A Robust Data-Driven Model Predictive Control Approach Towards Feedback Linearisation of Nonlinear Mechanical Systems

Floren, Merijn M.P.

Award date:
2021

Department of Mechanical Engineering

Systems & Control Master Thesis

A Robust Data-Driven Model Predictive Control Approach Towards Feedback Linearisation of Nonlinear Mechanical Systems

Author:
M.M.P. (Merijn) Floren 0950600

Committee:
Dr. Ir. T.A.E. Oomen *Chair*
Dr. Ir. K. Tiels *Member*
Dr. Ir. A.Y. Pogromskiy *Member*
Dr. Ir. J.M.M.G. Noël *Advisor*
Ir. K.H.J. Classens *Advisor*

Supervisors:
Dr. Ir. J.M.M.G. Noël
Dr. Ir. T.A.E Oomen

TU/e — Control Systems Technology
Eindhoven, March 22, 2021

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date 3/22/21

Name Merijn Floren

ID-number 0950600

Signature 

Insert this document in your Master Thesis report (2nd page) and submit it on Sharepoint

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

Abstract

Classical feedback linearisation is a commonly used approach to control nonlinear systems. By means of a nonlinear state feedback control law, the input-output (IO) response of a nonlinear system is transformed into an equivalent (LTI) system. By doing so, well-known linear control techniques can be applied in order to control the nonlinear system. The main limitation of classical feedback linearisation is that it heavily relies on accurate nonlinear models, which are typically obtained by means of first-principle-based techniques. Yet, in times where systems are becoming increasingly complex, first-principle modelling is a resource-consuming and often an impossible task. Another drawback of classical feedback linearisation is that the IO response of any nonlinear system is transformed into a chain of integrators. This is non-intuitive from an engineering point of view, since the linear dynamics typically contain important system properties. This thesis presents a novel approach towards feedback linearisation, which differs from the classical approach in three ways: (a) the models in this work are obtained by exclusively processing IO data; (b) the important linear dynamics are preserved; and (c) the underlying methodology exploits model predictive control (MPC). Yet, both methods have in common that the IO response of nonlinear systems is transformed into an equivalent LTI system.

An internal reference tracking framework is introduced to achieve the posed objective. For any outer-loop input, the plant input is modified, such that an online generated linear output reference is tracked. This output reference is precisely the response of the desirable linear part of the model to the new outer-loop input. An MPC controller is proposed to solve the reference tracking problem. The MPC controller has integral action in the sense that robust convergence can be achieved in case of model uncertainty and imperfect state information. Here, the states are estimated by means of the unscented Kalman filter.

The data-driven models of this work are obtained by means of a three-step identification procedure: first, a nonparametric best linear approximation (BLA) is estimated from averaged frequency-response functions. Next, the BLA is parameterised by means of the frequency-domain subspace method. Lastly, the parametric BLA is used to initialise a weighted least-squares cost function, in order to obtain the full nonlinear model.

An extensive set of simulation examples is exploited in order to demonstrate the performance and robustness properties of the presented framework. Ultimately, the correct working of the entire control structure is validated on an experimental setup. A nonparametric nonlinear distortion analysis is introduced in order to quantify the performance of the framework over the entire frequency range of interest. The simulation and experimental results are of excellent quality and are therefore promising in view of the applicability of the proposed methodology on more real-life setups.

Contents

Chapter 1

Introduction

Along with the rapid development of modern society, industrial systems are becoming increasingly complex, and their degree of nonlinearity is becoming increasingly high. Consider for instance the aerospace sector, which meets the ever-more present demand for lighter aircraft structures in order to reduce fuel consumption, increase payload, and increase flight range [1]. An inevitable result is that these lighter and more flexible structures are inherently more nonlinear [2]. Obviously, these nonlinear phenomena can only be captured by nonlinear models. In terms of control however, it is much more appealing to work with linear models, since linear equations simplify control design, controller implementation and performance analysis. In order to meet the engineers' desire, many research efforts amongst a range of disciplines focus on finding linear approximations of nonlinear dynamics, which allows for the exploitation of linear control design techniques to control nonlinear systems. A commonly used technique is feedback linearisation [3–5], which intends to find a direct linear relationship between the system output and a new control input. Figure 1.1 illustrates the simplified working of feedback linearisation for a single-input single-output (SISO) nonlinear system. In essence, the plant input $u(t)$ is modified, through a choice of a nonlinear state feedback control law, such that the input-output (IO) relation between the new outer-loop input $v(t)$ and system output $y(t)$ is linear. This linear relation holds for the entire specified operating range, which means that, technically, the IO response of an equivalent linear time-invariant (LTI) system is obtained. Subsequently, well-known linear control techniques can be used to design an outer-loop controller that is stable over the entire operating range and exhibits the same transient and steady-state behaviour for different set-points, e.g., the PID controller of Figure 1.1. In a recent study [6], feedback linearisation was used in combination with an outer-loop linear quadratic regulator in order to control the desired altitude and attitude levels of a quadrotor. In [7], feedback linearisation was combined with model predictive control (MPC) to track a given power reference of a nonlinear wind turbine. A comparison between feedback linearisation and gain scheduling was drawn in [8], where feedback linearisation in combination with a PI controller showed significant improvements in terms of performance and robustness when a nonlinear boiler-turbine unit was controlled.

However, since the introduction of feedback linearisation in the early eighties, its promise has not been solidly realised. The main reason for this is that the underlying methodology heavily relies on accurate nonlinear models, which are significantly more difficult to obtain than linear models. Modelling is usually achieved by applying first-principle-based methods. Yet, due to the ever-increasing degree of nonlinearity in these systems, first-principle modelling appears to be ever more resource-consuming. Another limitation of feedback linearisation is that exact IO linearisation can only be achieved for either continuous-time systems with continuous-time

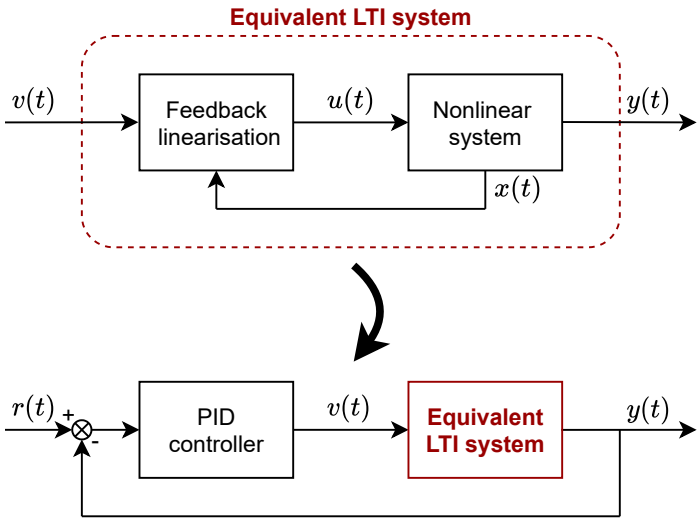


Figure 1.1: Block diagram illustrating the simplified working of feedback linearisation. The plant input $u(t)$ is modified such that for the new input $v(t)$ a linear output response is obtained. By doing so, the equivalent LTI system can be integrated in any linear outer-loop control scheme, in this example a classical PID feedback loop with reference tracking.

controllers or discrete-time systems with discrete-time controllers. Most mechanical control systems are however sampled-data systems, i.e., a control system in which a continuous-time plant is controlled with a digital device. For these kind of systems, research has shown that the effect of sampling imposes severe restrictions [9] or may even result in the impossibility to linearise [10]. Moreover, in cases where the relative degree of the system is smaller than the system order, the state equation is only partially linearised [4], meaning that there may be some remaining nonlinear dynamics in the system which do not have a direct influence on the IO relation. Yet, these remaining dynamics can cause internal instability, so global stability of the internal dynamics must be guaranteed. This can be achieved by considering input-to-state stability [11]. However, this may result in additional bounds on the outer-loop input $v(t)$.

In the present thesis, the conflicting situation between ever-increasing nonlinearities and the engineers' desire to work with linear tools is solved for by means of a novel approach towards feedback linearisation. Specifically, the focus of this work is on nonlinear mechanical vibrating systems. The effectiveness of the linearisation is analysed in the frequency domain over the specified range of interest. In order to overcome the complex task of first-principle modelling, the models in this work are solely identified based on experimental IO data. An extensive set of simulation examples is exploited in order to demonstrate the performance and robustness properties of the solution. Finally, the linearising control structure is validated on an experimental setup.

1.1 Background

Nonlinear mechanical systems, albeit in an implicit form, can be considered as LTI systems with a feedback path containing static nonlinearities [12], or in short, the so-called Lur'e form [13]. A visualisation is shown in Figure 1.2. Consider, for instance, the mass-spring-damper system with a cubic spring nonlinearity

$$m\ddot{y}(t) + c_l\dot{y}(t) + k_l y(t) + k_c y^3(t) = u(t), \tag{1.1}$$

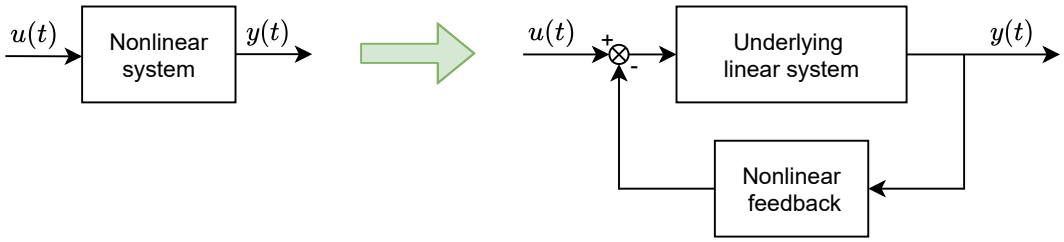


Figure 1.2: Visualisation of the Lur'e system. A nonlinear mechanical system can be considered as an underlying linear system with a static nonlinear output function wrapped around it in feedback.

where the output $y(t)$ is the displacement of the mass m, while the input $u(t)$ is a force exiting the mass. Moreover, c_l and k_l are the linear damping and spring constants, respectively; k_c is the cubic spring coefficient and the over-dot indicates a derivative with respect to the time variable t. The dynamics can easily be rewritten to obey the Lur'e, namely

$$m\ddot{y}(t) + c\dot{y}(t) + k_1 y(t) = u(t) - k_2 y^3(t), \tag{1.2}$$

which suggests that the cubic spring nonlinearity acts as an additional input to the underlying linear system. A system that can be represented in the Lur'e form allows for a more careful selection of nonlinear basis functions for nonlinear system identification. That is to say, it is known that the nonlinearities are a function of the output only. In this way, there is no need for overly complex monomial combinations of the states, input and output in the basis functions. Furthermore, systems that can be represented in the Lur'e form allow for an easy distinction between underlying linear dynamics and undesirable nonlinearities. As for vibrating systems, the linear dynamics are stable and contain the important vibration properties of the system. In this study, all systems obey the Lur'e form.

1.2 Problem Statement

The effectiveness of feedback linearisation is mainly dictated by the accuracy of the nonlinear model, which is typically obtained by exploiting first-principle-based techniques. However, for complex systems, it can be extremely resource-consuming or even impossible to apply these techniques. To overcome this problem, systems are often identified by exclusively processing IO data, resulting in so-called data-driven models. System identification of nonlinear mechanical systems has gained significant popularity over the past two decades [14, 15], mainly driven by the growing industrial need for these more complex models. In line with the developments in research and industry, the models in this work are purely data-driven. More specifically, the focus is on discrete-time SISO polynomial nonlinear state-space models, defined at time instant $k \in \mathbb{N}$ as

$$\begin{cases} x(k+1) = \boldsymbol{A}x(k) + \boldsymbol{B}u(k) + \boldsymbol{E}\zeta(y(k)) \\ \quad y(k) = \boldsymbol{C}x(k) + \boldsymbol{D}u(k), \end{cases} \tag{1.3}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^n$, $\boldsymbol{C} \in \mathbb{R}^{1\times n}$ and $\boldsymbol{D} \in \mathbb{R}$ are the linear state, input, output and direct feedthrough matrices, respectively, with n being the model order. Furthermore, $x(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}$ the plant input and $y(k) \in \mathbb{R}$ the plant output. Moreover, $\zeta(y(k)) \in \mathbb{R}^s$ is a vector containing the s monomial terms of the nonlinear polynomial basis function, with $\boldsymbol{E} \in \mathbb{R}^{n\times s}$ as the associated coefficient matrix. The model in the proposed form of (1.3) corresponds to the Lur'e form of Figure 1.2, with $\boldsymbol{E}\zeta(y(k))$ being the undesired static nonlinearity in the feedback loop.

As mentioned, this work presents a novel approach towards feedback linearisation that overcomes the main deficiencies of the classical method. In order to make a valid comparison in the end, it is first necessary to present a slightly more detailed introduction of classical feedback linearisation. Classical feedback linearisation is well-defined for SISO nonlinear systems which can be represented by the following model class:

$$\begin{cases} \dot{x}(t) = \boldsymbol{f}(x(t)) + \boldsymbol{g}(x(t))u(t) \\ y(t) = \boldsymbol{h}(x(t)), \end{cases} \tag{1.4}$$

where $\boldsymbol{f}(\cdot)$, $\boldsymbol{g}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are (non)linear functions of the state vector $x(t)$. By exploiting the notion of so-called Lie derivatives [16], the nonlinear state feedback control law converts the IO dynamics of any nonlinear system that obeys the form of (1.4) into a chain of ρ integrators, with ρ being the relative degree of the system. Here, the definition of the relative degree differs from the definition of the relative degree of an LTI system, where it is simply the difference between the number of poles and the number of zeros. As for feedback linearisation, the notion of relative degree is defined as the number of times the output needs to be differentiated until the input explicitly appears [4]. The classical approach claims to eliminate all nonlinearities *exactly*, which is however only possible in case of a perfect model and perfect state information. Of course, this is never the case for any real-life system, which means that inaccuracies are inevitable. To overcome this shortcoming, several studies [17–20] proposed a robust feedback linearising controller by simultaneously designing an outer-loop controller that has integral action. In this way, outer-loop setpoints can still be tracked with zero error. However, this is more or less working around the inevitable inaccuracies of classical feedback linearisation rather than truly proposing a robust *linearising* controller.

Yet, it is still interesting to consider the underlying machinery of classical feedback linearisation for data-driven models. Namely, one could ask if, under perfect conditions, exact linearisation could be achieved for data-driven models. In order to answer this question, the differences between models (1.3) and (1.4) need to be investigated first. A threefold comparison can be established between the three: (a) model (1.3) is in discrete time whereas (1.4) is in continuous time; (b) the nonlinearity in (1.3) is a function of the output while it is a function of the states in (1.4); and (c) model (1.3) contains direct feedthrough whereas (1.4) does not. However, if direct feedthrough is ignored for the moment, the continuous-time equivalent of the data-driven model can be rewritten such that it obeys the structure of (1.4) perfectly. Namely, since $y(t) = \boldsymbol{C}x(t)$, $\boldsymbol{A}_c x(t) + \boldsymbol{E}_c \zeta(y(t)) = \boldsymbol{f}(x(t))$, $\boldsymbol{B}_c = \boldsymbol{g}(\cdot)$ and $\boldsymbol{C}x(t) = \boldsymbol{h}(x(t))$; here, the subscript c denotes the continuous-time equivalent of a discrete-time state-space matrix.

Now, by the notions of Lie algebra, it *is* possible to perfectly linearise an IO response by using a data-driven model. However, it is not reliable in practice, since data-driven models induce significant potential problems. This inherently follows from the fact that all matrix entries of these models are nonzero. A first problem is that a single integrator is *always* obtained after linearisation, regardless of the model order. This directly follows from the definition of the relative degree; namely, the input always appears after differentiating the output once. Concretely speaking, if one would linearise the relation between the input force and output displacement of a mass-spring-damper system with hardening spring as in (1.2), while using the first-principle-based model, the linearised IO dynamics would be described by a *double* integrator. Now, if a data-driven model for the exact same system would have been used, the linearised IO dynamics would have been described by a *single* integrator. Another significant problem is that, due to the full matrices, every nonlinearity affects every state; while at the same time, the output is a combination of all the states. To eliminate all undesired nonlinearities, a very complex and non-intuitive state feedback control law would be required. One can imagine that

the complexity of this control law quickly increases when the model order and/or the number of monomials in the nonlinear polynomial grow. In addition to being complex, the required control law might also lead to undesirably large actuation signals. For example, an extremely high input amplitude may be required to exactly eliminate a nonlinearity that only marginally affects the output. In the present form, there is no guarantee that physically realistic inputs will be generated. So, although it is mathematically feasible to exactly linearise a data-driven model, the procedure is far from intuitive, and, more problematically, the obtained control law is extremely complex and unpredictable.Concluding, data-driven models allow for capturing more complex dynamics more precisely. However, this comes at a price, since they are not suitable to be used for classical feedback linearisation.

1.3 Research Objective

As mentioned, a nonlinear system represented in the Lur'e form allows for a more careful selection of the nonlinear basis functions. Moreover, the feedback structure also allows for the easy distinction between underlying linear dynamics and undesired nonlinearities in the system. From an engineering point of view, it is sensible to preserve the vibrating properties of a mechanical system, which are captured in the linear dynamics. With the latter in mind, the main objective of this study is formulated as follows:

Robustly eliminate all undesired nonlinearities in the IO response of nonlinear mechanical systems, while preserving the stable underlying linear dynamics.

So, in addition to the fact that data-driven models are used, the objective clearly differs from classical feedback linearisation in the sense that the underlying linear dynamics are preserved. Moreover, the linearising solution is intended to be robust to modelling errors and imperfect state information, which is something that cannot be directly achieved with classical feedback linearisation. Yet, both methods have in common that the IO response of a nonlinear system is transformed into an equivalent LTI system, in accordance with Figure 1.1. In order to make the research objective more manageable, it is divided in the following sub-questions:

(i) How to obtain accurate data-driven nonlinear models?

(ii) Which kind of control method is most suitable to achieve the posed objective?

(iii) How to ensure that the controller is robust to modelling errors and imperfect state information?

(iv) How to quantify the effectiveness of the linearisation over the entire operating range?

1.4 Proposed Solution

An internal reference tracking framework is proposed to achieve the research objective. The reference is generated by exploiting the discrete-time property of the considered model (1.3). That is to say, a discrete-time model always allows for the construction of a one-step-ahead reference point, for any outer-loop input $v(k)$. This reference point, which is generated online at any time instant k, is precisely the one-step-ahead output response of the desirable linear part of the model to the new outer-loop input $v(k)$. Specifically, the one-step-ahead reference point at time instant k is defined as follows

$$\begin{cases} x_{ref}(k+1) = \boldsymbol{A}x_{ref}(k) + \boldsymbol{B}v(k) \\ y_{ref}(k+1) = \boldsymbol{C}x_{ref}(k+1), \end{cases} \tag{1.5}$$

where $x_{ref}(k+1)$ is the state reference and $y_{ref}(k+1)$ is the output reference to be tracked. Moreover, $v(k) \in \mathbb{R}$ is the new input that renders the linear IO relation, in correspondence with Figure 1.1. So, a controller that is capable of perfectly tracking this one-step-ahead reference point at any time k for any input $v(k)$, will generate an IO relation between $v(k)$ and $y(k)$ that is governed by the linear part of the identified model, in this way cancelling out the undesired nonlinearities. Note that here, a slight modification is made regarding the direct feedthrough matrix, which is linear as well. Typically, a mechanical system does not contain any direct feedthrough, but due to the nature of the data-driven identification procedure, there will still be a non-zero value assigned to the $\boldsymbol{D}$ matrix. This can generally be considered as undesirable; for this reason, it is chosen to eliminate it from the IO response.

An MPC [21–25] solution is proposed to achieve the tracking objective. The choice for MPC is a self-evident one, since the predictive nature of MPC is fully exploited as the controller must always anticipate on future references. In the described form, the prediction horizon is equal to one. Although such a short prediction horizon is often seen in the field of power electronics [26–29], it is almost never used for systems with slower dynamics. As for these systems, in order to obtain a stabilising MPC controller, the sample time must be sufficiently low while the prediction horizon must be sufficiently high. Common practice for improving performance is to decrease the sample time and/or increase the prediction horizon, at the cost of computational complexity. With this in mind, a framework is introduced where the inner-loop (the linearising MPC) runs at a higher sample rate than the outer-loop (any linear controller). More specifically, this means that the outer-loop input signal $v(k)$ is supplied at a lower rate than the execution rate of the MPC. In this way, the one-step-ahead reference *point* can be reached by following a reference *trajectory*, with smaller intermediate time steps. By doing so, the sample time is obviously decreased whilst the prediction horizon length is

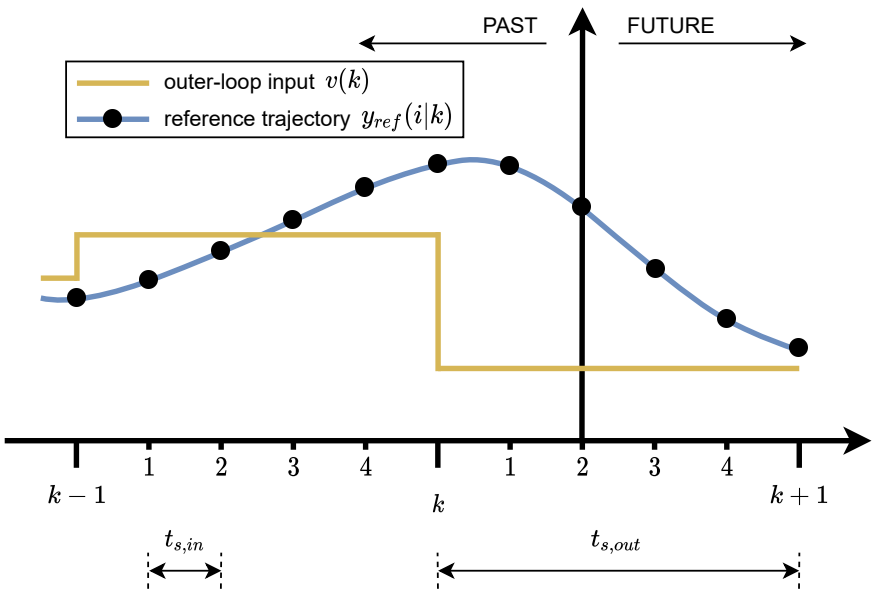


Figure 1.3: Visualisation of the proposed MPC solution with online reference generation and dynamic prediction horizon length. The notation $y_{ref}(i|k)$ indicates the output reference point at the ith inner-loop sub-sample of the kth outer-loop time instant. At the current time instant, there are three out of five future reference available. The number of available references reduces to one as time progresses up until time $k+1$, where a new outer-loop input will allow for generation of five new reference points.

increased. A consequence of the proposed solution is that the prediction horizon has a *dynamic* length, as a result of the ever-changing number of available reference points. A visualisation of this phenomenon is provided in Figure 1.3, where the MPC controller is currently at the second sub-sample of time instant k, corresponding to time $t = kt_{s,out} + 2t_{s,in}$, where $t_{s,out}$ and $t_{s,in}$ are the outer and inner-loop sample times respectively. In the sequel, the notation $t(i|k)$ indicates the ith inner-loop sub-sample of the kth outer-loop time instant. Moreover, $t(N_{p,\max}|k) = t(0|k+1) = t(k+1)$, where $N_{p,\max} = t_{s,out}/t_{s,in} \in \mathbb{N}$. At the current time instant in the figure, there are only 3 out of the 5 future reference points available. The number of available reference points will reduce to 1 as time progresses up until time $k+1$; at this point, a new outer-loop input will be available which allows for the generation of 5 new reference points. In general, the future references according to Figure 1.3 are generated as

$$\begin{cases} x_{ref}(i+1|k) = \boldsymbol{A}x_{ref}(i|k) + \boldsymbol{B}v(i|k) \\ y_{ref}(i+1|k) = \boldsymbol{C}x_{ref}(i+1|k), \end{cases} \tag{1.6}$$

where $x_{ref}(i+1|k)$ and $y_{ref}(i+1|k)$ are the future state and output sub-reference points based on the outer-loop input $v(i|k) = v(k)$, for $i \in \{0, \ldots, N_{p,\max} - 1\}$. Moreover, note that $\boldsymbol{A}$ and $\boldsymbol{B}$ here correspond to sample time $t_{s,in}$ as opposed to $t_{s,out}$ in (1.5).

