

Intelligent Mechatronics through Learning

Citation for published version (APA):

Mooren, N. F. M. (2022). *Intelligent Mechatronics through Learning: from Gaussian Processes to Repetitive Control and Adaptive Feedforward*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mechanical Engineering]. Eindhoven University of Technology.

Document status and date:

Published: 12/05/2022

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

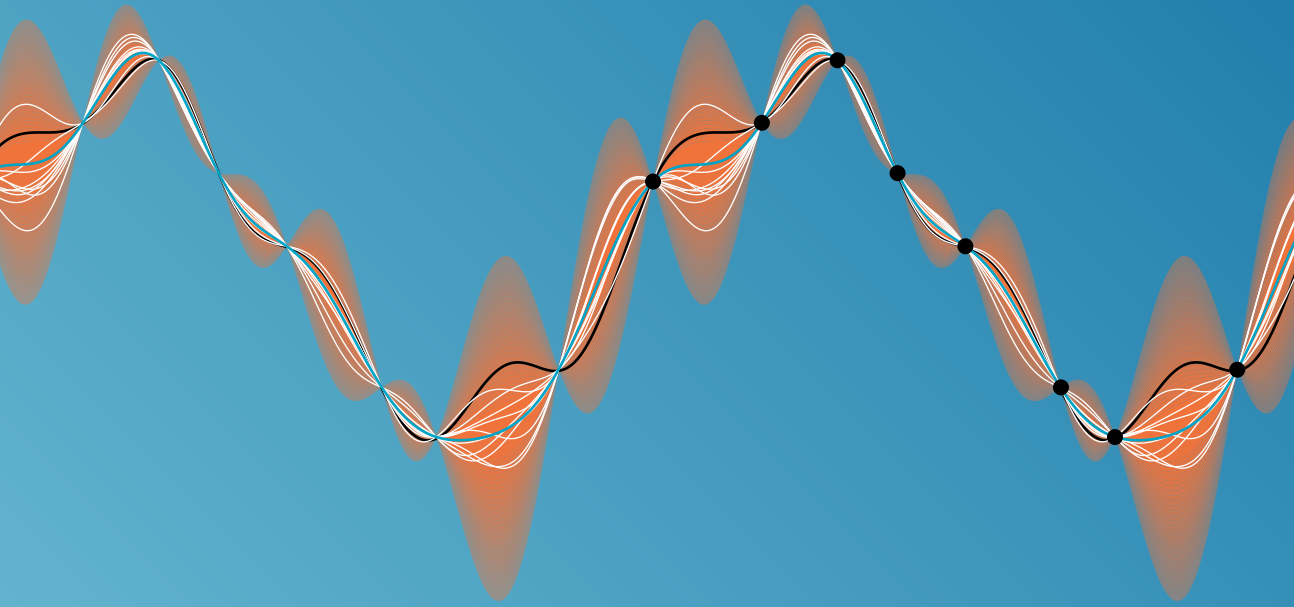
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Intelligent Mechatronics through Learning

from Gaussian Processes to Repetitive Control
and Adaptive Feedforward



Noud Mooren

Intelligent Mechatronics through Learning

from Gaussian Processes to Repetitive Control
and Adaptive Feedforward

Noud Mooren

disc

The author successfully completed the educational program of the Graduate School of the Dutch Institute for Systems and Control (DISC).



This work received funding from the European Union Horizon 2020 program under the project I-MECH under grant agreement 737453.



A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-5497-3

Reproduction: Ipskamp printing, Enschede, the Netherlands.

© 2022 by N.F.M. Mooren. All rights reserved.

Intelligent Mechatronics through Learning from Gaussian Processes to Repetitive Control and Adaptive Feedforward

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een
commissie aangewezen door het College voor
Promoties, in het openbaar te verdedigen
op donderdag 12 mei 2022 om 16:00 uur

door

Noud Franciscus Marie Mooren

geboren te Venlo

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ir. N. van de Wouw
promotor:	prof.dr.ir. T.A.E. Oomen
copromotor:	dr.ir. G. Witvoet
leden:	prof.dr.ir. M. Steinbuch prof.dr. K. Barton (University of Michigan) prof.dr. A. Visioli (University of Brescia) prof.dr. K.G.W. Goossens
adviseur:	dr.ir. G. van der Veen (Nexperia)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Summary

