, "Near-optimal control of nonlinear dynamical systems: A brief survey," *Annual Reviews in Control*, vol. 47, pp. 71–80, Jan. 2019.

[56] H. G. Bock, M. Diehl, P. Kühl, E. Kostina, J. P. Schiöder, and L. Wirsching, "Numerical Methods for Efficient and Fast Nonlinear Model Predictive Control," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, pp. 163–179, Berlin, Germany: Springer, 2007.

[57] P. G. Cisneros and H. Werner, "Fast Nonlinear MPC for Reference Tracking Subject to Nonlinear Constraints via Quasi-LPV Representations," *IFAC-PapersOnLine*, vol. 50, pp. 11601–11606, July 2017.

[58] P. S. G. Cisneros and H. Werner, "Stabilizing Model Predictive Control for Nonlinear Systems in Input-Output quasi-LPV Form," *ResearchGate*, Oct. 2019.

[59] P. S. G. Cisneros and H. Werner, "A Velocity Algorithm for Nonlinear Model Predictive Control," *IEEE Trans. Control Syst. Technol.*, vol. PP, pp. 1–6, Apr. 2020.

[60] D. J. Leith and W. E. Leithead, "Gain-scheduled and nonlinear systems: Dynamic analysis by velocity-based linearization families," *Int. J. Control*, vol. 70, pp. 289–317, Jan. 1998.

[61] E. Alcalá, V. Puig, and J. Quevedo, "LPV-MPC Control for Autonomous Vehicles," *IFAC-PapersOnLine*, vol. 52, pp. 106–113, Jan. 2019.

[62] L. Cavanini, G. Ippoliti, and E. F. Camacho, "Model Predictive Control for a Linear Parameter Varying Model of an UAV," *J. Intell. Rob. Syst.*, vol. 101, pp. 1–18, Mar. 2021.

[63] J. Åkesson and P. Hagander, "Integral action — A disturbance observer approach," in *2003 European Control Conference (ECC)*, pp. 01–04, IEEE.

[64] K. Selvam, S. Kanev, J. W. van Wingerden, T. van Engelen, and M. Verhaegen, "Feedback–feedforward individual pitch control for wind turbine load reduction," *Int. J. Robust Nonlinear Control*, vol. 19, pp. 72–91, Jan. 2009.

[65] A. Keyvani Janbahan, M. S. Tamer, J. W. van Wingerden, J. F. L. Goosen, and A. van Keulen, "A comprehensive model for transient behavior of tapping mode atomic force microscope," *Nonlinear Dynamics: an international journal of nonlinear dynamics and chaos in engineering systems*, vol. 97, no. 2, 2019.

[66] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-MW Reference Wind Turbine for Offshore System Development," Feb. 2009. [Online; accessed 16. May 2024].

[67] "TurbSim," May 2024. [Online; accessed 24. May 2024].

[68] S. P. Mulders, T. G. Hovgaard, J. D. Grunnet, and J.-W. van Wingerden, "Preventing wind turbine tower natural frequency excitation with a quasi-lpv model predictive control scheme," *Wind Energy*, vol. 23, no. 3, pp. 627–644, 2020.