Throughout literature, MPC is often used as an outer-loop controller in combination with feedback linearisation, for example in [7, 30–34]. Yet, it is never used as the linearising controller itself. Also, an MPC controller with dynamic prediction horizon length as proposed here is not encountered in literature. Nevertheless, there are studies that consider time-varying prediction horizons, but they appear in a specific context. In [35] for example, the prediction horizon length is dynamically adapted such that it guarantees stabilisation with the smallest horizon possible, in this way saving computational expenses. Another example is found in [36], where the length of the future reference signal, i.e., the prediction horizon length, is determined by time-varying delays of arriving packets. Lastly, in [37], the prediction horizon length is dynamically optimised by minimising a cost function that takes into account MPC preferences (longer horizons are typically preferred) and the accuracy of the available reference signal (shorter horizons may be preferred in case of an inaccurate reference). To overcome one of the main deficiencies of traditional feedback linearisation, the proposed linearising MPC solution has integral action in the sense that robust convergence can be achieved in case of modelling errors and imperfect state information. In other words, the reference output must always be tracked with the smallest error possible and zero error should be achieved for constant setpoints. There exist two intuitive approaches to achieve offset-free tracking for linear MPC, namely the state disturbance approach [38–40] and the velocity form approach [41–43]. Both methods are closely related and employ an augmented state-space representation of the original model. More advanced studies exist regarding offset-free tracking of nonlinear MPC, see for example [44–46]. The problem is that these nonlinear solutions require an online optimisation problem to be solved at every time instant. Online optimisation is not suitable for the proposed framework, since the low-level linearising MPC runs at a relatively high sample rate. Moreover, this would make the comparison with classical feedback linearisation less consistent. Therefore, the rationale behind the linear offset-free methods will be extended to the nonlinear framework, such that no online optimisation is required. This also means that the present research does not impose any constraint bounds on the input or output signals, since this would similarly require online optimisation.

1.5 Outline

This work is outlined as follows: first, Chapter 2 describes the details of the entire novel control strategy. Here, the necessary building blocks for the linearising MPC are presented, which also includes the nonlinear state observer. Moreover, the nonparametric analysis that detects, qualifies and quantifies the nonlinear distortions in the frequency domain is introduced. This nonparametric framework is used throughout this thesis in order to analyse the effectiveness of linearisation over the entire frequency range of interest. The last part of Chapter 2 concerns the nonlinear data-driven modelling procedure. In Chapter 3, a rich set of simulation examples is presented in order to demonstrate the performance and robustness properties of the proposed solution. Next, in Chapter 4, the complete control structure is validated by means of experimental tests on a real-life setup. Finally, conclusions are drawn and recommendations for future works are presented in Chapter 5.

Chapter 2

MPC-Based Feedback Linearisation: Control Design and Data-Driven Aspects

This chapter describes the details of the entire proposed framework, of which a schematic overview is provided in Figure 2.1. First, the MPC with integral action is elaborated upon. This concerns the underlying machinery of the MPC, a discussion on how to select the inner and outer-loop sample times, and a high-level description of the nonlinear state estimator, i.e, unscented Kalman filter (UKF). The second part of this chapter describes the data-driven modelling and analysis aspects. First, the random-phase multisine excitation signal is introduced, being one of the key components of this thesis. Next, the nonparametric nonlinear distortion analysis is presented, which allows for the quantification of linearising performance over the entire frequency range of interest. Lastly, the data-driven modelling procedure is introduced. This procedure consists of three consecutive steps: (1) a nonparametric best linear approximation (BLA) is estimated from averaged frequency-response functions; (2) the BLA is parameterised by means of the frequency-domain subspace method; and (3) the parametric BLA is used to initialise a weighted least-squares cost function, in order to obtain the full nonlinear model.

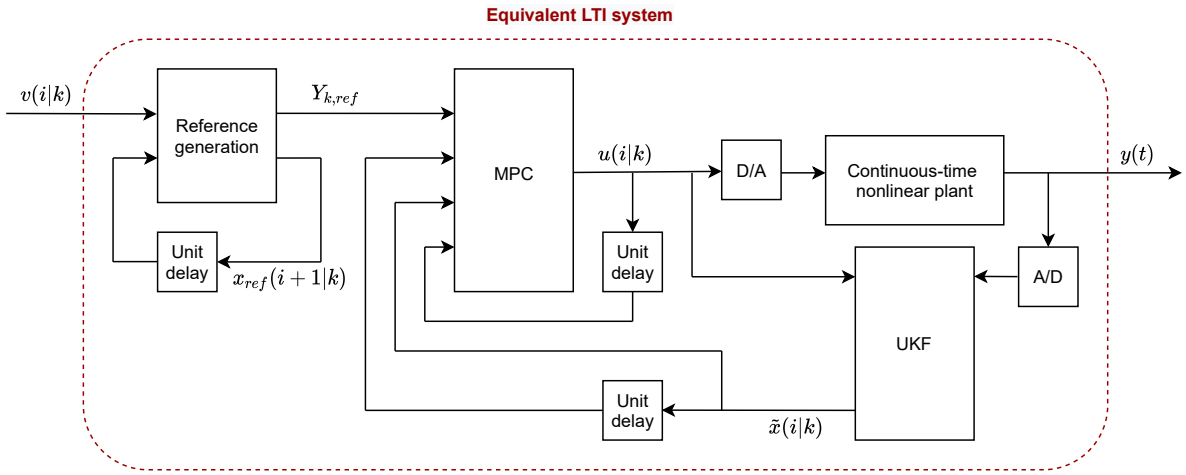


Figure 2.1: Schematic overview of internal reference tracking framework.

2.1 Linearising MPC with Integral Action

An MPC controller is proposed to linearise the IO response of the nonlinear system. Typically, MPC is used to track state references while at the same time assuring that constraints on the input and the states are not violated. In this study however, the states are of less importance since they do not have a physical meaning, because the model is identified on IO data only. Constraint bounds on the input could be more of an interest. However, here is chosen not to incorporate them since this would require an online optimisation problem to be executed at every time step, which would significantly increase the computational load. Moreover, the reference to be tracked already resembles the most dominant dynamics of the nonlinear system; so, the controller only corrects for the undesired nonlinearities. Therefore, it is safe to say that these corrections do not require excessive input signals. As mentioned, the proposed linearising MPC controller has integral action in the sense that robust convergence can be achieved in case of modelling errors and imperfect state information. To achieve this, the original model must be augmented such that the state equation also contains the current measured output, as inspired by the velocity form approach in [41–43]. The velocity form approach is however designed for linear systems, whereas a full nonlinear model is considered here. Yet, by utilising the available information of the reference signal, the nonlinear MPC problem can be transformed into a conventional linear MPC with a disturbance matrix.

Consider again the nth order SISO discrete-time full nonlinear model

$$\begin{cases} x(i+1|k) = \boldsymbol{A}x(i|k) + \boldsymbol{B}u(i|k) + \boldsymbol{E}\zeta(y(i|k)) \\ y(i|k) = \boldsymbol{C}x(i|k) + \boldsymbol{D}u(i|k), \end{cases} \tag{2.1}$$

corresponding to sample time $t_{s,in}$. In order to augment the state-space, the following relations are defined first:

$$\begin{cases} \Delta x(i|k) & := x(i|k) - x(i-1|k) \\ \Delta u(i|k) & := u(i|k) - u(i-1|k) \\ \Delta\zeta(y(i|k)) & := \zeta(y(i|k)) - \zeta(y(i-1|k)). \end{cases} \tag{2.2}$$

Next, the state-space model (2.1) in augmented form writes

$$\begin{cases} \bar{x}(i+1|k) = \underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{0}_{n\times 1} \\ \boldsymbol{CA} & 1 \end{bmatrix}}_{\bar{\boldsymbol{A}}} \bar{x}(i|k) + \underbrace{\begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{CB} \end{bmatrix}}_{\bar{\boldsymbol{B}}} \Delta u(i|k) + \underbrace{\begin{bmatrix} \boldsymbol{E} \\ \boldsymbol{CE} \end{bmatrix}}_{\bar{\boldsymbol{E}}} \Delta\zeta(y(i|k)) \\ y(i|k) = \underbrace{\begin{bmatrix} \boldsymbol{0}_{1\times n} & 1 \end{bmatrix}}_{\bar{\boldsymbol{C}}} \bar{x}(i|k) + \boldsymbol{D}u(i|k), \end{cases} \tag{2.3}$$

where $\bar{x}(i|k) = \begin{bmatrix} \Delta x(i|k)^\top & y(i|k) \end{bmatrix}^\top$. At time $t(i|k)$, the predicted future outputs of (2.3) over prediction horizon N_p, i.e.,

$$Y_k = [y(i+1|k), \ldots, y(i+N_p|k)]^\top \in \mathbb{R}^{N_p}, \tag{2.4}$$

is a function of the current state, input sequence, disturbance sequence and previous plant input; it is defined as

$$Y_k = S_x\bar{x}(i|k) + (S_u + \boldsymbol{D}\mathcal{E})\Delta U_k + S_g\Delta G_k + \boldsymbol{D}\mathcal{M}u(i-1|k), \tag{2.5}$$

where S_x, S_u, S_g, $\mathcal{E}$ and $\mathcal{M}$ are all auxiliary matrices that describe the evolution of the states and output over the prediction horizon. A detailed derivation of these matrices is provided in Appendix A. Furthermore,

$$\Delta U_k = \begin{bmatrix} \Delta u(i|k) \\ \vdots \\ \Delta u(i+N_p-1|k) \end{bmatrix} \in \mathbb{R}^{N_p}; \quad \Delta G_k = \begin{bmatrix} \Delta\zeta(y(i|k)) \\ \vdots \\ \Delta\zeta(y(i+N_p-1|k)) \end{bmatrix} \in \mathbb{R}^{sN_p}, \tag{2.6}$$

where ΔU_k is the vector to be optimised and ΔG_k is the vector of future disturbances. Here, the word disturbance is used because the elements of ΔG_k cannot be controlled. In fact, there is the conflicting situation between ΔG_k and Y_k since they are mutually dependent on each other's information. Fortunately, ΔG_k can very well be estimated in advance, by substituting the available reference information of (1.6) for the future outputs. Thus, the estimate of ΔG_k becomes

$$\Delta\widetilde{G}_k = \begin{bmatrix} \zeta(y(i|k)) - \zeta(y(i-1|k)) \\ \zeta(y_{ref}(i+1|k)) - \zeta(y(i|k)) \\ \vdots \\ \zeta(y_{ref}(i+N_p-1|k)) - \zeta(y_{ref}(i+N_p-2|k)) \end{bmatrix}. \tag{2.7}$$

By doing so, it is assured that Y_k can be computed solely based on available data and the to be optimised input sequence ΔU_k. Besides, it is important to note that in (2.5) it is only possible to include the effect of the direct feedthrough explicitly up to prediction $y(i+N_p-1|k)$. As a result, for prediction $y(i+N_p|k)$ it is assumed that $u(i+N_p|k) = u(i+N_p-1|k)$.

Next, the following MPC cost function is proposed

$$J = (Y_k - Y_{k,ref})^\top \Omega (Y_k - Y_{k,ref}) + \Delta U_k^\top \Psi \Delta U_k, \tag{2.8}$$

where

$$\Omega = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix} \in \mathbb{R}^{N_p \times N_p}; \quad \Psi = \begin{bmatrix} R_\Delta & 0 & \cdots & 0 \\ 0 & R_\Delta & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_\Delta \end{bmatrix} \in \mathbb{R}^{N_p \times N_p}, \tag{2.9}$$

with Q being a positive scalar penalising any deviation from the reference and R_Δ being a positive scalar penalising the rate of change of the plant input. These values are typically chosen following performance specifications. Moreover, from (1.6),

$$Y_{k,ref} = [y_{ref}(i+1|k), \ldots, y_{ref}(i+N_p|k)]^\top \in \mathbb{R}^{N_p}. \tag{2.10}$$

By replacing ΔG_k for $\Delta\widetilde{G}_k$ and substituting (2.5) in (2.8), cost function J becomes a quadratic and convex function of ΔU_k. Hence, by setting the derivative of J with respect to ΔU_k to zero and solving for ΔU_k, the optimal input sequence that leads to the global minimum of the cost function is found to be

$$\Delta U_k^* = -W^{-1} F\big(S_x \bar{x}(i|k) + S_g \Delta\widetilde{G}_k + \boldsymbol{D}\mathcal{M} u(i-1|k) - Y_{k,ref}\big), \tag{2.11}$$

where

$$W = 2\Big(\Psi + (S_u + \boldsymbol{D}\mathcal{E})^\top \Omega (S_u + \boldsymbol{D}\mathcal{E})\Big), \tag{2.12}$$

$$F = 2(S_u + \boldsymbol{D}\mathcal{E})^\top \Omega. \tag{2.13}$$

Finally, the control input that is applied to the plant at time $t(i|k)$ is defined as

$$u(i|k) = u(i-1|k) + \begin{bmatrix} 1 & \mathbf{0}_{1\times N_p-1} \end{bmatrix} \Delta U_k^*. \tag{2.14}$$

Note that all auxiliary matrices that describe the evolution of the states are calculated only once, offline, for $N_p = N_{p,\max}$. Subsequently, online, the correctly sized matrices corresponding to the current value of N_p are extracted from the $N_p = N_{p,\max}$ matrices. By doing so, the seemingly complicated nonlinear MPC controller simply boils down to plugging in information about the states, reference and previous input into a pre-calculated control law. In this way, the algorithm is computationally extremely efficient.

2.1.1 Selecting the Sampling Times

In addition to Q and R_Δ, there are two more very important tuning parameters; namely, the MPC sample time $t_{s,in}$ and the outer-loop sample time $t_{s,out}$. First, $t_{s,out}$ is considered. In the proposed framework, the outer-loop sample time is technically the sample time of the outer-loop controller. Throughout literature there is no general consensus on what would be the best sample time of a digital controller. However, it is self-evident that the sample rate must be sufficiently higher than the desired closed-loop bandwidth. A rough guideline is that the sample time should be at least five times faster than the desired closed-loop bandwidth. Typically, higher sample rates lead to better performance and are therefore preferred. However, for the proposed framework, one must keep in mind that the inner-loop MPC runs at a sample rate which is an integer multiple greater than the outer-loop sample rate, implying that high outer-loop sample rates quickly lead to excessive computational expenses. Therefore, $t_{s,out}$ must be chosen such that it is at least 5 times faster than the desired closed-loop bandwidth, preferably more, but at the same time respecting the inner-loop computational burden. Next, a choice for $t_{s,in}$ can be made, where the relation $N_{p,\max} = t_{s,out}/t_{s,in} \in \mathbb{N}$ must be remembered. Again, there is no general consensus on what is the best choice. Typically, when $t_{s,in}$ decreases, reference tracking and disturbance rejection improves up until a certain height where it plateaus. This plateauing point is usually adopted for $t_{s,in}$, considering that the computational effort is respected.

2.1.2 Nonlinear Observer: the Unscented Kalman Filter

From (2.11), it can be seen that the proposed MPC solution needs the current (and previous) state information in order to calculate the input signal. Whereas only the output is measured, a nonlinear observer is needed to estimate the virtual states of the data-driven model. State estimation is performed of the unscented Kalman filter (UKF) [47–50], which is a nonlinear extension of the well-known linear Kalman filter. Moreover, it is a more advanced continuation of the extended Kalman filter (EKF) [50, 51], which is also a nonlinear extension of the linear Kalman filter. Both filters assume that the state can be represented by a Gaussian random value. The EKF on the one hand, propagates the Gaussian random value analytically through the local Jacobian linearisation of the nonlinear system. By doing so, the true posterior mean and covariance is approximated This is sufficient if a nonlinear system can be well approximated by this linearisation. However, the algorithm might diverge if the consecutive linearisations are not a good approximation of the nonlinear system [49], i.e., if the system is highly nonlinear. The UKF on the other hand, specifies the Gaussian state distribution using a minimal set of carefully chosen sample points, so-called sigma points. These sigma points are propagated through the nonlinear system by means of the unscented transformation [47], which is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation. In this way, an accurate approximation of the posterior mean and covariance is obtained, for *any* nonlinear system. At the same time, there is no noticeable increase of computational load

when compared to the EKF [52]. For the estimation of the discrete-time state at time $t(i|k)$, the UKF assumes the following nonlinear model with additive Gaussian white noise

$$\begin{cases} x(i|k) = \boldsymbol{f}(x(i-1|k), u(i|k), y(i|k)) + w(i|k) & w(i|k) \sim N\left(0, Q_{UKF}\right) \\ y(i|k) = \boldsymbol{h}(x(i|k), u(i|k)) + z(i|k) & z(i|k) \sim N\left(0, R_{UKF}\right), \end{cases} \tag{2.15}$$

where the functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are governed by the state and output equation of (2.1), respectively. Moreover, $w(i|k)$ is the zero-mean process noise with covariance Q_{UKF} and $z(i|k)$ is the zero-mean measurement noise with covariance R_{UKF}.

Although the mean and covariance of the probability distribution are propagated through the nonlinear function in a considerably different way, the calculation of the estimated state $\tilde{x}(i|k)$ is in accordance with the standard Kalman filter update algorithm, namely

$$\tilde{x}(i|k) = \tilde{x}^{-}(i|k) + K\big(y(i|k) - \tilde{y}^{-}(i|k)\big), \tag{2.16}$$

where $\tilde{x}^{-}(i|k)$ and $\tilde{y}^{-}(i|k)$ can be considered as the predicted *a priori* state and output estimate, respectively. Moreover, $y(i|k)$ is the measured output and K is the Kalman gain. More details on the computation of the predictions and the Kalman gain in (2.16) can be found in [47–49].

2.2 Data-Driven Modelling and Analysis

One of the crucial challenges in nonlinear data-driven modelling, or nonlinear system identification, is the compromise between capturing complex nonlinearities of the fitted model and the ability to possess a low number of parameters, or, in short, the trade-off between flexibility and parsimony [12]. For complex models, i.e. models with high flexibility and low parsimony, it is often difficult to interpret the obtained nonlinear coefficients physically. Moreover, there is a potential risk of overfit [53], and complicated models tend to become unstable when extrapolated outside their fitting region [54]. Also, research has shown that these simpler models can equal the more complicated ones in terms of flexibility [12], or even outperform them when exposed to validation data [53]. In the present thesis, two types of *a priori* system knowledge are exploited: namely, 1) where to expect the nonlinearities, i.e., in the output, and 2) how the nonlinearities are qualified, i.e., odd, even or both; derived from a nonparametric nonlinear distortion analysis and engineering judgement. This knowledge significantly reduces the amount of monomial terms in the nonlinear polynomial basis function of the model, in this way considerably improving model parsimony. The outline of this section is as follows. First, the excitation signal is introduced. Next, it is described how this excitation signal is used to perform the nonparametric nonlinear distortion analysis. Lastly, the means and steps for the identification of the full nonlinear state-space model are presented.

2.2.1 Random-Phase Multisine

One of the key components of this work is the random-phase multisine excitation signal [55], which features random behaviour in the time domain, while having the advantages of a deterministic signal in the frequency domain; it is defined as

$$u(t) = \frac{1}{\sqrt{N}} \sum_{q=-N/2+1}^{N/2-1} U_q e^{j(2\pi q f_0 t + \varphi_q)}, \tag{2.17}$$

with $\varphi_{-q} = -\varphi_q$, $U_{-q} = U_q$, $U_0 = 0$ and frequency resolution $f_{res} = f_s/N = 1/T$, where f_s is the sampling frequency and T the period of the multisine. The phases φ_q are independent and

uniformly distributed on the interval $[0, 2\pi)$, such that $E\{\varphi_q\} = 0$. Moreover, j is the imaginary unit and N is the number of time samples, while the division by $\sqrt{N}$ normalises the root-mean-square (RMS) value of the excitation signal. The amplitude U_q can be chosen to follow a desired amplitude spectrum, in this way lowering uncertainties on the measured output spectrum. Since each excited frequency line is given a random phase, the contributions at the unexcited frequency lines, i.e., the nonlinear distortions, occur in a stochastic way. However, when N approaches infinity (or is sufficiently large), the signal $u(t)$ will be normally distributed.