Intelligent Mechatronics through Learning from Gaussian Processes to Repetitive Control and Adaptive Feedforward

Intelligent control methods are essential to keep up with the increasing performance demands of next-generation high-tech systems, ranging from high-speed printing, astronomy, and health-care applications to the semiconductor back-end, while at the same time also pushing for lower costs. Examples include the positioning of semiconductor packages of several micrometers that is performed with tens of thousands of packages per hour at a sub-micrometer positioning accuracy, or laser satellite communication where a pointing accuracy of micro-radians is required with respect to a vibrating satellite at several hundred kilometers distance. To keep on pushing these performance demands in the future, motion controller design is becoming increasingly challenging and the field of learning control is particularly promising. Learning control algorithms, especially in combination with machine learning, enable the development of intelligent controllers that learn automatically from the abundance of available data, enabling an excellent trade-off between accuracy, speed, and cost.

In this thesis the control performance for future systems is improved, i.e., suppression of the influence of both known and unknown exogenous disturbances, by utilizing the vast amount of data that is available and by employing new techniques from the field of machine learning and control. Traditional learning controllers enable a performance improvement up to the reproducible part of the error. However, these are not directly applicable due to the increasing complexity for control of future systems. First, the complexity of unknown disturbances increases such that also multi-physical disturbances are encountered, i.e., disturbances that have multiple generating domains such as the position-, time-, or rotation-angle domain. The rejection of these disturbances requires new disturbance models that are preferably constructed from data. Second, for known disturbances, i.e., reference tracking problems, learning algorithms require repeatable tasks to learn, whereas the flexibility of motion tasks is becoming increasingly important for future systems. Alternatively, adaptive approaches exist to tune feedforward parameters in real-time while also relaxing the

requirement of repeatable tasks, but these often suffer from closed-loop identification problems hampering performance improvement. These challenges have to be addressed to enable successful implementation on a wide range of mechatronic systems.

The main contribution in this thesis aims at the development of systematic design approaches for control of mechatronic systems by learning from data and employing new aspects from machine learning. First, the flexibility and design of internal disturbance models for repetitive control (RC) are significantly improved by employing a Gaussian-process-based buffer, i.e., combining data and prior knowledge to learn a continuous disturbance model from potentially non-equidistant data. It is shown that Gaussian process regression can be employed very efficiently in RC and allows to effortlessly deal with spatially periodic disturbances but also multi-physical disturbances. Second, adaptive feedforward tuning is presented for on-line learning of feedforward parameters from data. This is done by employing an optimal instrumental variable estimator such that unbiased parameter estimates are obtained even during closed-loop operation. The presented approach allows to learn and update feedforward parameters automatically within a split second, which is a major performance improvement compared to manual feedforward parameter tuning and existing task-domain approaches.

The overall results of this thesis contribute to practically relevant and theoretical results that enable the implementation of complex methods originating from machine learning into current state-of-the-art motion control techniques. Moreover, several RC approaches are successfully validated on an industrial substrate carrier and a coarse pointing assembly for laser satellite communication. By employing these recent developments in the field of machine learning together with well-known learning control techniques, there is a large potential to be gained.