2.2.2 Nonlinear Distortion Analysis

A nonparametric nonlinearity analysis is performed to detect, quantify and qualify the nonlinear distortions in a system, based on the measured frequency-response function (FRF) [56]. The nonlinear distortion analysis is the main tool to validate the performance of the controller, since it allows to quantify the effectiveness of linearisation over the entire specified operating range based on a simple nonparametric test. The cornerstone of the analysis is the property of nonlinear systems to transfer power from one frequency to another, whereas a linear system cannot.

2.2.2.1 Detection, Quantification and Qualification of the Distortions

To detect the nonlinearities, the random-phase multisine (2.17) is modified such that only odd frequency lines (corresponding to odd values of q) are excited. Moreover, for every group of n_f frequency lines, one line is randomly excluded. This rejected line will serve as an odd detection line measuring odd nonlinear distortions. All the even frequency lines serve as even detection lines, measuring even nonlinear distortions. Since the linear contributions can only be present at the excited lines, any contribution at an even frequency line indicates an even nonlinearity. Similarly, any contribution at an odd detection line indicates an odd nonlinearity. Odd nonlinear distortions can also be present at the excited lines, it is however not possible to separate the distortions from the linear contribution. Figure 2.2 visualises the analysis; in this case, groups of 3 odd frequency lines, with 1 randomly rejected line, are excited. From the detection lines in the output spectrum can be derived that there are both odd and even nonlinear distortions present; moreover, there is also a noise contribution on every frequency line.

It must be noted that the described procedure is based on systems that can be represented by Volterra series [57, 58], i.e., nonlinear systems where the output depends on *all* previous inputs.

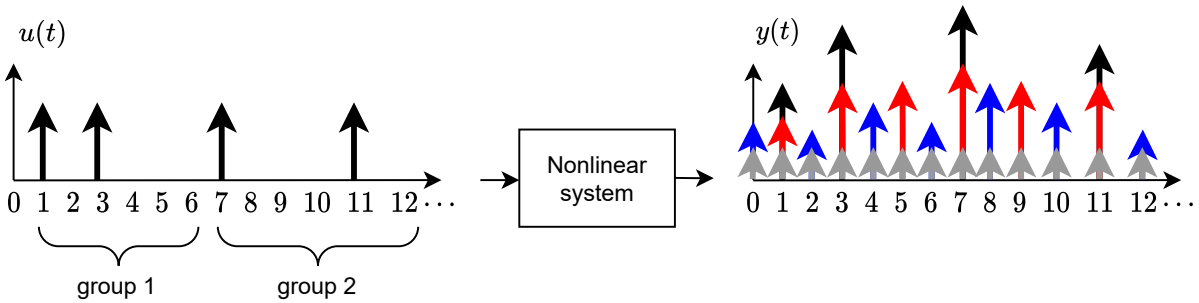


Figure 2.2: Conceptual visualisation of the nonlinear distortion analysis. Left: input spectrum of the multisine. Right: the output spectrum with the quantification and qualification of the odd (in red) and even (in blue) nonlinear distortions. The grey lines represent the contributions of the disturbing noise.

In [59] it is shown that the Volterra series can, under some reasonable conditions, arbitrarily well approximate fading-memory systems [60, 61]. However, the Lur'e form that is considered in this work contains a nonlinear feedback loop. Yet, these systems behave as fading-memory systems for sufficiently small input amplitudes [56], meaning that the nonlinear distortion analysis can still be applied.

2.2.2.2 Discriminating Noise from Nonlinear Distortions

In addition to the linear and nonlinear contributions, the FRF will also contain a disturbing noise contribution at every measured frequency line. For the nonlinear analysis, it is important to discriminate between these different contributions. The procedure to do so is described as follows: consider an integer number of $P > 1$ steady-state measurements of the exact same realisation of the multisine (2.17). Because the multisine realisations are exactly the same, the nonlinear distortions will also be the same for every realisation. However, the disturbing noise will vary. An easy way to separate the distortions from the noise is by calculating the discrete Fourier transform (DFT) of the time-domain output data. The DFT output spectra for the different realisations as a function of the frequency lines q is defined as $Y^{[p]}(q)$, where $p \in \{1, \ldots, P\}$. Subsequently, the sample mean of the output spectra is calculated according to

$$\hat{Y}(q) = \frac{1}{P} \sum_{p=1}^{P} Y^{[p]}(q). \tag{2.18}$$

In this study, the excitation signal is assumed to be noiseless, i.e., error-free and independent of the output noise. Therefore, the standard deviation on the mean value of the FRF is a good estimate for the disturbing noise contributions; it is calculated as

$$\hat{\sigma}_Y(q) = \sqrt{\frac{1}{P(P-1)} \sum_{p=1}^{P} |Y^{[p]}(q) - \hat{Y}(q)|^2}. \tag{2.19}$$

In order to obtain more reliable measurements for both the nonlinear distortions and the disturbing output noise, the measurements can be repeated over R different realisations of P steady-state measurements of the random-phase multisine. Note that each realisation of the multisine must excite the same frequency lines, but their random-phase realisations should be different. By doing so, the measurements can be averaged over the different realisations, which results in a reduction of the standard deviation of the FRF, due to the nonlinear distortions and the disturbing noise, by a factor $\sqrt{R}$.

2.2.2.3 Experimental Illustration on an F-16 Fighter Jet

The experimental illustration in this section serves as an example of the nonlinear distortion analysis, and, more importantly, it demonstrates how to interpret the main research objective of this thesis in the frequency domain. The experimental data [62] is acquired on a full-scale F-16 fighter jet, where a shaker was placed underneath the right wing to apply input signals. Two dummy payloads were mounted on the same wing to simulate the mass and inertia properties of real devices typically equipping an F-16 in flight. The measured output in this case was the acceleration signal at one of the payloads, close to the wing tip. The random-phase multisine signal was designed such that it only excited the odd frequency lines, and for every group of 4 successive odd frequency lines, 1 line was randomly rejected. Frequencies from 1 to 60 Hz where excited, and 9 realisations of 2 steady-state periods were measured. The sampling frequency

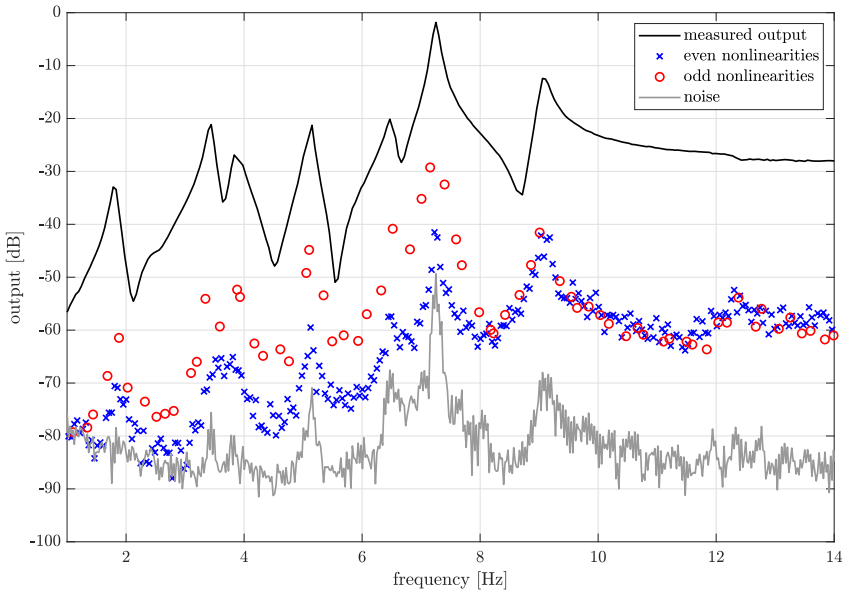


Figure 2.3: Nonlinear distortion analysis obtained from an experimental test on an F-16 fighter jet. The right wing is excited with a shaker and the acceleration near a dummy payload at the tip of the same wing is the measured output. Clearly, the level of nonlinear distortions is well above the noise level. The objective of this research is to design a controller that pushes all nonlinear distortions to the noise floor, for the entire operating range (in this case, 1 to 14 Hz).

was equal to 400 Hz and 16384 points (frequency lines) per period were considered. All excited frequency lines have the same RMS amplitude of 97.1 N.

Figure 2.3 shows the nonlinear distortion analysis obtained from the time-domain data of the measured acceleration, for the frequency band between 1 and 14 Hz. It was expected that there would be nonlinearities in the structural dynamics originating from the mounting interfaces of the two payloads. Clearly, the analysis verifies that there is significant nonlinear behaviour; the distortions are well above the noise level and are dominated by the odd nonlinearities.

The research objective of this work is to robustly eliminate all undesired nonlinearities from the IO response while preserving the linear dynamics. It is already shown how to interpret this in time domain, i.e., by simply following the linear reference. In frequency domain it is also very intuitive. Moreover, it offers the great advantage of measuring the linearising performance of the controller over the entire specified operating range by simply analysing the results of the nonlinear distortion analysis. Ultimately, the exact same multisine input realisations will be used to excite the original and linearised system under test. Then, eliminating all undesired nonlinearities boils down to pushing the odd and even nonlinear distortions as far as possible to the noise floor. Then, maintaining the linear dynamics refers to roughly preserving the measured output at the excited frequency lines; the overall shape should stay the same but the resonances might shift, both in magnitude and in frequency.

2.2.3 Nonlinear State-Space Identification

This section concerns the three-step identification procedure of the full nonlinear model. Again, the random-phase multisine excitation signal is exploited. Moreover, the obtained insights from the distortion analysis are utilised for the selection of the nonlinear basis functions.

2.2.3.1 Nonparametric BLA

Because of the nice properties of the random-phase multisine (2.17), it is particularly suitable for system identification. Specifically, the random-phase multisine behaves as a random signal in time domain, thus having the attractive properties of Gaussian noise. Moreover, it has the advantages of a deterministic signal in the frequency domain, meaning that the amplitude spectrum does not contain dips at the excited frequencies. In order to obtain the nonparametric BLA, it is stressed that new test need to be performed; this time, full multisines without any detection lines are used. In this way, the resolution of the excited frequency band increases by more than a factor 2. Besides, the nonparametric BLA should be smooth in the sense that the influence of the disturbing noise and the nonlinear distortions is minimal. Therefore, the number of periods and realisations must be sufficiently high. The nonparametric BLA $\hat{G}_{BLA}(j\omega_q) \in \mathbb{C}$ at frequency ω_q is found as

$$\hat{G}_{BLA}(j\omega_q) = \frac{1}{R}\sum_{r=1}^{R} \hat{G}^{[r]}(q), \tag{2.20}$$

with

$$\hat{G}^{[r]}(j\omega_q) = \frac{\hat{Y}^{[r]}(q)}{\hat{U}^{[r]}(q)}, \tag{2.21}$$

where $\hat{Y}^{[r]}(q)$ is the sample mean of the different periods in the output spectra according to (2.18), for the rth realisation. Similarly, $\hat{U}^{[r]}(q)$ is the sample mean of the different periods in the DFT input spectra, for the rth realisation. Note that for a particular realisation $\hat{U}(q) = U^{[p]}(q)$ for $p \in \{1, \ldots, P\}$, since the input signal is assumed to be noiseless.

2.2.3.2 Frequency-Domain Subspace Parameterisation of the BLA

Next, the nonparametric BLA is transformed into a parametric discrete-time model. To achieve this, the frequency-domain subspace method of [63] is applied with the frequency weighting in [64]. According to linear state-space matrices in (1.3), the vector of model parameters is introduced as

$$\theta_L = [\text{vec}(\boldsymbol{A});\ \text{vec}(\boldsymbol{B});\ \text{vec}(\boldsymbol{C});\ \boldsymbol{D}], \tag{2.22}$$

where vec denotes the operation of stacking the columns of matrices on top of each other. The parametric estimates for the state-space matrices are obtained by minimising the following weighted least-squares cost function

$$V_L(\theta_L) = \frac{1}{N}\sum_{q=1}^{N} \frac{\epsilon_L^H(q,\theta_L)\epsilon_L(q,\theta_L)}{\hat{\sigma}^2_{BLA}(q)}, \tag{2.23}$$

where H denotes the Hermitian transpose and $\hat{\sigma}^2_{BLA}(q)$ is the weighting function, which is defined later. Moreover,

$$\epsilon_L(q,\theta_L) = G_L(j\omega_q,\theta_L) - \hat{G}_{BLA}(j\omega_q), \tag{2.24}$$

with

$$G_L(j\omega_q,\theta_L) = \boldsymbol{C}(z_q\boldsymbol{I}_n - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}, \tag{2.25}$$

where $z_q = e^{j(2\pi q/N)}$ is the z-transform variable and $\boldsymbol{I}_n$ the $n \times n$ identity matrix. In [63], cost function (2.23) is weighted by the true noise covariance matrix, which is in practice never available. Yet, in [64] is shown that the stochastic properties of the obtained model parameters

are almost not affected if the true noise covariance matrix is replaced by the sample noise covariance matrix, obtained from the nonparametric BLA as follows

$$\hat{\sigma}^2_{BLA}(q) = \frac{1}{R(R-1)} \sum_{r=1}^{R} |\hat{G}^{[r]}(j\omega_q) - \hat{G}_{BLA}(j\omega_q)|^2. \tag{2.26}$$

Generally speaking, the frequency-domain subspace method generates reasonable model estimates. Nevertheless, $V_L(\theta_L)$ can be further minimised with respect to all parameters in θ_L by means of the Levenberg-Marquardt (LM) optimisation algorithm [65, 66], which combines the large convergence region of the gradient descent method with the fast convergence of the Gauss–Newton method. This algorithm requires the calculation of the Jacobian matrix of $\epsilon_L(q, \theta_L)$ with respect to the model parameters. The technicalities regarding these computations are elaborated upon in [67]. Besides, $V_L(\theta_L)$ is also heavily dependent on the model order n and the dimension parameter $\beta > n$ chosen in the identification procedure [67]. In practice, different combinations of these parameters are used to generate a number of different models, after which the one that corresponds to the lowest cost function is preserved.

2.2.3.3 Nonlinear Optimisation

The last step of the identification procedure is to estimate the full nonlinear model (1.3). First, the vector of model parameters is defined as

$$\theta_{NL} = [\text{vec}(\boldsymbol{A});\ \text{vec}(\boldsymbol{B});\ \text{vec}(\boldsymbol{C});\ \boldsymbol{D};\ \text{vec}(\boldsymbol{E})], \tag{2.27}$$

where the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are initialised by the estimates of the parametric BLA, while the entries of $\boldsymbol{E}$ are set to zero. By doing so, it is guaranteed that the nonlinear model performs as least as good as the linear one [67]. Again, a weighted least-squares cost function is introduced as

$$V_{NL}(\theta_{NL}) = \frac{1}{N} \sum_{q=1}^{N} \frac{\epsilon^H_{NL}(q, \theta_{NL}) \epsilon_{NL}(q, \theta_{NL})}{\hat{\sigma}^2_{BLA}(q)}, \tag{2.28}$$

with

$$\epsilon_{NL}(q, \theta_{NL}) = Y_m(q, \theta_{NL}) - \hat{Y}(q), \tag{2.29}$$

where $Y_m(q, \theta_{NL})$ and $\hat{Y}(q)$ are the modelled and measured output DFT spectra, respectively. Both spectra are acquired based on one particular realisation of the FRF, e.g., $\hat{G}^{[1]}(j\omega_q)$ in (2.21). The measured output spectra then clearly is $\hat{Y}^{[1]}(q)$, while the modelled time-domain output is obtained by simulating the full nonlinear state-space (1.3) for the time-domain equivalent of $\hat{U}^{[1]}(q)$. During this simulation, $\zeta(y(k))$ must contain the s monomial terms of the nonlinear polynomial basis function. Although all kinds of basis functions could be used, only polynomial basis functions are considered. The reason for this is that polynomials are simple to compute, linear-in-the-parameters, easy to extend to the multivariate case, and possess universal approximation properties [68]. Typically, the nonlinear basis function contains plenty of cross products between the input, output and states, up to a certain polynomial degree. In this work, the Lur'e form restricts that the nonlinear polynomial is a function of the output (and possibly its derivatives) only. Besides, information from the nonlinear distortion analysis provides insight into the type of nonlinearities (even or odd). This knowledge considerably simplifies the identification procedure. Nevertheless, multiple realisations of basis functions are addressed, in order to obtain a model with the lowest fitting error.

Chapter 3

Simulation Results on a Nonlinear Oscillator

In this chapter, simulations are performed on a nonlinear oscillator with a quadratic and cubic spring nonlinearity. The system is an extension of the Duffing oscillator [69], which has become a famous system for studying nonlinear dynamics. The plant dynamics are governed by

$$m\ddot{y}(t) + c_l\dot{y}(t) + k_l y(t) + k_q y^2(t) + k_c y^3(t) = u(t), \tag{3.1}$$

where m is the mass, c_l is the linear damping coefficient and k_l, k_q and k_c are the linear, quadratic and cubic stiffness coefficients, respectively. The output $y(t)$ is the displacement of the mass m, while the input $u(t)$ is a force exciting the mass. The system parameters are summarised in Table 3.1. With these parameters, the linear natural frequency and damping ratio are equal to 3.56 Hz and 2.24 %, respectively. The cubic nonlinearity in (3.1) models the dominant hardening effect of the spring. A visualisation of the hardening effect is shown in Figure 3.1, which shows the FRFs of (3.1), when excited with full random-phase multisines for increasing levels of the RMS amplitude. For the lowest RMS amplitude, the system acts as a linear system, i.e., almost exactly the *underlying* linear system with the given natural frequency and damping ratio. For the increasing amplitudes, a shift in the resonance frequency is observed, which is a direct result of the hardening effect of the nonlinear spring. The seemingly noisy behaviour is due to the increase of nonlinear distortions on the FRF. Furthermore, the mentioned extension of the Duffing oscillator refers to the inclusion of a significant quadratic nonlinearity. From a physical point of view, this nonlinearity can be seen as an asymmetry in the nonlinear spring.

The simulations of this chapter are all performed in MATLAB/Simulink, where the plant (3.1) is implemented as depicted in Figure 3.2. The simulations are presented for an increasing level of difficulty. First, a proof of concept is shown for an ideal situation, i.e., there are no modelling errors, no disturbances and all the states can be measured perfectly. This environment is also used to show the correct working of the MPC in case there would have been a significant direct

	m	c_l	k_l	k_q	k_c
value	1	1	$5 \cdot 10^2$	$5 \cdot 10^4$	$1 \cdot 10^8$
unit	kg	Ns/m	N/m	N/m^2	N/m^3

Table 3.1: Physical parameters of the nonlinear plant. With these values, the natural frequency and the damping ratio of the underlying linear system are 3.56 Hz and 2.24 %, respectively.

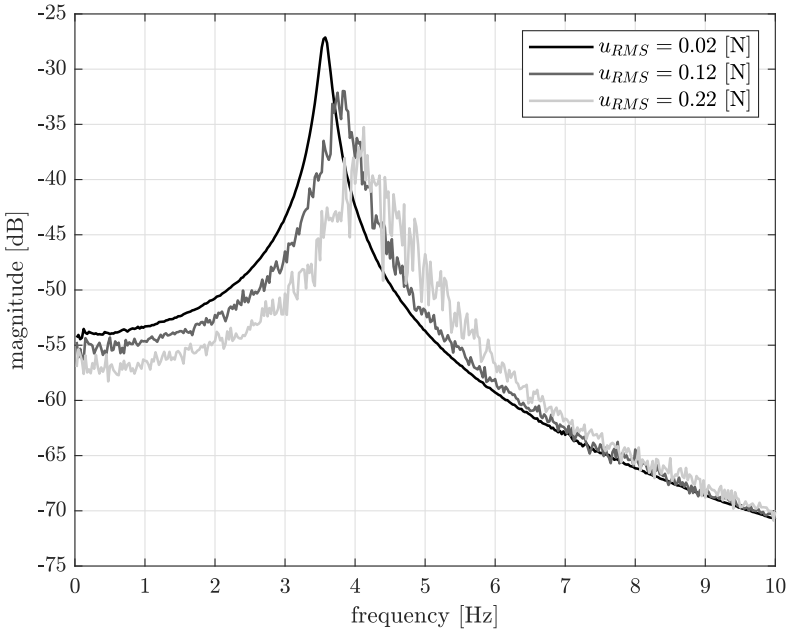


Figure 3.1: FRFs of system (3.1) for increasing levels of the RMS amplitude of full random-phase multisines. The FRFs show the dominant hardening effect of the nonlinear spring, which induces a shift in the natural frequency for increasing levels of excitation. The seemingly noisy behaviour is due to the increase of nonlinear distortions on the FRF.

feedthrough term. Next, the more realistic scenario is created where the model is obtained according to the procedure in Chapter 2.2. Lastly, it is investigated how robust the control structure is in case of modelling errors and imperfect state information.

3.1 Proof of Concept

As mentioned, the simulation results in this section are all obtained under ideal conditions, i.e., without any modelling errors, disturbances, and under the assumption that all states can be perfectly measured. First, system (3.1) is cast into the continuous-time equivalent of (1.3) as follows

$$\begin{cases} \dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -k_l/m & -c_l/m \end{bmatrix}}_{\boldsymbol{A}_c} x(t) + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{\boldsymbol{B}_c} u(t) + \underbrace{\begin{bmatrix} 0 & 0 \\ -k_q/m & -k_c/m \end{bmatrix}}_{\boldsymbol{E}_c} \zeta(y(t)) \\ y(t) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\boldsymbol{C}_c} x(t), \end{cases} \tag{3.2}$$

where $x(t) = [y(t) \;\; \dot{y}(t)]^\top$ and $\zeta(y(t)) = \left[y^2(t) \;\; y^3(t)\right]^\top$, with $\top$ denoting the transpose operation. Moreover, $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$ and $\boldsymbol{E}_c$ are the continuous-time representations of $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{E}$, respectively. In order to obey the form of (1.3), (3.2) is discretised assuming zero-order-hold (ZOH) dynamics. Subsequently, the discretised model is augmented and the linearising MPC is implemented as described in Chapter 2.

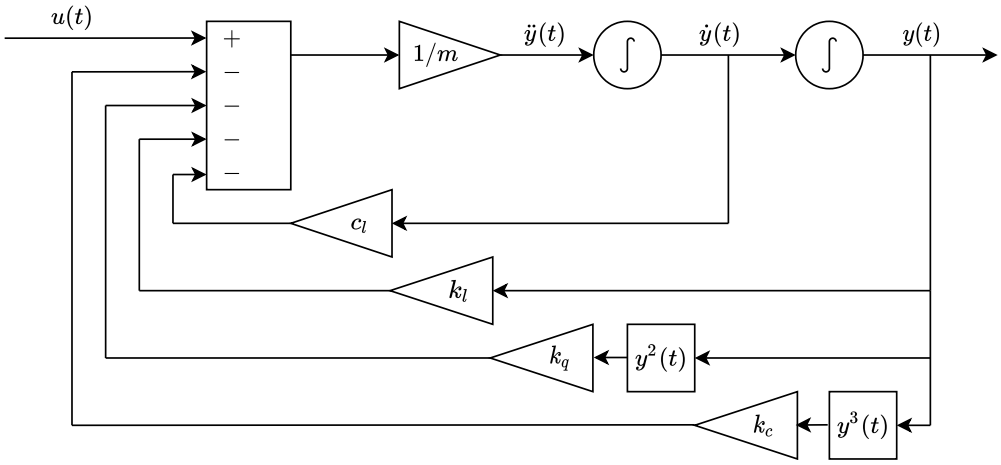
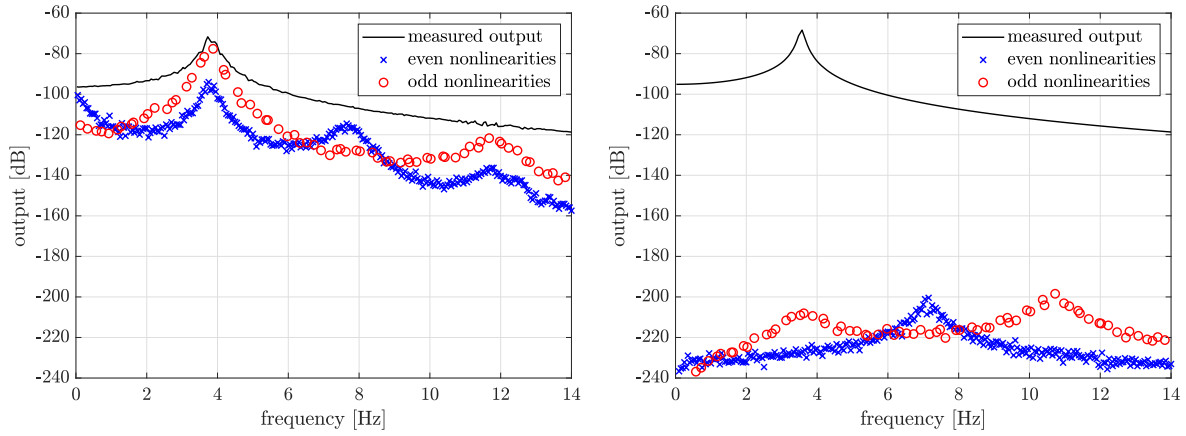


Figure 3.2: Block diagram of the nonlinear plant implementation in Simulink.

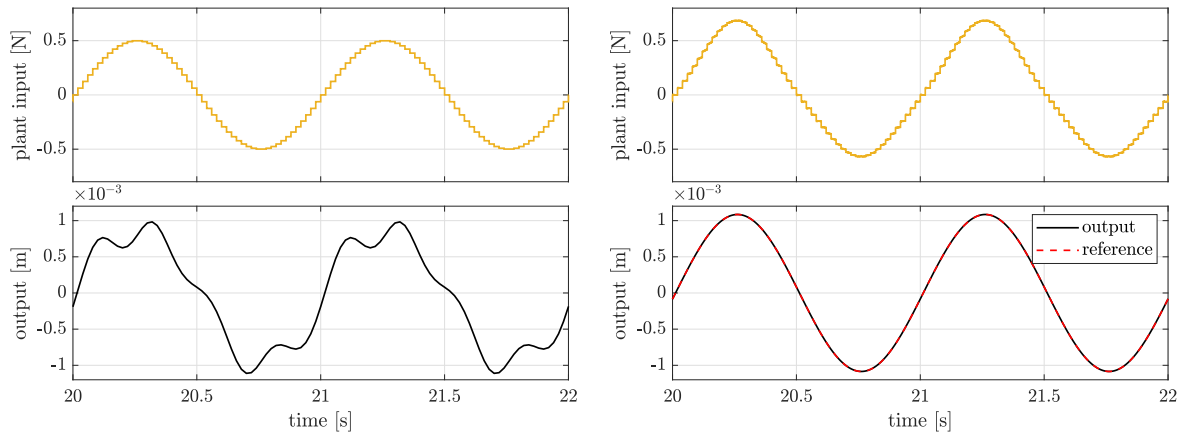
3.1.1 Performance under Perfect Conditions

As stated in Chapter 2, the first important tuning parameter to choose is the outer-loop sample time $t_{s,out}$. From the information in Table 3.1 is derived that the linear natural frequency is equal to 3.56 Hz. Based on this information, the desired closed-loop bandwidth for an outer-loop controller is chosen as 14 Hz, well beyond the linear resonance. In other words, the frequency band of interest for the linearising MPC ranges between 0 and 14 Hz; the outer-loop controller will not be designed in this study. Remember that the outer-loop sample time should be at least 5 times faster than the desired closed-loop bandwidth, it is therefore chosen as $t_{s,out} = 10$ ms, or equivalently, 100 Hz. The MPC sample time is set to $t_{s,in} = 1$ ms, or equivalently, 1000 Hz. Moreover, the values for Q and R_Δ are chosen as $1 \cdot 10^{12}$ and 1, respectively. Figure 3.3 shows the nonlinear distortion analyses of system (3.1) before and after linearisation. The system is excited with 10 realisations of 5 periods of an odd random-phase multisine with an RMS amplitude of 0.12 Hz. For every group of 4 odd frequency lines, 1 line is randomly rejected to serve as an odd detection line. Moreover, 4000 points per period are considered, so $f_{res} = 0.025$



(a) (b)

Figure 3.3: Nonlinear distortion analysis on system (3.1) before (a) and after (b) linearisation. The distortions are suppressed with approximately 100 dB over the entire frequency range of interest, so excellent linearisation is achieved.



(a) (b)

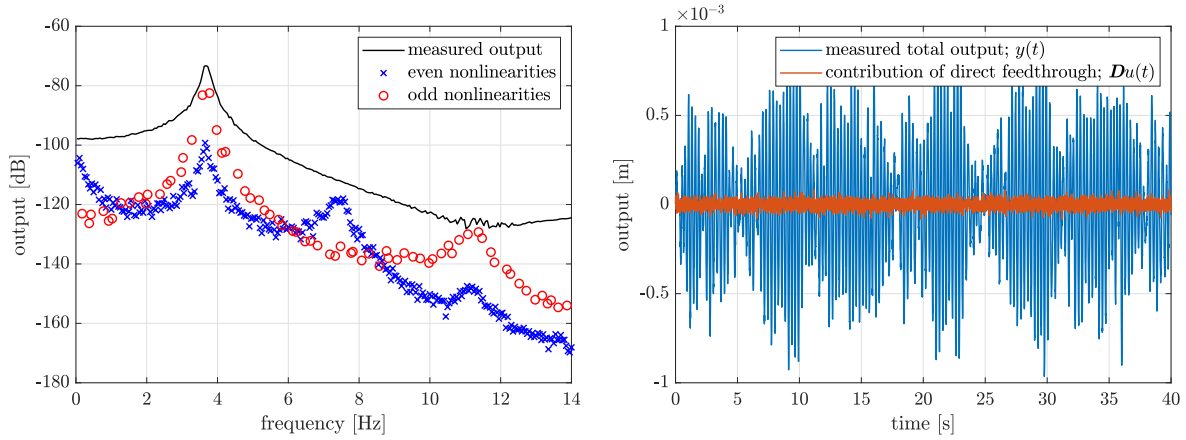
Figure 3.4: Comparison between the original nonlinear (a) and linearised (b) system when excited with an outer-loop sine excitation signal. From (a) can be seen that the sine input renders a nonlinear output displacement, while (b) shows how this sine input is modified by the MPC in order to obtain perfect tracking of the linear output reference.

Hz. The nonlinear distortion analysis would take more than 30 minutes on an experimental setup; it took less than 1 minute to complete the simulations that used the linearising MPC. Therefore, it can be concluded that the computational burden of the MPC is rather low. From Figure 3.3a, it can be seen that the nonlinear distortions in the system are significant; especially around the resonance frequency, where the odd distortions are as large as the linear contributions. Figure 3.3b shows the excellent performance of the linearising MPC; all nonlinear distortions are suppressed by approximately 100 dB. As a result, the measured output is more smooth and the resonance frequency is slightly shifted due to the elimination of the hardening effect of the nonlinear spring. It is safe to say that the level of nonlinear distortions is pushed well below any potential noise level. Therefore, it can be concluded that the IO response of the nonlinear plant is successfully linearised.

Figure 3.4 shows the response of the system to an outer-loop sine excitation with an amplitude of 0.5 N and a frequency of 2 Hz. The left column shows the input and output of the original uncontrolled system; in this case, the plant input equals the outer-loop sine input. Clearly, this sine excitation renders a nonlinear output. On the other hand, the right column shows the plant input and output signals of the controlled system. It can be seen that the linearising MPC appropriately modifies the outer-loop sine signal, such that the linear output reference is perfectly tracked, i.e., with an error of $2.16 \cdot 10^{-4}\%$. Here, the error percentage is the ratio between the standard deviation of the error signal and the standard deviation of the output. The effect of both the quadratic and cubic nonlinearity are visible in the controlled plant input; namely, a higher amplitude than 0.5 N is needed in both directions of displacement in order to compensate for the hardening effect of the spring, i.e., the cubic nonlinearity. At the same time, more force is needed in the positive direction to compensate for the asymmetry in the spring, i.e., the quadratic nonlinearity.

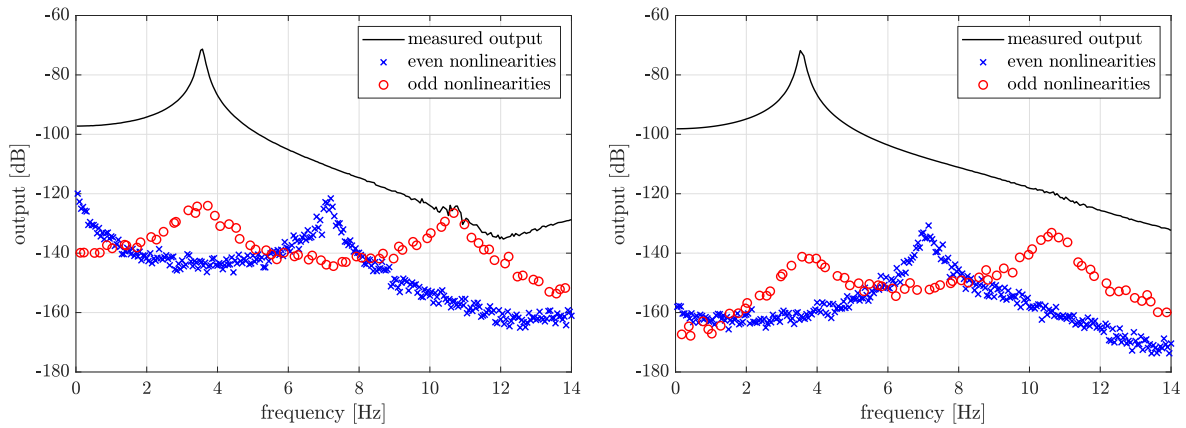
3.1.2 Studying the Effect of Direct Feedthrough

Whereas most mechanical systems with displacement as output do not contain a direct feedthrough term, the identification procedure still assigns a small value to the $\boldsymbol{D}$ matrix. It is chosen to



(a) (b)

Figure 3.5: Nonlinear distortion analysis on the uncontrolled plant with significant direct feedthrough (a) and time-domain plot of one period of the measured output (b) which shows the contribution of the direct feedthrough on the total output.



(a) (b)

Figure 3.6: Nonlinear distortion analyses after linearisation, when the direct feedthrough is disregarded in the calculation of the MPC input signal (a) and when it is incorporated (b). Significant performance reduction is achieved when compared to the plant without direct feedthrough. However, decent linearisation is achieved up to and including 5 Hz even for case (a), implying that the MPC is robust to significant modelling errors.

incorporate this matrix in the calculation of the MPC input signal intending to increase performance and also to make the MPC solution applicable to systems that *do* contain a direct feedthrough term. In this section it is studied how performance is affected when a significant direct feedthrough term is added to the plant (3.2); it is implemented as $\boldsymbol{D} = 2 \cdot 10^{-4}$. The exact same nonlinear distortion analysis as in the previous Section 3.1.1 is performed to the system with direct feedthrough. Figure 3.5 shows the nonlinear distortion analysis before linearisation on the left and the time-domain plot of one period of the measured output on the right. From the latter, it can be seen how the direct feedthrough contribution compares to the total output. The nonlinear distortion analysis shows many similarities with the nonlinear system without direct feedthrough in Figure 3.3a, especially up to and including 5 Hz. After this resonance, the

effect of the direct feedthrough is more present, which makes sense since the system dynamics quickly decrease in magnitude whereas the direct feedthrough term does not.

Figure 3.6 shows the nonlinear distortion analyses after linearisation for two cases; namely, (a) when the direct feedthrough is disregarded in the calculation of the MPC input signal, and (b) when it is incorporated. It must be noted that the MPC sample rate was increased to 2000 Hz in order to achieve the results; all the other tuning parameters remained the same. From Figure 3.6, three things catch attention when compared to the linearised nonlinear distortion analysis for the system without direct feedthrough in Figure 3.3b: 1) significantly less linearising performance is achieved; 2) still, an approximately 40 dB of reductions of nonlinear distortions is achieved up to and including 5 Hz even when the direct feedthrough term is disregarded by MPC; and 3) as expected, including the direct feedthrough in MPC calculations improves the performance over the entire range of interest significantly. From the figures can be concluded that the main problems arise at the higher frequencies, where the effect of the direct feedthrough is dominant over the system dynamics. Note that from (1.5), the MPC always tries to eliminate the undesired effect of direct feedthrough by not including it in the linear reference. However, to overcome high-frequency performance decrease, it might be sensible to include the direct feedthrough in the linear reference in case the system contains a significant natural direct feedthrough term. Besides, one could also argue that the frequency range of interest here is chosen as too large, since the difference between the magnitude at 0 and 14 Hz is approximately 40 dB. The range between 0 and 5 Hz might have been more suitable; and in that case, the high-frequency performance problem does not apply. Also, when the direct feedthrough term is marginal, it can also be chosen to not include it in the MPC calculations; the integral action property should manage this small modelling error, and potential high-frequency performance decrease is prevented.

3.2 Realistic Simulation Environment

In this section, the simulation example is extended to a form that more resembles a real life experiment. Specifically, white Gaussian noise is synthetically added to the output to simulate sensor noise. The mean standard deviation of the output measurements of the ideal simulations is divided by a factor 100 to obtain the disturbing noise standard deviation. By doing so, a signal-to-noise ratio (SNR) of approximately 40 dB is achieved, which is a typical ratio for mechanical systems. The nonlinear distortion analysis after adding the noise is shown in Figure 3.7, the random-phase multisines have the same specifications as in the Section 3.1.1. Moreover, the model is identified based on IO data only. So, in contrast to Section 3.1.1, there will be mismatches between the plant (Figure 3.2) and the model. This also means that the UKF is needed to measure the virtual states of the plant. Here, the word virtual is used since the states of the model do not have a physical meaning. To make meaningful comparisons with the ideal simulations, the MPC tuning parameters are not altered.

3.2.1 System Identification

The identification data for the system is obtained by performing 20 realisations of 5 periods of a full random-phase multisine with an RMS amplitude of 0.12 N. The number of data points per period is 40 000 at a sampling frequency of 1000 Hz. To ensure steady-state conditions, the first period of every realisation is rejected. Moreover, 1 realisation is saved for validation. Based on the measured input and output data, both the nonparametric (2.20) and the second order parametric (2.23) BLA are obtained as described in Chapter 2.2. Figure 3.8 shows the fitting results of the BLAs in frequency domain; it can be seen that the parametric BLA matches the nonparametric one very well. Specifically, the residual of the estimate, i.e., $G_L - \hat{G}_{BLA}$, has the

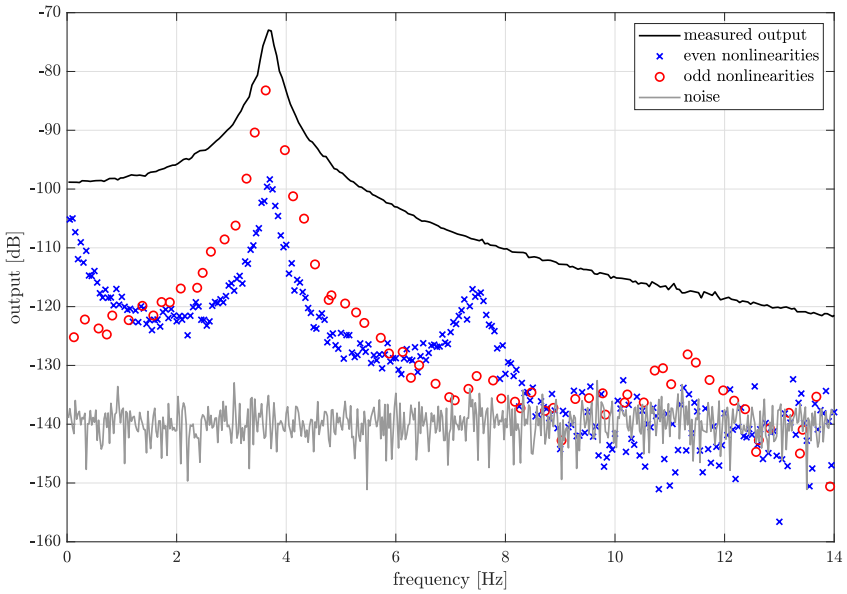


Figure 3.7: Nonlinear distortion analysis of the system with output noise before linearisation.

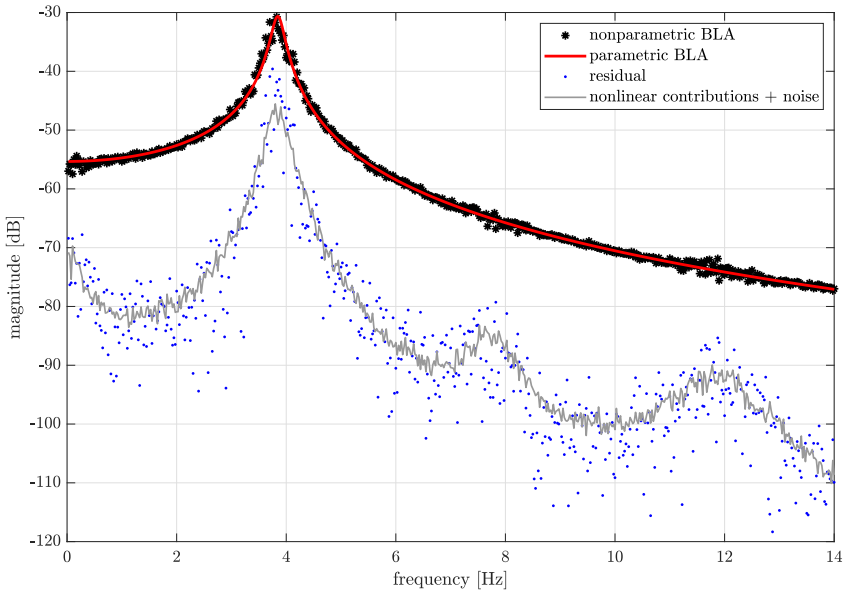
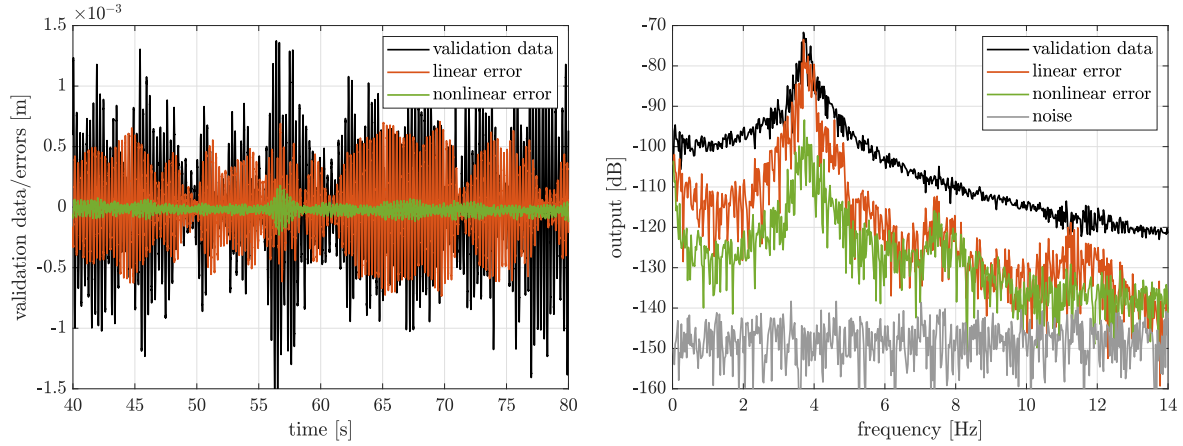


Figure 3.8: Fitting results of the parametric and nonparametric BLA in the frequency domain. From the insightful figure can be concluded that the parametric BLA is estimated in the best way possible, since the model residual is equal to the sum of the nonlinear distortions and the disturbing noise.

same magnitude as the total standard deviation on the nonparametric BLA, i.e., the sum of the nonlinear distortions and the noise. Obviously, a linear model is never capable of capturing nonlinear distortions, so it can be concluded that the frequency-domain subspace method is exploited to the best of its abilities.

Next, the full nonlinear model is estimated by minimising the weighted least squares cost function



(a) (b)

Figure 3.9: One period of time-domain validation data and the corresponding identification errors of the parametric BLA and the full nonlinear model (a). Here, the identification error is reduced from 59.2% to 6.64%. The same data is also represented in the frequency domain (b).

(2.28). Figure 3.9 shows the results of the nonlinear optimisation, when the full nonlinear model is exposed to the validation realisation of the multisine. The left figure shows one period of the time-domain validation data, as well as the corresponding linear and nonlinear identification error signals. Clearly, the nonlinear error is significantly smaller than the linear one. Specifically, the linear and nonlinear errors are 59.2% and 6.64%, respectively. These percentages are obtained by dividing the standard deviation of the errors by the standard deviation of the validation output data. The right figure shows the same validation data and error signals in the frequency domain. Ideally, the nonlinear error would approach the noise floor, which is obviously not the case in the figure. Still, the nonlinear error is consistently (more than) 20 dB below the validation data, which is in rough accordance with the time-domain findings. Only around 7 Hz, the peak of even nonlinear distortion (see Figure 3.7) is poorly captured by the nonlinear model; there is no noticeable improvement when compared to the linear error. The parameter values of the obtained nonlinear state-space model are provided in Table B.1 of Appendix B.

3.2.2 Tuning the UKF Covariance Matrices

In addition to the inner and outer-loop sample times, the MPC also has the tuning parameters Q and R_Δ. By appropriately tuning these parameters, a balance between performance and aggressiveness of the MPC is determined. Previously, the optimal trade-off for the system in the ideal simulation environment was obtained; these values will also be used for the following, more realistic simulations. As for these simulations, the UKF covariance matrices Q_{UKF} and R_{UKF} are important tuning parameters that affect the performance of the entire control structure. In this section, the effect of these matrices on performance and aggressiveness is studied.

Figure 3.10 shows the measured output when the *uncontrolled* plant is excited with a sine input signal of 1 Hz and an amplitude of 0.5 N. In addition, the UKF filtered output is shown for different diagonal values of the covariance matrix of the process noise Q_{UKF}. For all simulations, the measurement noise covariance is set to the mean value of the output noise covariance of the nonlinear distortion analysis, i.e., $R_{UKF} = 1.13 \cdot 10^{-14}$ m^2. From the figure, it can be seen that the higher the diagonal values in Q_{UKF}, the better the measured output is approximated. Of course, this is inherently the result of the rationale behind these tuning matrices; namely,

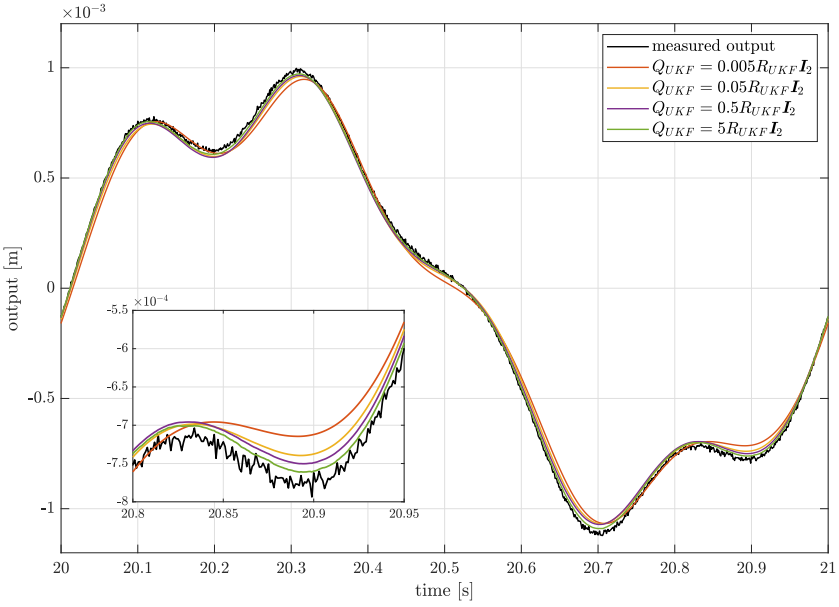
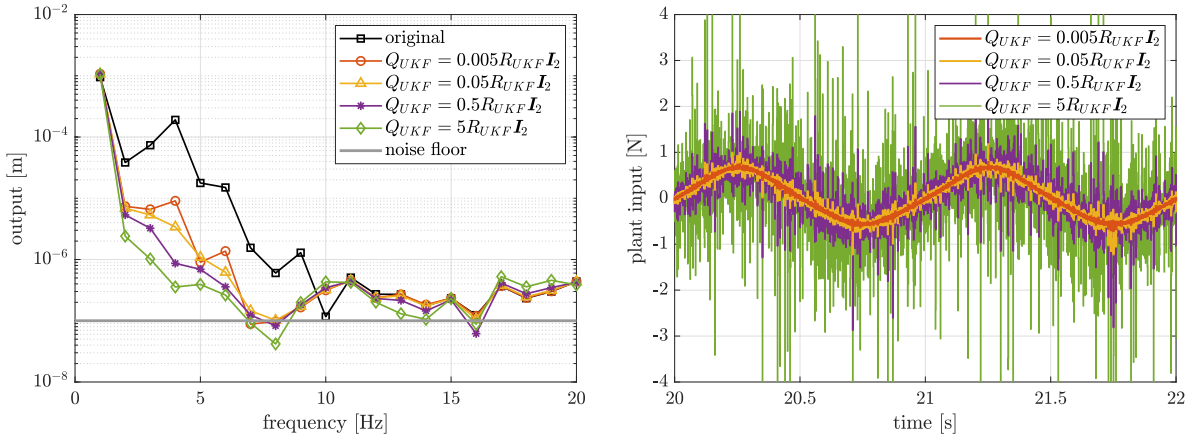


Figure 3.10: UKF performance for different realisations of Q_{UKF}, for the uncontrolled plant. Higher diagonal values of Q_{UKF} lead to better approximations of the measured output.

relatively high diagonal values in Q_{UKF} imply that the measurements can be considered reliable (i.e., not too noisy), whereas a relatively high R_{UKF} implies that the UKF should mostly rely on the model (since the measurements are noisy). Next, it is investigated how these findings transfer to the *controlled* situation. Figure 3.11 shows the response of controlled plant when the sine input is applied as the outer-loop input. In the left figure, the output spectrum for the same realisations of Q_{UKF} is shown. The figure should be read as follows. In black, the amplitude spectrum of the uncontrolled output of Figure 3.10 is shown. The excitation frequency of the sine input was 1 Hz, yet there are also significant output contributions at higher multiples of the excitation frequency, which implies that there are nonlinearities in the system. Therefore, the objective of the MPC in terms of this output amplitude spectrum is to remove the contributions at all frequencies higher than 1 Hz. It can be seen that for every different realisation of Q_{UKF}, the MPC reduces the nonlinearities by at least a factor 10 until the approximate noise floor is reached. As expected, linearising performance increases as the diagonal values of Q_{UKF} increase. However, the aggressiveness of the required input signal also increases as the diagonal values of Q_{UKF} increase. This can be observed from Figure 3.11b, which shows the required plant inputs for 2 sine periods. For the lowest diagonal values of Q_{UKF}, the plant input is very similar to the input in Figure 3.4, i.e., the noise and error-free simulations. For the highest diagonal values of Q_{UKF}, there are peaks which are more than 10 times higher in magnitude than the original amplitude (0.5 N) of the outer-loop excitation signal. For the next simulations, $Q_{UKF} = 0.05R_{UKF}\boldsymbol{I}_2$ is selected since it provides a good trade-off between linearising performance and control aggressiveness. Figure 3.12 shows one time-domain period of the measured output, filtered output and MPC reference corresponding to the right realisation of Q_{UKF} in Figure 3.11. Here, the filtered output is used for the MPC calculations. It can be seen that this filtered output almost perfectly tracks its reference signal. However, the filtered output does not always perfectly represent the clean version of the measured output, especially at the sine peak values. This explains why there are still nonlinear contributions at the higher frequencies in Figure 3.11a.



(a) (b)

Figure 3.11: Output amplitude spectra (a) for different realisations of Q_{UKF}, when the system is excited by an outer-loop sine excitation signal, and the required plant input signals (b). It can be observed that better linearising performance is achieved for higher diagonal values of Q_{UKF}, at the cost of more aggressive input signals.

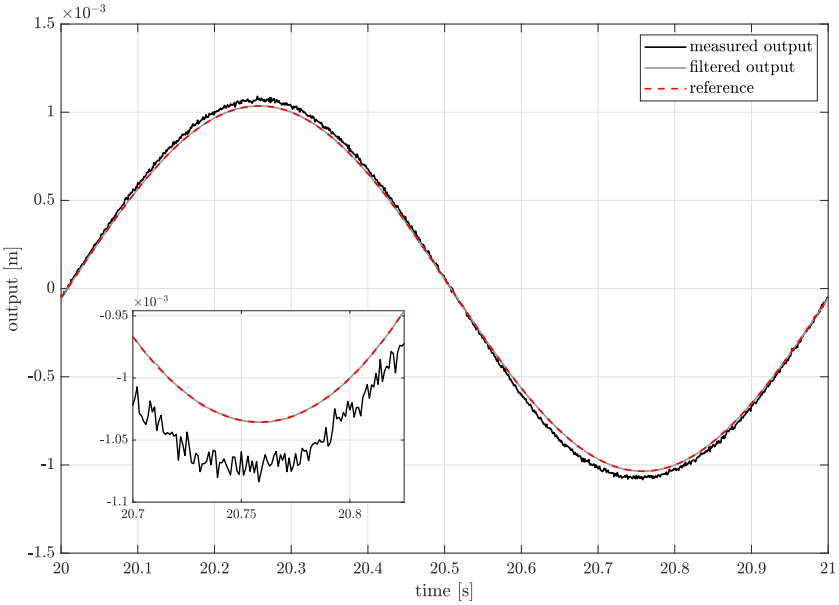


Figure 3.12: One time-domain period of the measured output, filtered output and MPC reference, for $Q_{UKF} = 0.05R_{UKF}\boldsymbol{I}_2$. It is shown that the MPC almost perfectly tracks it reference signal. However, the filtered output does not always perfectly represent the clean version of the measured output, especially at the sine peak values.

3.2.3 Performance under Realistic Conditions

Finally, the linearising performance over the entire frequency range of interest is analysed. The chosen tuning parameters are summarised in Table 3.2. Again, the random-phase multisine excitation signal has the same properties as in Section 3.1.1. Figure 3.13 shows this nonlinear distortion analysis of the linearised plant. The results are compared to Figure 3.7, which shows the nonlinear distortion analysis before linearisation. Great linearisation is achieved over the

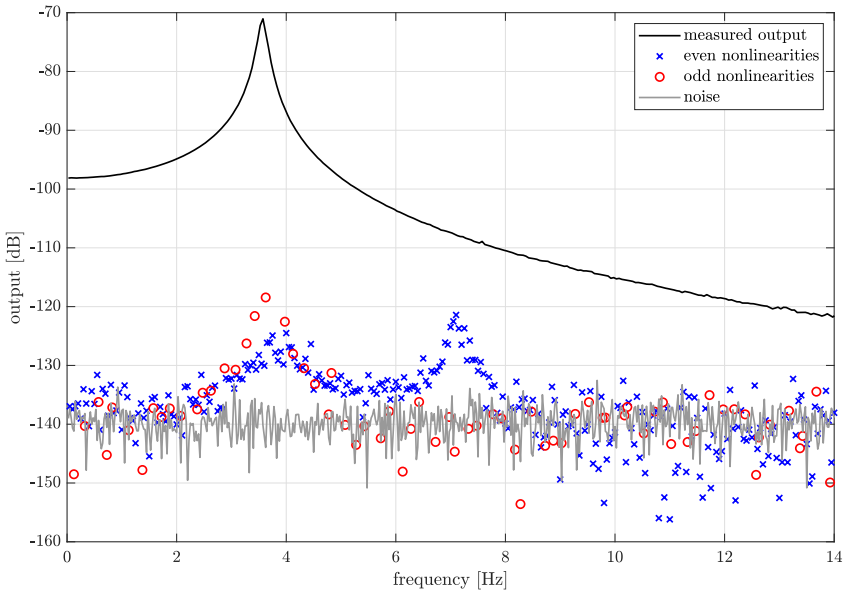


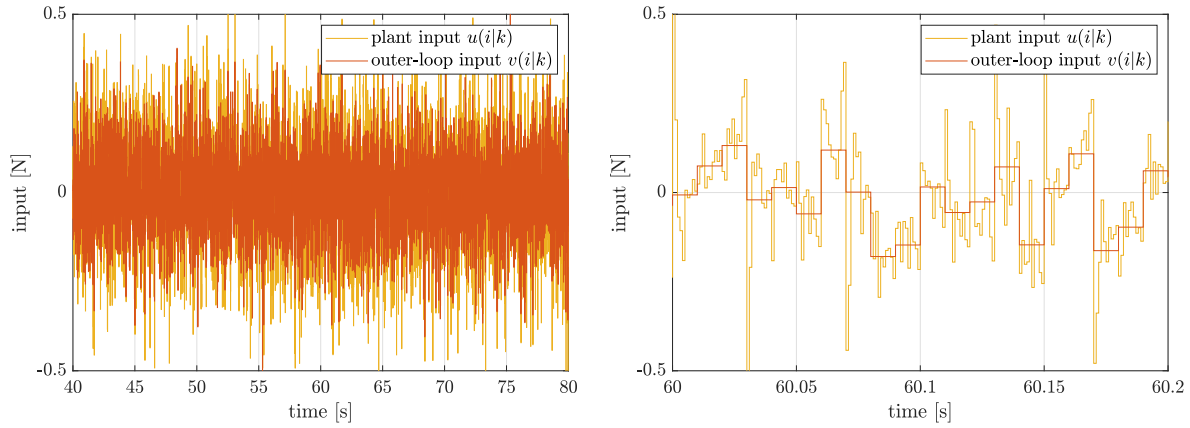
Figure 3.13: Nonlinear distortion analysis after linearisation. The figure should be compared to the findings in Figure 3.7, i.e., the nonlinear distortion analysis before linearisation. Excellent linearisation is achieved since all distortions of Figure 3.7 are pushed (close to) the noise floor.

entire frequency range; for most frequencies, the nonlinear distortions are pushed to the noise floor, which is the best possible achievability. There are some small nonlinear residuals around the resonance frequency, but they are 50 dB below the measured output. The relatively poor suppression around 7 Hz can be explained by the frequency-domain validation graph of Figure 3.9, i.e, the nonlinear model was not able to capture the nonlinearities around this frequency, so they are not adequately suppressed. Yet, it must be noted that the remaining nonlinear residuals can be pushed to the noise floor arbitrarily close by increasing the diagonal values in Q_{UKF}. However, as shown in Section 3.2.2, this would require excessive input signal. The frequency plot of Figure 3.9 also confirms that the integral action property of the MPC is working correctly; namely, the nonlinear validation error is significantly further away from the noise floor than the suppressed distortions in Figure 3.13. This implies that, even in case of modelling errors, the MPC is able to track the linear reference such that the nonlinear distortions are pushed to the noise floor.

The outer-loop input $v(i|k)$ and the plant input $u(i|k)$ for one period of the multisine are shown in Figure 3.14. From the figure can be observed that no excessive inputs were required to achieve the linearisation. Especially the zoomed-in plot is interesting since it shows the underlying MPC actions required to achieve linearisation. Seemingly, the most aggressive inputs are calculated at the start of every outer-loop sample period, i.e., when the prediction horizon is at its maximum value, i.e., $N_{p,\max} = t_{s,out}/t_{s,in} = 10$.

	$t_{s,out}$	$t_{s,in}$	Q	R_Δ	Q_{UKF}	R_{UKF}
value	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{12}$	1	$5.65 \cdot 10^{-16} \boldsymbol{I}_2$	$1.13 \cdot 10^{-14}$
unit	s	s	-	-	-	m^2

Table 3.2: Summary of the tuning parameters of the entire control structure.



(a) (b)

Figure 3.14: Time-domain inputs of one period of the random-phase multisine (a) and a close-up around $t = 60$ s (b). Both the outer-loop input and the plant input are shown. It can be seen that acceptable MPC inputs are required to perform the linearisation.

3.3 Studying Robustness

This section studies the robustness properties of the proposed control structure. Two tests are performed; in the first test, it is investigated how performance is affected in case the quadratic nonlinearity is not modelled. In the second test, the system is excited with a random-phase multisine whose RMS amplitude is significantly higher than previously considered. In this way, the extrapolation properties of the entire control structure are examined.

3.3.1 The Influence of Unmodelled Nonlinearities

Figure 3.15 shows the nonlinear distortion analysis when only the cubic nonlinearity is considered. This is achieved by setting the first column of the $\boldsymbol{E}$ matrix of the model in Table B.1 to zero. By doing so, the linear parts of both models remain exactly the same, in this way allowing for a more valid comparison between the two. As expected, the even nonlinearities are significantly more present when compared to Figure 3.13. In addition, almost no improvement regarding suppressing the even nonlinearities is achieved when compared to the original system before linearisation in Figure 3.7. Only around the resonance frequency there is a reduction of approximately 15 dB. Nevertheless, the odd nonlinearities are all reduced to a satisfactory level, similar to Figure 3.13. Regarding the even nonlinearities, one could argue that they *should* in fact be suppressed, even when the quadratic nonlinearity is not modelled. After all, the MPC is tracking the exact same linear reference since the linear parts of both models are the same. So, the integral action property of the MPC should therefore be able to handle the induced modelling errors. Figure 3.16 allows for more insight in the observations made from the distortion analysis. Here, two different error signals for both the complete and incomplete model are shown; the left figure shows the MPC error, i.e., the filtered output minus the reference output, whereas the right figure shows the UKF error, i.e., the filtered output minus the true output. Here, the true output is the noise-free output, which is accessible in simulation. It can be seen that there is no noticeable difference between the MPC error signals. Specifically, this means that the MPC is able to match the *filtered* output and the reference output equally well for both the complete and incomplete model. Therefore can be concluded that the integral action property of the MPC is indeed working correctly. The UKF error plot explains where the poor suppression

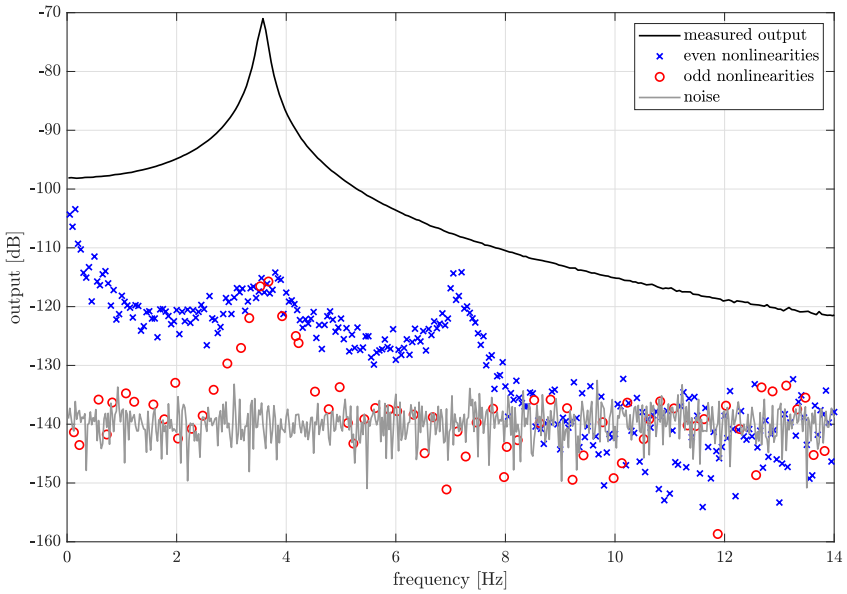
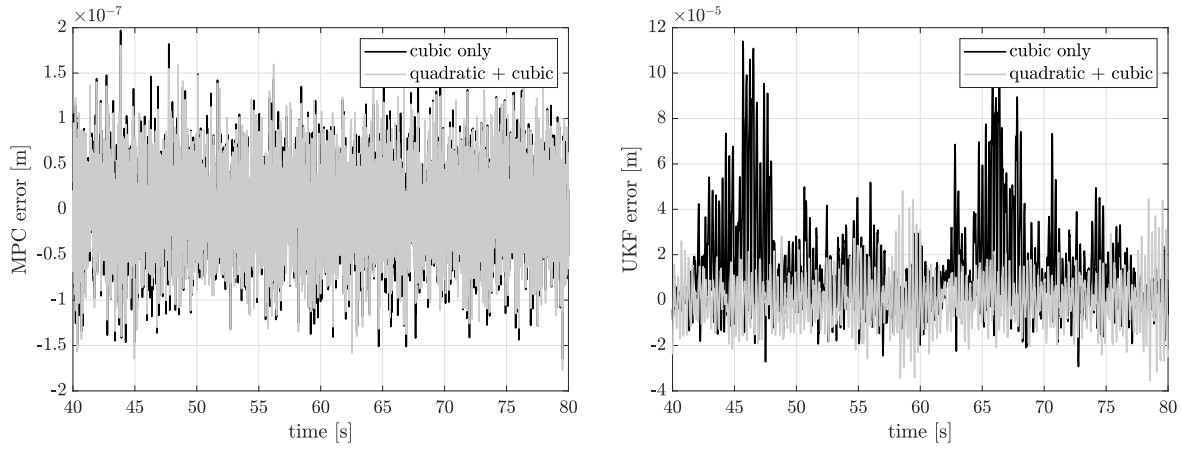


Figure 3.15: Nonlinear distortion analysis after linearisation with the incomplete model. Still, decent performance regarding the odd nonlinearities is achieved. However, only very little even nonlinearities are suppressed.



(a) (b)

Figure 3.16: MPC (a) and UKF (b) error plots for one period of the multisine. The plot shows the errors for the complete (quadratic + cubic) and incomplete model (cubic only). The MPC handles modelling errors very well due to its integral action property. On the contrary, the performance of the UKF significantly degrades when the quadratic nonlinearity is not modelled.

of the even nonlinearities is caused by. A significant increase of the UKF is obtained for the incomplete model. Table 3.3 presents the RMS values of all the errors and the ratio of these errors compared to the measured output. The values confirm the observations from the error plots; the MPC is able to track the reference equally well whereas the UKF error more than doubles. Concluding, the incomplete model is not able provide an accurate estimate of the clean output, which explains the poor reduction of the even nonlinearities in Figure 3.15.

3.3.2 Extrapolation Properties of the Control Framework

So far, a random-phase multisine with an RMS amplitude of 0.12 N was used for identification and performance analysis. In this section, the RMS amplitude is increased to 0.22 N in order to study the extrapolation properties of the entire control structure. By doing so, the linearisation becomes more challenging in two ways; namely, 1) the model is extrapolated outside its fitting domain, and 2) a higher input amplitude induces more nonlinearities to be eliminated. Figure 3.17 shows the nonlinear distortion analysis for specified amplitude level. A significant increase in nonlinear distortions is observed when compared to the analysis at 0.12 Hz of Figure 3.7. Also, when compared to Figure 3.7, the magnitude of the measured output has increased by approximately 10 dB over the entire frequency range.

Figure 3.18 shows the nonlinear distortion analysis after linearisation. Although the RMS amplitude was increased significantly, still a satisfactory linearisation is achieved; the results are rather similar to Figure 3.13, but the peaks around 3.5 and 7 Hz are slightly amplified. It is interesting to see that the low-frequent performance is considerably better than in the case of the model without the quadratic nonlinearity in Figure 3.15. Again, the MPC and UKF error plots are consulted to provide more insight in this matter. Figure 3.19 shows these plots, while Table 3.3 presents the corresponding RMS values. The magnitude of the MPC error is almost similar

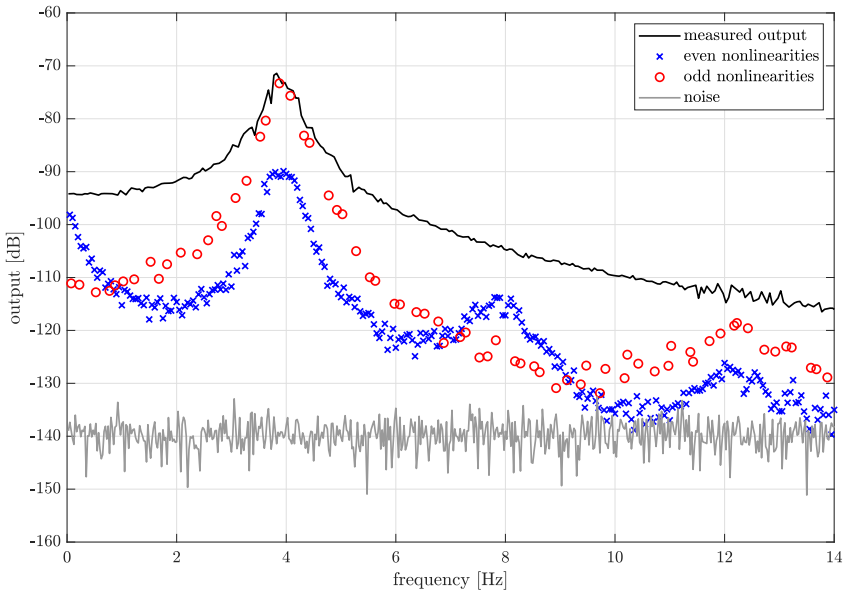


Figure 3.17: Nonlinear distortion analysis before linearisation when excited with random-phase multisines with an increased RMS amplitude level of 0.22 N.

	measured output	MPC error	UKF error
cubic only	$4.14 \cdot 10^{-4}$	$2.27 \cdot 10^{-7}$ (0.0548%)	$2.64 \cdot 10^{-5}$ (6.38%)
quadratic + cubic	$3.82 \cdot 10^{-4}$	$2.22 \cdot 10^{-7}$ (0.0581%)	$1.14 \cdot 10^{-5}$ (2.98%)
extrapolated	$6.51 \cdot 10^{-4}$	$2.36 \cdot 10^{-7}$ (0.0363%)	$2.12 \cdot 10^{-5}$ (3.26%)

Table 3.3: RMS values [m] of the measured output, MPC error and UKF error; for the complete and incomplete model (test 1), and the complete model when extrapolated (test 2). The percentages between brackets show the ratio of the errors compared to the measured output.

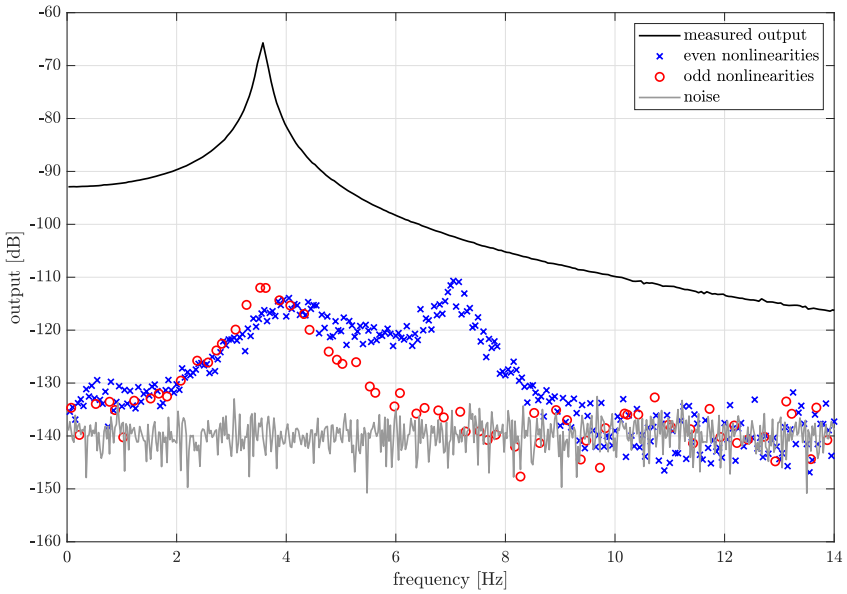
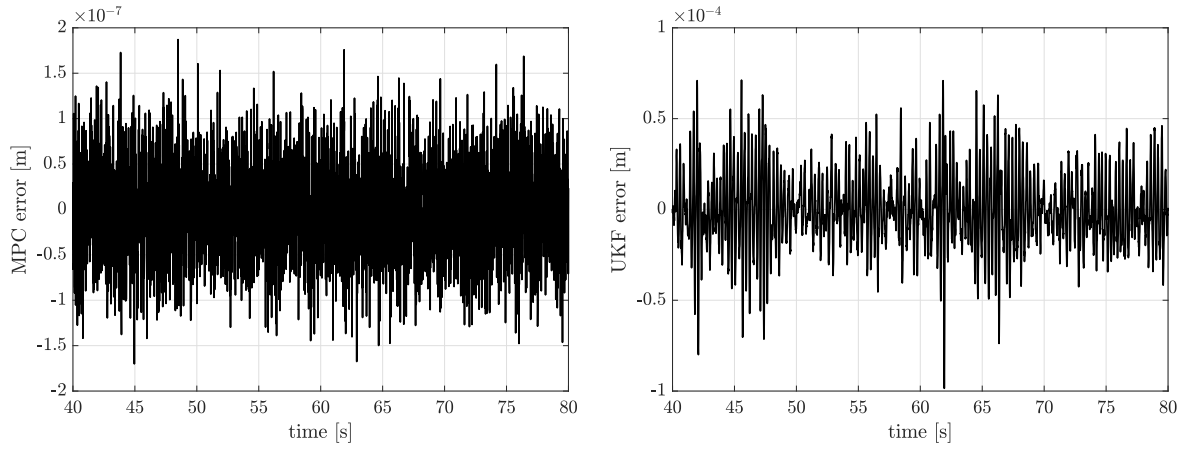


Figure 3.18: Nonlinear distortion analysis after linearisation when excited with random-phase multisines with an increased RMS amplitude of 0.22 Hz. The performance is comparable to the performance for the RMS amplitude of 0.12 in Figure 3.13, only the peaks around 3.5 and 7 Hz are slightly amplified.



(a) (b)

Figure 3.19: MPC (a) and UKF (b) error plots for one period of the multisine with increased RMS amplitude of 0.22 Hz. The MPC error is very similar to the errors in Figure 3.16. The UKF error has grown significantly due to the higher excitation signal. However, as opposed to Figure 3.16, the error signal has no offset, which is beneficial in terms of linearising performance.

to the magnitudes in the previous test, even though the RMS value of the measured output has increased by more than 70 percent. This even more confirms the excellent working of the integral action property. As expected, the magnitude of the UKF error has also increased significantly. Yet, its percentage when compared to the output RMS is comparable to the percentage of the full nonlinear model. It is therefore hypothesised that the poor low-frequent performance in the previous robustness test is due to the nature of the UKF error in Figure 3.16; namely,

there clearly exists an offset in the error signal. From a physical point of view, this offset is explainable since the quadratic nonlinearity, i.e., the spring asymmetry, is not modelled. In the frequency domain, an offset is represented as a frequency at 0 Hz. It is therefore coherent that the low-frequency region is mostly affected by the UKF-induced offset.

3.4 Conclusions of the Chapter

In this chapter, the proposed control solution of this work was tested for various simulation examples, presented in order of increasing difficulty. Under ideal conditions, almost perfect linearisation was achieved over the entire frequency range of interest. Similarly, great results were obtained for the more realistic simulation environment. The robustness tests revealed that the MPC is very robust to severe modelling errors. However, the linearising performance is mainly limited by the UKF, which is significantly less robust to those errors. Overall, the results in numerical conditions are of excellent quality, and are therefore promising in view of the application of the proposed methodology in real-life conditions. This will be confirmed in the next chapter, where the feedback linearisation framework is integrated into a nonlinear experimental test setup.

Chapter 4

Validation on an Experimental Setup

In this chapter, the proposed control structure is demonstrated on an experimental setup. The considered setup is a prototype for a high-precision flexible positioning system, developed for evaluating control strategies. The system, shown in Figure 4.1, consists of a lightweight flexible steel beam with dimensions $500 \times 20 \times 2$ mm. The boundary conditions of the beam are realised by means of leaf springs, which fix 4 out of 6 motion degrees of freedom (DOFs). This means that the beam can only exhibit motion in the translational and rotational directions. The setup is equipped with 3 collocated sensor-actuator pairs, operating at a sampling frequency of 4096 Hz. The actuators are current-driven voice-coil actuators whereas the sensors are contactless fiberoptic displacement sensors with an approximate accuracy of 1 μm. Since the system has more actuators than DOFs, it is over-actuated. However, over-actuation is not considered in this work; in fact, only the middle sensor-actuator pair is used.

First, the nonlinearities in the setup are qualified and quantified. The system is excited with 5 realisations of 4 periods of an odd random-phase multisine with an RMS amplitude of 0.015 V. For every group of 4 odd frequency lines, 1 line is randomly rejected. Moreover, 3280 points per period are considered at a sampling frequency of 102.4 Hz. The resulting nonlinear distortion

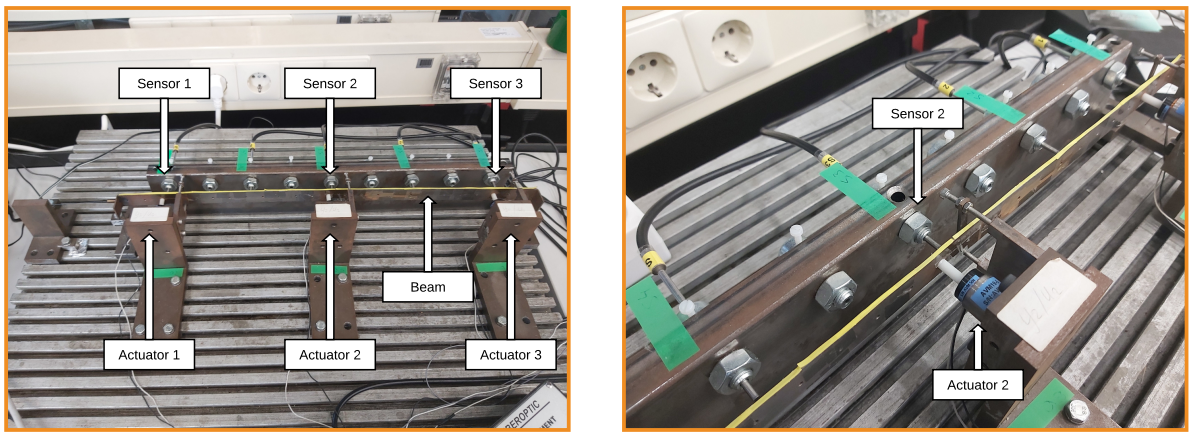


Figure 4.1: Photographs of the experimental flexible beam setup, equipped with 3 collocated sensor-actuator pairs. In this work, only the middle pair is considered.

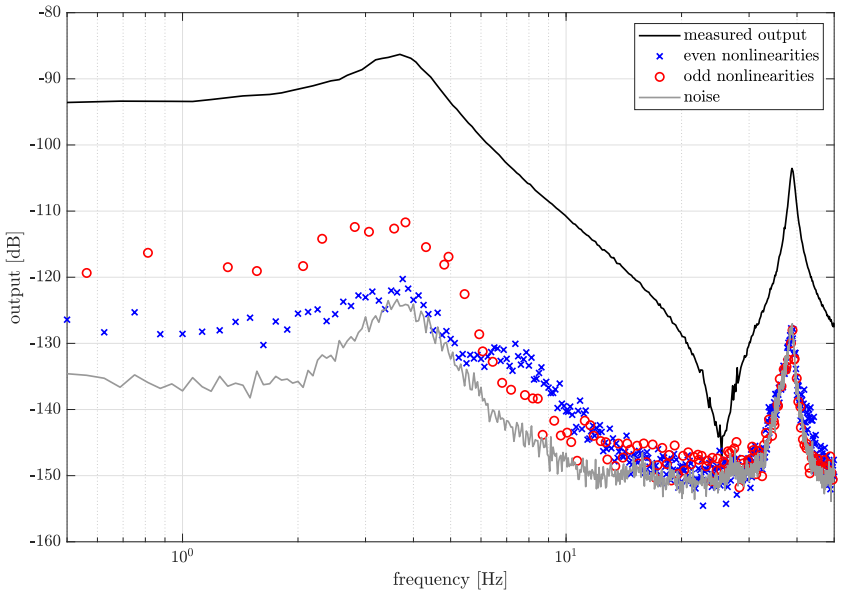


Figure 4.2: Nonlinear distortion analysis of the flexible beam. The resonance phenomenon around 4 Hz corresponds to the rigid-body mode of the flexible beam, whereas the resonance phenomenon around 40 Hz corresponds to its first flexible mode.

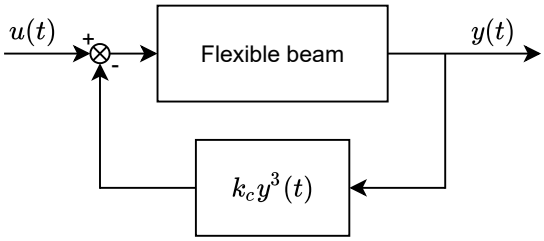


Figure 4.3: Block diagram of the nonlinear setup. A hardening spring is artificially created by feeding back the output sensor measurements, through a cubic function, to the input. The cubic spring constant is set to $k_c = 2 \cdot 10^9$ V/m^3.

analysis is shown in Figure 4.2. Here, the resonance phenomenon around 4 Hz corresponds to the rigid-body mode of the flexible beam, whereas the resonance phenomenon around 40 Hz corresponds to its first flexible mode. The nonlinear distortions are only marginally present. In order to increase the amount of nonlinearities in the system, a cubic nonlinearity, i.e., a hardening spring, is artificially implemented as is shown in Figure 4.3. The value of the nonlinear spring constant is set to $k_c = 2 \cdot 10^9$ V/m^3. By doing so, the setup obeys the structure of an underlying (almost) linear system with a static nonlinear feedback.

Figure 4.4 shows the nonlinear distortion analysis of the system with artificial nonlinearity. The inclusion of the hardening spring caused a significant increase of odd nonlinearities; which are now (less than) 10 dB below the measured output. The nonlinear distortions are also responsible for the seemingly noisy behaviour of the measured output. Note that, in contrast to the results in Chapter 3, the noise level in Figure 4.4 is not flat. From a physical point of view, it is irrational that the noise level grows around the resonance modes, since the sensor noise should be independent of the system dynamics. Note however that the output noise is calculated as the standard deviation on the mean value of the FRF. This estimation was made under the

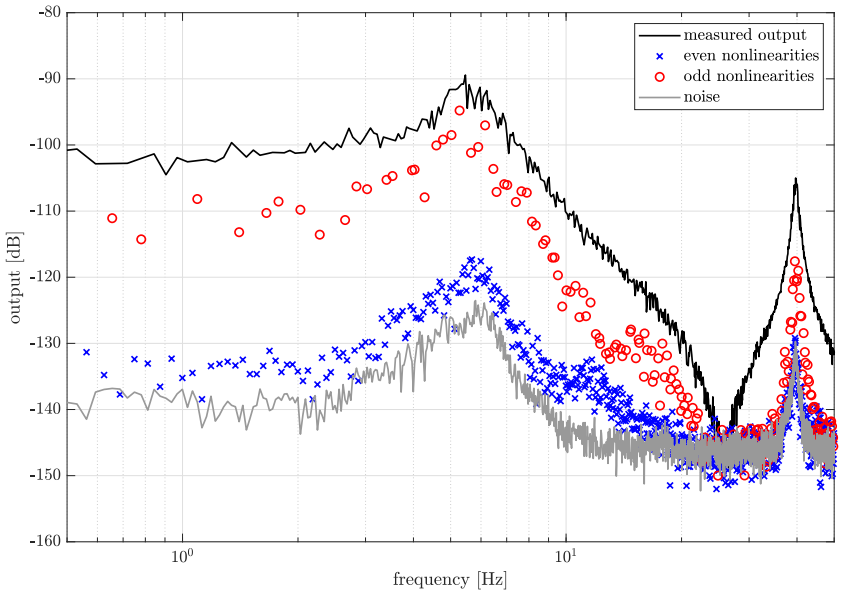


Figure 4.4: Nonlinear distortion analysis after inclusion of the artificial hardening spring. The odd nonlinear distortions have increased significantly when compared to Figure 4.2.

assumption that the input signal was error-free and independent of the output noise, which was true for the results in Chapter 3. However, this is not necessarily true for the experimental setup, which might explain that the noise levels in this chapter are seemingly biased.

4.1 System Identification

The identification data for the system is obtained by performing 10 realisations of 5 periods of a full random-phase multisine with an RMS amplitude of 0.015 V. The number of data points per period is 16 384 at a sampling frequency of 1024 Hz. To ensure steady-state conditions, the first period of every realisation is rejected. Moreover, 1 realisation is saved for validation. As for linearisation, the primary frequency range of interest is from 0 to 15 Hz, i.e., the rigid-body range. Yet, in order to study the performance of the proposed framework at the first flexible mode, the frequency range of interest is extended to 50 Hz. Figure 4.5 shows the fitting results of the nonparametric and parametric BLAs in frequency frequency domain. The model order of the parametric BLA is 4. The results are in line with the identification findings in the previous chapter in the sense that the parametric BLA is performing in the best way possible; namely, the residual of the estimate has a similar magnitude as the total standard deviation on the nonparametric BLA. As for the nonlinear model optimisation, the vector of nonlinear monomial is simply $\zeta(y(t)) = y^3(t)$; the small even distortions in Figure 4.4 are not modelled by an even monomial since they are considered negligible. Figure 4.6 shows the results after parametrisation of the full nonlinear model, when exposed to validation data. The left figure shows one time-domain period of validation data with the corresponding linear and nonlinear modelling errors; similar results as in Section 3.2.1 are obtained, the full nonlinear parametrisation reduced the identification error significantly, from 57.4% to 7.34%. In the right figure, the same validation data is presented in the frequency domain. Up until 15 Hz, the nonlinear error is consistently (more than) 20 dB below the validation data. As mentioned, the noise level is not fully trustworthy, which might explain the distorted impression of a seemingly perfect identification. The parameter values of

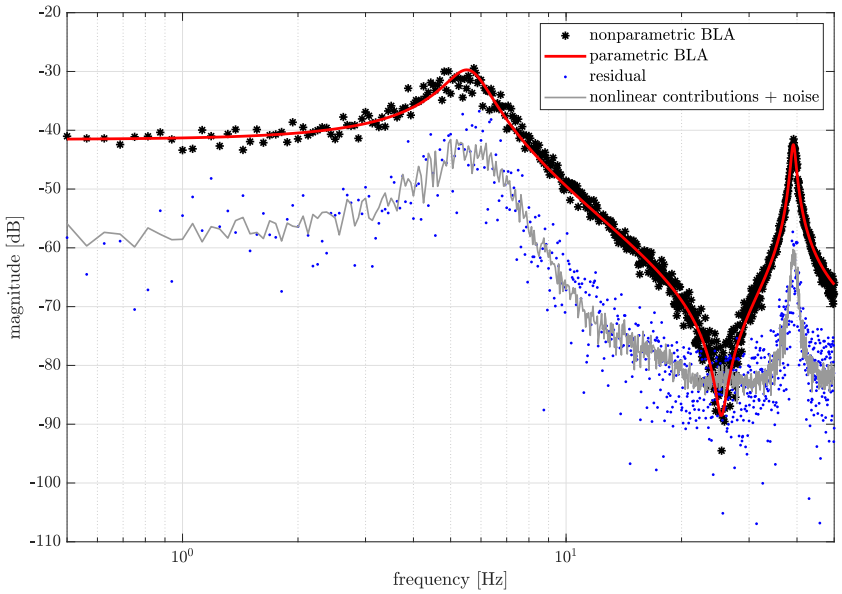
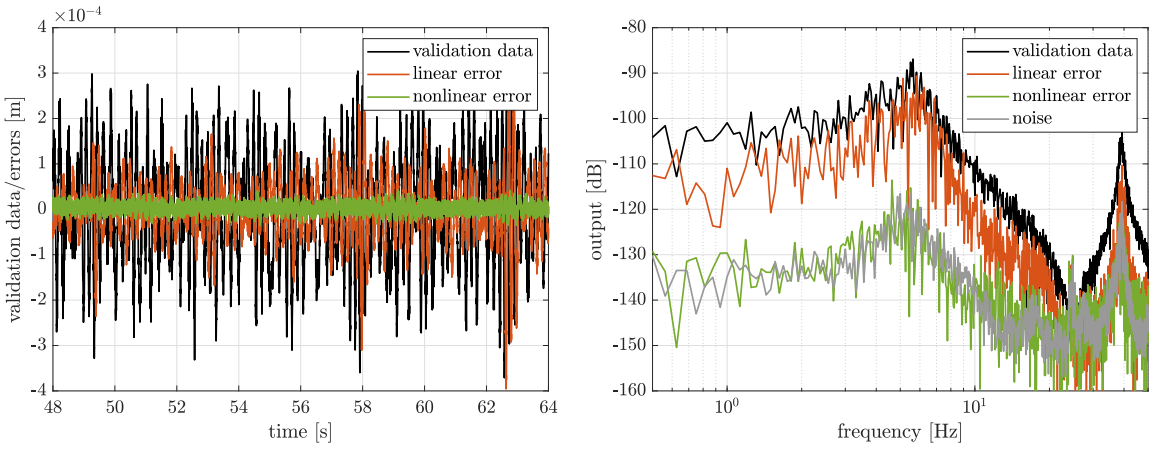


Figure 4.5: Fitting results of the parametric and nonparametric BLA in the frequency domain. The parametric BLA is estimated in the best way possible, since the model residual is equal to the sum of the nonlinear distortions and the disturbing noise.



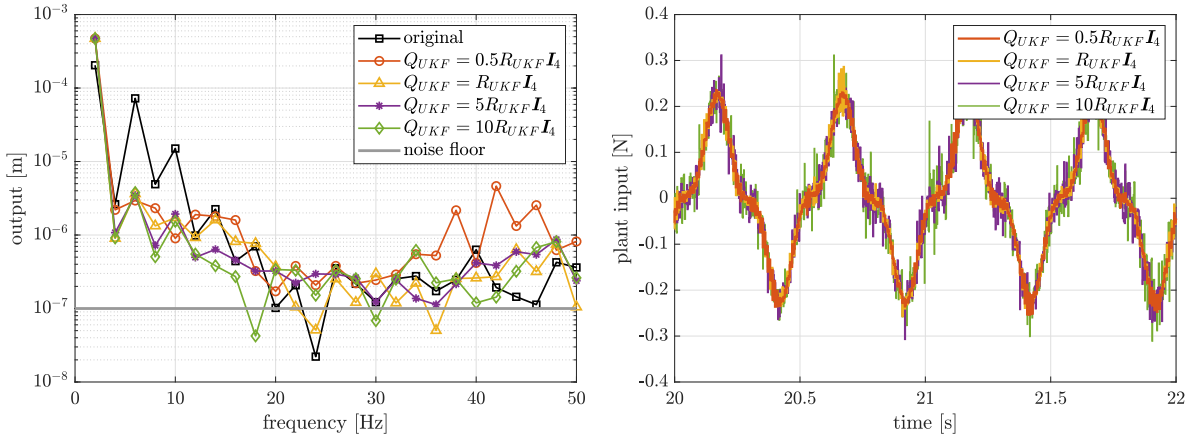
(a) (b)

Figure 4.6: One period of time-domain validation data and the corresponding identification errors of the parametric BLA and the full nonlinear model (a). Here, the identification error is reduced from 57.4% to 7.34%. The same data is also represented in the frequency domain (b).

the obtained nonlinear state-space model are provided in Table B.2 of Appendix B.

4.2 Selecting the Tuning Parameters

The tuning parameters for the setup are chosen following the guidelines of Section 2.1.1. Here, one must also consider the sampling time of the sensor-actuator pair. Namely, the nonlinear feedback structure of Figure 4.3 represents the continuous plant. However, the sensor output is only subtracted from the input at the next time step, considering a sample rate of 4096 Hz.



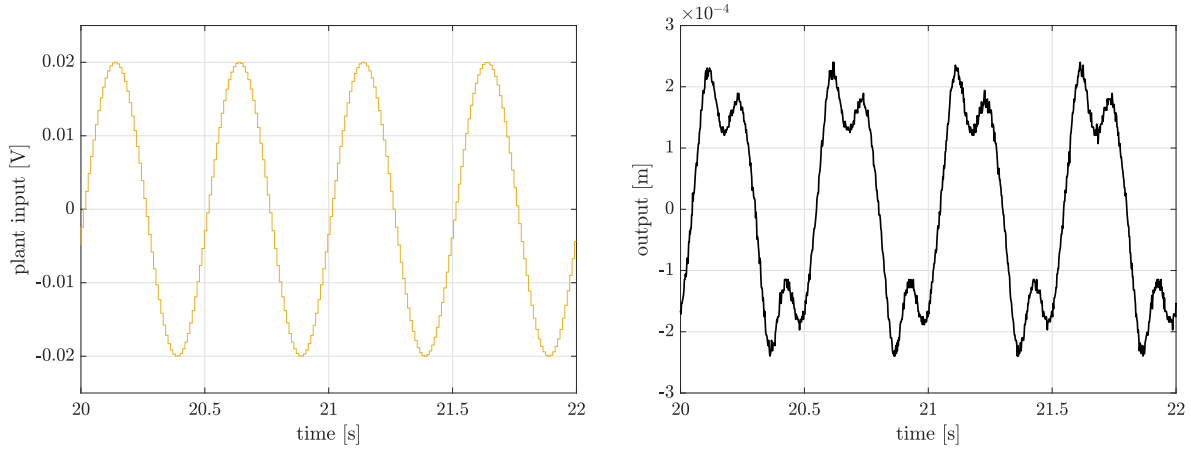
(a) (b)

Figure 4.7: Output amplitude spectra (a) for different diagonall values of Q_{UKF}, when the system is excited by an outer-loop sine excitation signal, and the required plant input signals (b). It can be observed that better linearising performance is achieved for higher diagonal values of Q_{UKF}, while the aggressiveness is only increased moderately.

Therefore, the MPC sampling time should be sufficiently higher than 1/4096 s. While keeping this in mind, the outer-loop sample time is set to 1/102.4 s. In this way, a desired closed-loop bandwidth of around 20 Hz can be achieved by an outer-loop controller. Next, the sampling time of the MPC is set to 1/512 s, i.e., 5 times faster than the outer-loop while still sufficiently below 1/4096 Hz. The values of Q and R_Δ are determined by performing noise-free experiments with the obtained model in simulation. The values of the covariance matrices of the UKF are tuned on the setup itself, which is done in a similar fashion as in Chapter 3. Figure 4.7 shows the response of the controlled plant when a sine input with amplitude 0.02 V and frequency 2 Hz is applied as outer-loop excitation signal. In the left figure, the output amplitude spectrum is shown for different diagonal values of Q_{UKF}, whereas the right figure shows the corresponding plant inputs. Again, the measurement noise covariance is set to the mean value of the output noise covariance of the nonlinear distortion analysis, i.e., $R_{UKF} = 3.39 \cdot 10^{-15}$ m^2. The experimental results are in line with the simulation findings in the sense that higher diagonal values of Q_{UKF} lead to better linearising performance. At the same time, the required control action becomes more aggressive; however, not as excessive as in Chapter 3, since the diagonal values of Q_{UKF} are not as spread out as in Figure 3.11. Based on the observations, $Q_{UKF} = 10R_{UKF}\boldsymbol{I}_4$ is picked as the optimal trade-off between linearising performance and control aggressiveness. All tuning parameters are summarised in Table 4.1.

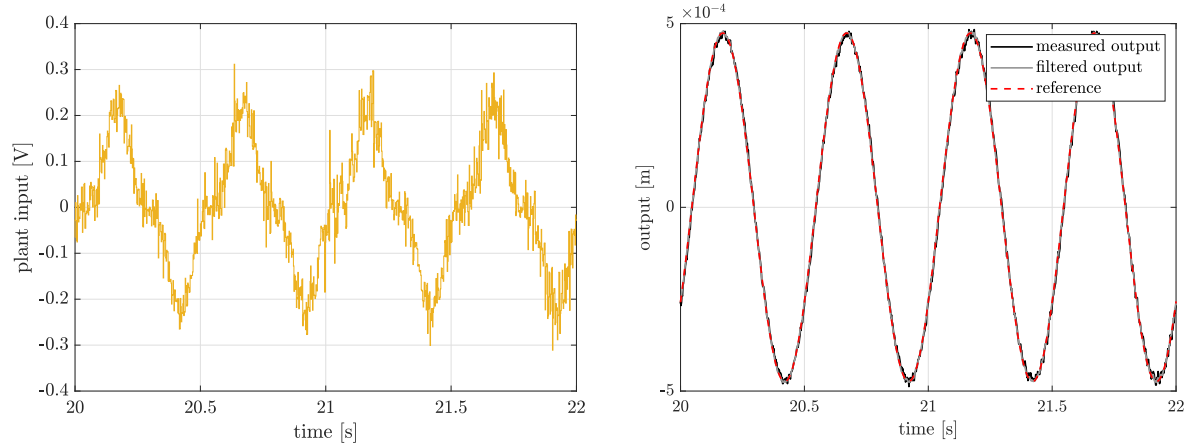
4.3 Results

This section presents the results of the linearising control framework for both a sine excitation test and the nonlinear distortion analysis. Figure 4.8 shows the plant input and output when the uncontrolled plant is excited with a sine signal. The sine has an amplitude of 0.02 V and an excitation frequency of 2 Hz. The effect of the nonlinear spring is clearly visible in the output plot. The response of the controlled plant to the same sine excitation is shown in Figure 4.9. It can be observed that, in order to track the linear output, the peak amplitudes of the required plant input are more than a factor 10 higher than the amplitude of the outer-loop sine input. In contrast to the simulation results of Figure 3.11b, this is not because the MPC reacts



(a) (b)

Figure 4.8: Plant input and output when the uncontrolled plant is excited with a sine signal with an amplitude of 0.02 V and an excitation frequency of 2 Hz.



(a) (b)

Figure 4.9: Plant input and output when the controlled plant is excited with the outer-loop sine signal of Figure 4.8. A significant increase of input amplitude is required to overcome the severe hardening effect of the spring.

	$t_{s,out}$	$t_{s,in}$	Q	R_Δ	Q_{UKF}	R_{UKF}
value	1/102.4	1/512	$3 \cdot 10^8$	1	$3.39 \cdot 10^{-14} \boldsymbol{I}_4$	$3.39 \cdot 10^{-15}$
unit	s	s	-	-	-	m^2

Table 4.1: Summary of the tuning parameters for the experimental setup.

aggressively on a too tightly tuned UKF; here, the high amplitude is required to compensate for the severe hardening effect of the spring. Figure 4.10 shows the zoomed-in version of the output plot in Figure 4.9b. From the figure, it can be observed that the MPC is tracking its reference very well, i.e., with an error of 1.05%. Moreover, the UKF filters the measured output better than in the simulation examples, i.e., there are no offsets at the sine peaks as in Figure 3.12. Yet, the simulation model and the experimental model have comparable identification errors when exposed to validation data, i.e., 6.64% and 7.34%, respectively.

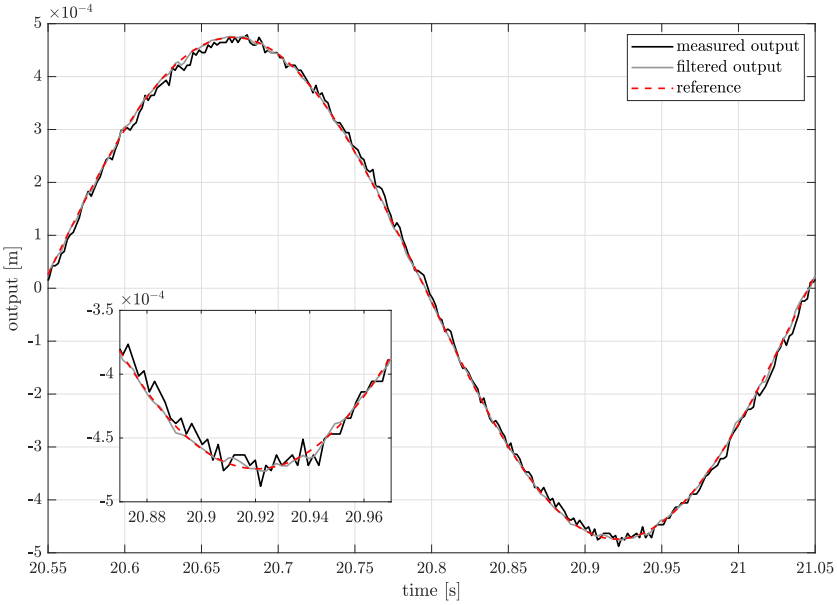


Figure 4.10: Zoomed-in version of Figure 4.9b. It can be seen that the reference is tracked very well by the MPC, i.e., with an error of 1.05%. Moreover, the UKF correctly filters the measured output; as opposed to Figure 3.12, there are no offsets at the sine peaks.

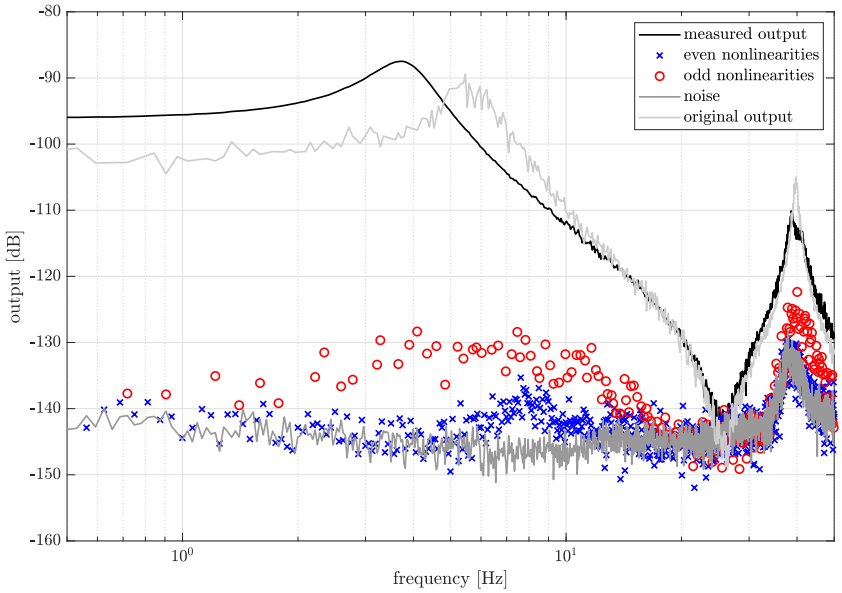


Figure 4.11: Nonlinear distortion analysis after linearisation, where the odd nonlinearities are suppressed significantly over the primary frequency range of interest. The seemingly noisy behaviour of the uncontrolled output is entirely removed; moreover, from the shift of the resonance phenomenon is clearly visible that the effect of the hardening spring is removed.

Figure 4.11 shows the nonlinear distortion analysis after linearisation. Excellent linearisation is achieved over the primary frequency range of interest; all distortions are located within 10 dB of the noise floor. Around the resonance frequencies, the odd distortions are suppressed with more

than 30 dB. The even nonlinearities around the resonance frequency are also suppressed, even though the cubic nonlinearity is modelled only. As expected, there is no significant performance improvement around the first flexible mode of the beam. For comparison reasons, Figure 4.11 also shows the output of the plant before linearisation. In this way, it becomes clearly visible that the effect of the hardening spring is eliminated. Namely, after linearisation, the resonance phenomenon is shifted to the left while the low-frequency magnitude of the output has increased, exactly as depicted in Figure 3.1. Moreover, the seemingly noisy behaviour is entirely eliminated.

4.4 Conclusions of the Chapter

In this chapter, the proposed control solution was demonstrated on a high-precision experimental setup. In order to increase the level of nonlinearity in the system, a significant hardening spring was artificially implemented. In line with the simulation findings of the previous chapter, the results over the primary frequency range of interest are of excellent quality, in this way validating the proposed control framework. This is particularly promising in view of the general applicability of the proposed methodology on real-life setups.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

In this thesis, a novel data-driven MPC approach towards feedback linearisation of nonlinear mechanical systems was presented. The resulting framework is particularly designed for nonlinear vibrating systems in the Lur'e form. These systems can be seen as underlying stable systems with undesirable static nonlinearities in the feedback loop. Therefore, a linearising reference tracking framework was proposed that eliminates the undesirable nonlinearities, while at the same time preserving the important linear dynamics. Based on a new outer-loop input, an online reference trajectory with time-varying length is generated at every discrete time step. By tracking this reference, the IO relation between the new outer-loop input and the measured output is governed by the desirable linear part of the nonlinear model. The tracking problem was solved for by means of an MPC controller, which, due to its integral action property, achieved very low tracking errors. Moreover, by smartly utilising the available reference information, the apparent nonlinear MPC control problem was transformed into a computationally efficient linear optimisation problem with an analytical global minimum.

The exploited models of this work were identified by exclusively processing IO data. The rationale behind the choice for these models was that data-driven models capture more complex nonlinearities more easily, whereas on the other hand, first-principle modelling is often infeasible for very complex systems. In line with the current developments in industry, i.e, systems are becoming increasingly complex, it is expected that data-driven models will become even more popular. However, in practice, classical feedback linearisation cannot handle such data-driven models. The contrary is true for the presented solution; data-driven models were seamlessly integrated into the MPC framework, which is undoubtedly a key contribution of this work.

Concluding, the presented framework incorporates all the steps from nonlinear system identification to controller implementation, as well as an intuitive nonparametric nonlinearity analysis that allows for quantification of the linearising performance over the specified frequency range of interest. Robustness tests in a realistic simulation environment showed that the MPC controller still achieves very low tracking errors in case of severe modelling errors, or when the model is extrapolated outside its fitting region. Here, linearising performance was mainly hampered by the UKF, which relies significantly more on an accurate model. Excellent experimental results of the control framework on a highly nonlinear positioning system confirmed the correct working of the proposed framework.

5.2 Recommendations

In this section, recommendations for future works are proposed. This concerns deeper analysis of particular mechanisms, possible solution directions, or simply curiosity.

Improving performance of the nonlinear observer: Particularly in simulation, the main limiting factor of performance was the UKF. Compared to the MPC controller, the UKF is significantly less robust to modelling errors. The UKF noise covariance matrices can be tuned such that the output is tracked and filtered better, but this quickly leads to more aggressive control inputs. In other words, the mutual dependence of every tuning parameter of the MPC and UKF made it a difficult task to properly tune the entire framework. It could be investigated how its performance can be improved, for example by finding a more pragmatic approach towards tuning the covariance matrices. Note that the presented framework is however not restricted to the UKF, any nonlinear state observer could work.

Extension to the multi-input multi-output (MIMO) framework: Classical feedback linearisation can be extended to the MIMO case under the condition that a system is *square*, i.e., it has as many inputs as outputs. The proposed solution in this study consists of three important components; namely, nonlinear system identification, the MPC controller, and the UKF. Separately, these components can be extended to the MIMO framework relatively easy. Moreover, none of them are restricted to the *square* condition. It would be an interesting study to integrate the three components into the more challenging MIMO setting.

Extension to systems outside mechanics: The focus of this thesis was on a rather particular type of system, i.e, nonlinear mechanical vibrating systems that can be represented in the Lur'e form. It would be an interesting study to extend the framework to systems outside of mechanics, or, systems that cannot be represented by the Lur'e form. Specifically this would mean that the system may contain input or state nonlinearities. Besides, when linearisation is concerned only, it is no hard requirement for the underlying linear dynamics to be stable; namely, in case of unstable linear dynamics, any other stable linear reference could be tracked instead. As for nonlinear system identification, the same procedure as presented here can be followed, taking into account that the nonlinear optimisation would consider input or states nonlinearities instead. Likewise, the UKF can handle such nonlinearities. For the MPC controller it would be more challenging, since it needs a reliable estimate of the vector of future disturbances. This is intuitive for the output nonlinearity case, where future reference information can be used to create this vector. Regarding input nonlinearities, a logical choice would be to use the predicted sequence of inputs of the previous time step for the estimation of the disturbance vector of the current time step. Here, one must be very careful that the predicted sequence of inputs is not too fluctuating. That is to say, when (higher order) monomials are considered, large fluctuations will amplify and propagate through the next predictions, which quickly leads to infeasible inputs. Another possible solution direction would be to utilise the rationale of a typical feedforward controller; namely, by making use of the available reference information, an estimate of the future inputs can be made by feeding the reference through an inverse model of the plant. Indeed, inverting a complex nonlinear plant is far from straightforward and would yield an approximation rather than an exact representation, but a moderate approximation might suffice due to the integral action property of the MPC controller. Note that if satisfactory performance is achieved for systems with input nonlinearities, another limitation of classical feedback linearisation, i.e., the restriction to input-affine systems, is overcome. Moreover, when a reliable estimate of the future inputs is obtained, one can simply simulate the model for the given input sequence in order to obtain an estimate of the future states.

Extension to truly complex systems: Following from the previous recommendation, extensions could be made to systems that contain multiple monomial combinations of the input, output and states. After all, a motivation for the use of data-driven models was that it is often impossible to model very complex systems by first-principle based techniques. Such systems can be identified by black-box modelling techniques, which typically combine these combined monomials up to a certain degree in the nonlinear basis function. Although the magnitude of the nonlinearities in this thesis were significant, the nonlinear basis functions and the model order were manageable. A more extensive study is required to investigate the practical applicability and generalisability of the presented framework to more complex real-life systems.

Bibliography

[1] T. Dursun and C. Soutis, "Recent developments in advanced aircraft aluminium alloys," *Materials & Design (1980-2015)*, vol. 56, pp. 862–871, 2014.

[2] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, "Nonlinear system identification in structural dynamics: current status and future directions," in *25th International Modal Analysis Conference, Orlando, 2007*, 2007.

[3] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer Science & Business Media, 2013, vol. 10.

[4] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[5] D. Wagg and S. Neild, *Nonlinear vibration with control*. Springer, 2016.

[6] T. Huang, D. Huang, Z. Wang, N. Qin, and A. Shah, "Robust control for a quadrotor uav based on linear quadratic regulator," in *2020 39th Chinese Control Conference (CCC)*. IEEE, 2020, pp. 6893–6898.

[7] N. Gionfra, H. Siguerdidjane, G. Sandou, D. Faille, and P. Loevenbruck, "Combined feedback linearization and MPC for wind turbine power tracking," in *2016 IEEE Conference on Control Applications (CCA)*. IEEE, 2016, pp. 52–57.

[8] H. Moradi, A. Alasty, and F. Bakhtiari-Nejad, "Control of a nonlinear boiler-turbine unit using two methods: gain scheduling and feedback linearization," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 43033, 2007, pp. 491–499.

[9] A. Arapostathis, B. Jakubczyk, H.-G. Lee, S. Marcus, and E. Sontag, "The effect of sampling on linear equivalence and feedback linearization," *Systems & control letters*, vol. 13, no. 5, pp. 373–381, 1989.

[10] J. Grizzle and P. Kokotovic, "Feedback linearization of sampled-data systems," *IEEE Transactions on Automatic Control*, vol. 33, no. 9, pp. 857–859, 1988.

[11] K. Bos, D. Heck, M. Heertjes, and R. van der Kall, "IO linearization, stability, and control of an input non-affine thermoelectric system," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 526–531.

[12] J.-P. Noël and J. Schoukens, "Grey-box state-space identification of nonlinear mechanical vibrations," *International Journal of Control*, vol. 91, no. 5, pp. 1118–1139, 2018.

[13] A. Lur'e and V. Postnikov, "On the theory of stability of control systems," *Prikladnaya Matematika Mehkhanika*, vol. 8, no. 3, pp. 246–248, 1944.

[14] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, "Past, present and future of nonlinear system identification in structural dynamics," *Mechanical systems and signal processing*, vol. 20, no. 3, pp. 505–592, 2006.

[15] J.-P. Noël and G. Kerschen, "Nonlinear system identification in structural dynamics: 10 more years of progress," *Mechanical Systems and Signal Processing*, vol. 83, pp. 2–35, 2017.

[16] O. Krupková and D. Saunders, "Remarks on the history of the notion of lie differentiation."

[17] E. Kayacan and T. I. Fossen, "Feedback linearization control for systems with mismatched uncertainties via disturbance observers," *Asian Journal of Control*, vol. 21, no. 3, pp. 1064–1076, 2019.

[18] J. S. Velez-Ramirez, L. A. Rios-Norena, and E. Giraldo, "Buck converter current and voltage control by exact feedback linearization with integral action."

[19] T. I. Fossen and M. Paulsen, "Adaptive feedback linearization applied to steering of ships," 1993.

[20] H. Guo, R. Yu, W. Qiang, and H. Chen, "Optimal slip based traction control for electric vehicles using feedback linearization," in *2014 International Conference on Mechatronics and Control (ICMC)*. IEEE, 2014, pp. 1159–1164.

[21] J. A. Rossiter, *Model-based predictive control: a practical approach*. CRC press, 2003.

[22] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[23] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice — a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[24] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*. Birkhäuser, 2012, vol. 26.

[25] L. Wang, *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.

[26] G. Papafotiou, J. Kley, K. G. Papadopoulos, P. Bohren, and M. Morari, "Model predictive direct torque control—part ii: implementation and experimental evaluation," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1906–1915, 2009.

[27] J. Rodríguez, R. M. Kennel, J. R. Espinoza, M. Trincado, C. A. Silva, and C. A. Rojas, "High-performance control strategies for electrical drives: an experimental assessment," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 812–820, 2011.

[28] P. Cortes, G. Ortiz, J. I. Yuz, J. Rodriguez, S. Vazquez, and L. G. Franquelo, "Model predictive control of an inverter with output LC filter for UPS applications," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1875–1883, 2009.

[29] H. Miranda, P. Cortés, J. I. Yuz, and J. Rodríguez, "Predictive torque control of induction machines based on state-space models," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1916–1924, 2009.

[30] W. Van Soest, Q. Chu, and J. Mulder, "Combined feedback linearization and constrained model predictive control for entry flight," *Journal of guidance, control, and dynamics*, vol. 29, no. 2, pp. 427–434, 2006.

[31] D. Simon, J. Löfberg, and T. Glad, "Nonlinear model predictive control using feedback linearization and local inner convex constraint approximations," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 2056–2061.

[32] J. Rehrl and M. Horn, "Temperature control for hvac systems based on exact linearization and model predictive control," in *2011 IEEE International Conference on Control Applications (CCA)*. IEEE, 2011, pp. 1119–1124.

[33] A. T. Hafez, M. Iskandarani, S. N. Givigi, S. Yousefi, C. A. Rabbath, and A. Beaulieu, "Using linear model predictive control via feedback linearization for dynamic encirclement," in *2014 American Control Conference*. IEEE, 2014, pp. 3868–3873.

[34] M. A. Kamel and Y. Zhang, "Linear model predictive control via feedback linearization for formation control of multiple wheeled mobile robots," in *2015 IEEE International Conference on Information and Automation*. IEEE, 2015, pp. 1283–1288.

[35] A. J. Krener, "Adaptive horizon model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 31–36, 2018.

[36] H. Arai and Y. Uchimura, "Model predictive control with variable prediction horizon for a system including time-varying delay," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1. IEEE, 2019, pp. 3281–3286.

[37] G. Droge and M. Egerstedt, "Adaptive time horizon optimization in model predictive control," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 1843–1848.

[38] P. Tatjewski, "Disturbance modeling and state estimation for offset-free predictive control with state-space process models," *International Journal of Applied Mathematics and Computer Science*, vol. 24, no. 2, pp. 313–323, 2014.

[39] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free model predictive control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, 2009.

[40] G. Pannocchia and J. B. Rawlings, "Disturbance models for offset-free model-predictive control," *AIChE journal*, vol. 49, no. 2, pp. 426–437, 2003.

[41] A. H. González, E. J. Adam, and J. L. Marchetti, "Conditions for offset elimination in state space receding horizon controllers: A tutorial analysis," *Chemical Engineering and Processing: Process Intensification*, vol. 47, no. 12, pp. 2184–2194, 2008.

[42] L. Wang, "A tutorial on model predictive control: Using a linear velocity-form model," *Developments in Chemical Engineering and Mineral Processing*, vol. 12, no. 5-6, pp. 573–614, 2004.

[43] G. Betti, M. Farina, and R. Scattolini, "A robust mpc algorithm for offset-free tracking of constant reference signals," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2394–2400, 2013.

[44] M. Morari and U. Maeder, "Nonlinear offset-free model predictive control," *Automatica*, vol. 48, no. 9, pp. 2059–2067, 2012.

[45] R. Huang, L. T. Biegler, and S. C. Patwardhan, "Fast offset-free nonlinear model predictive control based on moving horizon estimation," *Industrial & Engineering Chemistry Research*, vol. 49, no. 17, pp. 7882–7890, 2010.

[46] D. Q. Mayne, E. C. Kerrigan, E. Van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[47] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of 1995 American Control Conference-ACC'95*, vol. 3. IEEE, 1995, pp. 1628–1632.

[48] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. International Society for Optics and Photonics, 1997, pp. 182–193.

[49] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.

[50] J. J. LaViola, "A comparison of unscented and extended Kalman filtering for estimating quaternion motion," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 3. IEEE, 2003, pp. 2435–2440.

[51] B. F. La Scala and R. R. Bitmead, "Design of an extended Kalman filter frequency tracker," *IEEE Transactions on Signal Processing*, vol. 44, no. 3, pp. 739–742, 1996.

[52] R. Kandepu, B. Foss, and L. Imsland, "Applying the unscented Kalman filter for nonlinear state estimation," *Journal of process control*, vol. 18, no. 7-8, pp. 753–768, 2008.

[53] J. Decuyper, J.-P. Noël, T. De Troyer, M. Runacres, and J. Schoukens, "Polynomial nonlinear state-space modelling of vortex-induced vibrations: black-box vs grey-box approach," in *Proceedings of the International Forum on Aeroelasticity and Structural Dynamics*, 2017.

[54] J.-P. Noël, A. F. Esfahani, G. Kerschen, and J. Schoukens, "A nonlinear state-space approach to hysteresis identification," *Mechanical Systems and Signal Processing*, vol. 84, pp. 171–184, 2017.

[55] J. Schoukens, R. Pintelon, E. Van Der Ouderaa, and J. Renneboog, "Survey of excitation signals for FFT based signal analyzers," *IEEE Transactions on Instrumentation and Measurement*, vol. 37, no. 3, pp. 342–352, 1988.

[56] J. Schoukens, M. Vaes, and R. Pintelon, "Linear system identification in a nonlinear setting: nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation," *IEEE Control Systems Magazine*, vol. 36, no. 3, pp. 38–69, 2016.

[57] R. W. Brockett, "Volterra series and geometric control theory," *Automatica*, vol. 12, no. 2, pp. 167–176, 1976.

[58] A. Khan and N. Vyas, "Application of Volterra and Wiener theories for nonlinear parameter estimation in a rotor-bearing system," *Nonlinear Dynamics*, vol. 24, no. 3, pp. 285–304, 2001.

[59] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Transactions on circuits and systems*, vol. 32, no. 11, pp. 1150–1161, 1985.

[60] N. Wiener, *Nonlinear problems in random theory*. MIT Press, 1966.

[61] J. Barrett, "The use of functionals in the analysis of non-linear physical systems," *International Journal of Electronics*, vol. 15, no. 6, pp. 567–615, 1963.

[62] J.-P. Noël and M. Schoukens, "F-16 aircraft benchmark based on ground vibration test data," in *2017 Workshop on Nonlinear System Identification Benchmarks*, 2017, pp. 19–23.

[63] T. McKelvey, H. Akçay, and L. Ljung, "Subspace-based multivariable system identification from frequency response data," *IEEE Transactions on Automatic Control*, vol. 41, no. 7, pp. 960–979, 1996.

[64] R. Pintelon, "Frequency-domain subspace system identification using non-parametric noise models," *Automatica*, vol. 38, no. 8, pp. 1295–1311, 2002.

[65] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of Applied Mathematics*, no. 2, pp. 164–168, 1944.

[66] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[67] J. Paduart, L. Lauwers, J. Swevers, K. Smolders, J. Schoukens, and R. Pintelon, "Identification of nonlinear systems using polynomial nonlinear state space models," *Automatica*, vol. 46, no. 4, pp. 647–656, 2010.

[68] M. Fliess and D. Normand-Cyrot, "On the approximation of nonlinear systems by some simple state-space models," *IFAC Proceedings Volumes*, vol. 15, no. 4, pp. 511–514, 1982.

[69] I. Kovacic and M. J. Brennan, *The Duffing equation: nonlinear oscillators and their behaviour*. John Wiley & Sons, 2011.

Appendix A

Derivation of the MPC Prediction Matrices

This appendix concerns the derivation of the MPC prediction matrices, which are used to describe the evolution of the states and output over the prediction horizon. Remember the relations $t(0|k) = t(k) \in \mathbb{R}_{\geq 0}$, $\bar{x}(0|k) = \bar{x}(k) \in \mathbb{R}^n$, $\Delta u(0|k) = \Delta u(k) \in \mathbb{R}$ and $\Delta\zeta(y(0|k)) = \Delta\zeta(y(k)) \in \mathbb{R}^s$. Based on the augmented state-space (2.3), the N_p future predictions of the output at time $t(k)$, i.e.,

$$Y_k = \begin{bmatrix} y(1|k) \\ y(2|k) \\ \vdots \\ y(N_p|k) \end{bmatrix}, \tag{A.1}$$

are given by the following relations

$$\begin{cases} \bar{x}(1|k) & = \overline{\boldsymbol{A}}\bar{x}(k) + \overline{\boldsymbol{B}}\Delta u(k) + \overline{\boldsymbol{E}}\Delta\zeta(y(k)) \\ y(1|k) & = \overline{\boldsymbol{C}}\bar{x}(1|k) + \boldsymbol{D}u(1|k) \\ & = \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\bar{x}(k) + \overline{\boldsymbol{C}}\overline{\boldsymbol{B}}\Delta u(k) + \overline{\boldsymbol{C}}\overline{\boldsymbol{E}}\Delta\zeta(y(k)) + \boldsymbol{D}u(1|k), \end{cases}$$

$$\begin{cases} \bar{x}(2|k) & = \overline{\boldsymbol{A}}\bar{x}(1|k) + \overline{\boldsymbol{B}}\Delta u(1|k) + \overline{\boldsymbol{E}}\Delta\zeta(y(1|k)) \\ & = \overline{\boldsymbol{A}}^2\bar{x}(k) + \overline{\boldsymbol{A}}\overline{\boldsymbol{B}}\Delta u(k) + \overline{\boldsymbol{A}}\overline{\boldsymbol{E}}\Delta\zeta(y(k)) + \overline{\boldsymbol{B}}\Delta u(1|k) + \overline{\boldsymbol{E}}\Delta\zeta(y(1|k)) \\ y(2|k) & = \overline{\boldsymbol{C}}\bar{x}(2|k) + \boldsymbol{D}u(2|k) \\ & = \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^2\bar{x}(k) + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}}\Delta u(k) + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{E}}\Delta\zeta(y(k)) + \overline{\boldsymbol{C}}\overline{\boldsymbol{B}}\Delta u(1|k) \\ & \quad + \overline{\boldsymbol{C}}\overline{\boldsymbol{E}}\Delta\zeta(y(1|k)) + \boldsymbol{D}u(2|k), \end{cases}$$

$$\vdots \tag{A.2}$$

$$\begin{cases} \bar{x}(N_p|k) & = \overline{\boldsymbol{A}}\bar{x}(N_p-1|k) + \overline{\boldsymbol{B}}\Delta u(N_p-1|k) + \overline{\boldsymbol{E}}\Delta\zeta(y(N_p-1|k)) \\ & = \overline{\boldsymbol{A}}^{N_p}\bar{x}(k) + \overline{\boldsymbol{A}}^{N_p-1}\overline{\boldsymbol{B}}\Delta u(k) + \overline{\boldsymbol{A}}^{N_p-1}\overline{\boldsymbol{E}}\Delta\zeta(y(k)) \\ & \quad + \ldots + \overline{\boldsymbol{A}}\overline{\boldsymbol{B}}\Delta u(N_p-2|k) + \overline{\boldsymbol{A}}\overline{\boldsymbol{E}}\Delta\zeta(y(N_p-2|k)) + \overline{\boldsymbol{B}}\Delta u(N_p-1|k) \\ & \quad + \overline{\boldsymbol{E}}\Delta\zeta(y(N_p-1|k)) \\ y(N_p|k) & = \overline{\boldsymbol{C}}\bar{x}(N_p|k) + \boldsymbol{D}u(N_p|k) \\ & = \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{N_p}\bar{x}(k) + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{N_p-1}\overline{\boldsymbol{B}}\Delta u(k) + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{N_p-1}\overline{\boldsymbol{E}}\Delta\zeta(y(k)) \\ & \quad + \ldots + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}}\Delta u(N_p-2|k) + \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{E}}\Delta\zeta(y(N_p-2|k)) \\ & \quad + \overline{\boldsymbol{C}}\overline{\boldsymbol{B}}\Delta u(N_p-1|k) + \overline{\boldsymbol{C}}\overline{\boldsymbol{E}}\Delta\zeta(y(N_p-1|k)) + \boldsymbol{D}u(N_p|k), \end{cases}$$

which can be cast into a more compact form as follows

$$\begin{aligned} Y_k = \underbrace{\begin{bmatrix} \overline{\boldsymbol{C}}\overline{\boldsymbol{A}} \\ \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^2 \\ \vdots \\ \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{N_p} \end{bmatrix}}_{S_x} \bar{x}(k) + \underbrace{\begin{bmatrix} \overline{\boldsymbol{C}}\overline{\boldsymbol{B}} & 0 & \cdots & 0 \\ \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}} & \overline{\boldsymbol{C}}\overline{\boldsymbol{B}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{N_p-1}\overline{\boldsymbol{B}} & \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{N_p-2}\overline{\boldsymbol{B}} & \cdots & \overline{\boldsymbol{C}}\overline{\boldsymbol{B}} \end{bmatrix}}_{S_u} \Delta U_k \\ + \underbrace{\begin{bmatrix} \overline{\boldsymbol{C}}\overline{\boldsymbol{E}} & \mathbf{0}_{1\times s} & \cdots & \mathbf{0}_{1\times s} \\ \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{E}} & \overline{\boldsymbol{C}}\overline{\boldsymbol{E}} & \cdots & \mathbf{0}_{1\times s} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{N_p-1}\overline{\boldsymbol{E}} & \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{N_p-2}\overline{\boldsymbol{E}} & \cdots & \overline{\boldsymbol{C}}\overline{\boldsymbol{E}} \end{bmatrix}}_{S_g} \Delta G_k + \boldsymbol{D}U_k, \end{aligned} \tag{A.3}$$

where

$$\Delta U_k = \begin{bmatrix} \Delta u(k) \\ \Delta u(1|k) \\ \vdots \\ \Delta u(N_p-1|k) \end{bmatrix}; \quad \Delta G_k = \begin{bmatrix} \Delta\zeta(y(k)) \\ \Delta\zeta(y(1|k)) \\ \vdots \\ \Delta\zeta(y(N_p-1|k)) \end{bmatrix}; \quad U_k = \begin{bmatrix} u(1|k) \\ u(2|k) \\ \vdots \\ u(N_p|k) \end{bmatrix}. \tag{A.4}$$

Ultimately, ΔU_k is the vector to be optimised while $\bar{x}(k)$ is constructed from output measurements and ΔG_k is known (or estimated, technically) in advance. By means of ΔU_k and the previous plant input it is also possible to construct U_k up to $u(N_p-1|k)$ through the following relations

$$\begin{aligned} u(1|k) &= u(k) + \Delta u(1|k) \\ &= u(-1|k) + \Delta u(k) + \Delta u(1|k), \\ u(2|k) &= u(1|k) + \Delta u(2|k) \\ &= u(-1|k) + \Delta u(k) + \Delta u(1|k) + \Delta u(2|k), \\ &\vdots \\ u(N_p-1|k) &= u(N_p-2|k) + \Delta u(N_p-1|k) \\ &= u(-1|k) + \Delta u(k) + \ldots + \Delta u(N_p-1|k), \end{aligned} \tag{A.5}$$

where $u(-1|k) = u(N_p-1|k-1)$ is the previous plant input. Note that it is not possible to construct the direct feedthrough term for prediction $y(N_p|k)$. Therefore, it is assumed that $u(N_p|k) = u(N_p-1|k)$. Next, U_k is written in compact form as

$$U_k = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} u(-1|k) + \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \Delta U_k = \mathcal{M}u(-1|k) + \mathcal{E}\Delta U_k. \tag{A.6}$$

Finally, by substituting (A.6) in (A.3), the evolution of the output over the prediction horizon N_p at time $t(k)$ can be written as a function of the current state, input sequence, disturbance sequence and previous plant input as

$$Y_k = S_x\bar{x}(k) + (S_u + \boldsymbol{D}\mathcal{E})\Delta U_k + S_g\Delta G_k + \boldsymbol{D}\mathcal{M}u(-1|k). \tag{A.7}$$

The obtained relations can be generalised to any time $t(i|k)$ and for any length of the prediction horizon N_p.

Appendix B

Nonlinear State-Space Parameter Values

This appendix provides the parameter values of the two nonlinear state-space models in table format. Table B.1 corresponds to the simulation experiments of Section 3.2.1, whereas Table B.2 corresponds the model of the experimental setup of Section 4.1.

Parameter	Value	Parameter	Value	Parameter	Value
$\boldsymbol{A}(1,1)$	$9.992 \cdot 10^{-1}$	$\boldsymbol{B}(2,1)$	$2.916 \cdot 10^{-4}$	$\boldsymbol{E}(2,1)$	7.221
$\boldsymbol{A}(2,1)$	$-2.070 \cdot 10^{-2}$	$\boldsymbol{C}(1,1)$	$2.467 \cdot 10^{-3}$	$\boldsymbol{E}(1,2)$	$2.598 \cdot 10^{5}$
$\boldsymbol{A}(1,2)$	$2.428 \cdot 10^{-2}$	$\boldsymbol{C}(1,2)$	$1.854 \cdot 10^{-2}$	$\boldsymbol{E}(2,2)$	$-4.306 \cdot 10^{4}$
$\boldsymbol{A}(2,2)$	$9.994 \cdot 10^{-1}$	$\boldsymbol{D}(1,1)$	$5.190 \cdot 10^{-7}$		
$\boldsymbol{B}(1,1)$	$-2.468 \cdot 10^{-3}$	$\boldsymbol{E}(1,1)$	$1.326 \cdot 10^{2}$		

Table B.1: Discrete-time state-space parameters of the full simulation model ($t_s = 1/1000$ s).

Parameter	Value	Parameter	Value	Parameter	Value
$\boldsymbol{A}(1,1)$	$9.739 \cdot 10^{-1}$	$\boldsymbol{A}(3,3)$	1.001	$\boldsymbol{C}(1,1)$	$-1.297 \cdot 10^{-2}$
$\boldsymbol{A}(2,1)$	$-2.270 \cdot 10^{-1}$	$\boldsymbol{A}(4,3)$	$1.393 \cdot 10^{-2}$	$\boldsymbol{C}(1,2)$	$1.534 \cdot 10^{-3}$
$\boldsymbol{A}(3,1)$	$-4.9715 \cdot 10^{-4}$	$\boldsymbol{A}(1,4)$	$9.026 \cdot 10^{-4}$	$\boldsymbol{C}(1,3)$	$-5.293 \cdot 10^{-2}$
$\boldsymbol{A}(4,1)$	$-7.984 \cdot 10^{-4}$	$\boldsymbol{A}(2,4)$	$-2.339 \cdot 10^{-2}$	$\boldsymbol{C}(1,4)$	$9.680 \cdot 10^{-3}$
$\boldsymbol{A}(1,2)$	$2.421 \cdot 10^{-1}$	$\boldsymbol{A}(3,4)$	$-3.508 \cdot 10^{-2}$	$\boldsymbol{D}(1,1)$	$1.318 \cdot 10^{-4}$
$\boldsymbol{A}(2,2)$	$9.631 \cdot 10^{-1}$	$\boldsymbol{A}(4,4)$	$9.913 \cdot 10^{-1}$	$\boldsymbol{E}(1,1)$	$2.245 \cdot 10^{6}$
$\boldsymbol{A}(3,2)$	$5.329 \cdot 10^{-3}$	$\boldsymbol{B}(1,1)$	$1.287 \cdot 10^{-3}$	$\boldsymbol{E}(2,1)$	$7.039 \cdot 10^{6}$
$\boldsymbol{A}(4,2)$	$-4.458 \cdot 10^{-3}$	$\boldsymbol{B}(2,1)$	$-3.121 \cdot 10^{-3}$	$\boldsymbol{E}(3,1)$	$-2.326 \cdot 10^{6}$
$\boldsymbol{A}(1,3)$	$-7.025 \cdot 10^{-3}$	$\boldsymbol{B}(3,1)$	$1.750 \cdot 10^{-3}$	$\boldsymbol{E}(4,1)$	$-1.112 \cdot 10^{7}$
$\boldsymbol{A}(2,3)$	$-8.304 \cdot 10^{-3}$	$\boldsymbol{B}(4,1)$	$5.684 \cdot 10^{-3}$		

Table B.2: Discrete-time state-space parameters of the experimental setup ($t_s = 1/1024$ s).