Contents

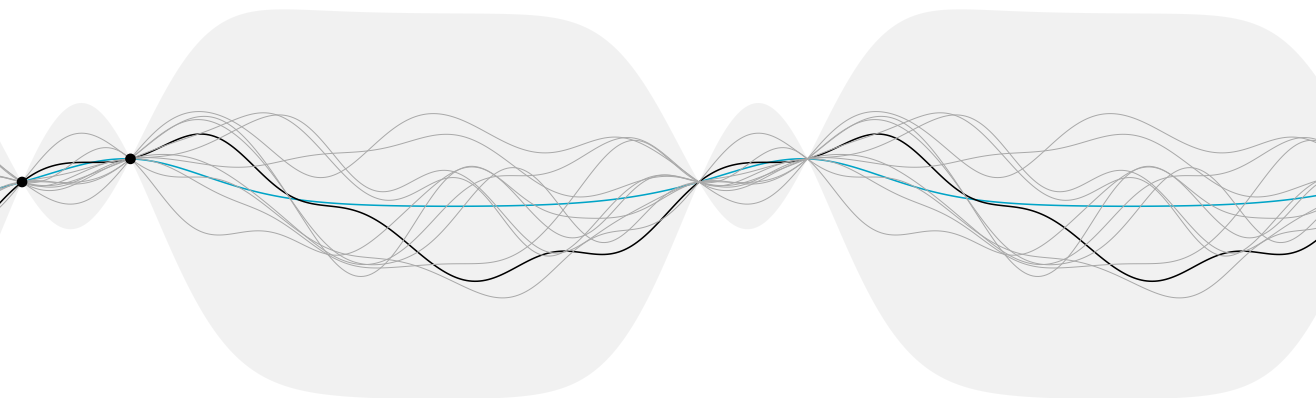
Summary	i
I Introduction	1
1 Learning Control for Intelligent Mechatronics	3
1.1 Impact of future mechatronic systems	3
1.2 Increasing demands for future systems	5
1.3 Current state-of-the-art in control	8
1.4 Towards intelligent mechatronics through learning	13
1.5 Estimation techniques for learning	18
1.6 Contributions and approach	22
1.7 Organization of the thesis	27
II Learning for disturbance rejection	31
2 Gaussian Process Repetitive Control: Beyond Periodic Internal Models through Kernels	33
2.1 Introduction	34
2.2 Problem formulation	36
2.3 Gaussian process buffer in repetitive control	39
2.4 Design methodology for Gaussian process RC	45
2.5 Performance and robustness	49
2.6 Implementation aspects: dealing with initial conditions	58
2.7 Generic case study	59
2.8 Conclusions	64
3 Gaussian Process Repetitive Control with Application to an Industrial Substrate Carrier System with Spatial Disturbances	65
3.1 Introduction	66
3.2 Problem formulation	68
3.3 Gaussian process spatial disturbance model	73

3.4	Spatial Gaussian process repetitive control	77
3.5	Computationally efficient GP-based RC	85
3.6	Simulation case study	87
3.7	Application to a substrate carrier system	93
3.8	Conclusions	99
4	A Gaussian Process Approach to Multiple Internal Models in Repetitive Control	101
4.1	Introduction	102
4.2	Problem formulation	103
4.3	Multi-dimensional repetitive control	107
4.4	Rejecting 2D spatio-temporal disturbances	112
4.5	Conclusion	116
5	Compensating Torque Ripples in a Coarse Pointing Mechanism for Free-Space Optical Communication: A Gaussian Process Repetitive Control Approach	117
5.1	Introduction	118
5.2	Problem definition	120
5.3	CPA disturbance analysis	123
5.4	Gaussian process torque ripple compensation	128
5.5	Application to the CPA	135
5.6	Conclusion	138
5.7	Proofs	139
III	Feedforward tuning for flexible motion tasks	141
6	Feedforward Motion Control: From Batch-to-Batch Learning to On-line Parameter Estimation	143
6.1	Introduction	144
6.2	Towards parameter optimization for feedforward control	145
6.3	Bias analysis in presence of noise	152
6.4	Towards unbiased estimates for feedforward	154
6.5	Illustrative case study	154
6.6	Experimental validation	158
6.7	Conclusions	161
6.8	Proof	161
7	On-line Instrumental Variable-Based Feedforward Tuning for Non-Resetting Motion Tasks	163
7.1	Introduction	164
7.2	Problem definition	167
7.3	On-line IV-based feedforward controller tuning	169

7.4	Design aspects	175
7.5	Wafer-stage case study	177
7.6	Experiments on a benchmark motion system	181
7.7	Conclusions	184
7.8	Proofs	185
IV	Closing	187
8	Conclusions and Recommendations	189
8.1	Conclusions	189
8.2	Recommendations	191
	Bibliography	195
	List of publications	209
	Dankwoord	211
	About the author	213

Part I

Introduction



CHAPTER

1

Learning Control for Intelligent Mechatronics

1.1 Impact of future mechatronic systems

The impact of the manufacturing industry on modern society is rapidly increasing. To continue this innovation there is a strong need for intelligent, high-performance, and low-cost manufacturing systems. This observation is in line with Industry 4.0, which is the current trend of automation and data exchange for innovation in manufacturing technologies (Vaidya et al., 2018). Future manufacturing systems have to be built to support this innovation (Almada-Lobo, 2015). Examples include, satellites to enable communication for the Internet of Things (IoT), cloud computing and the global 5G network (Gregory et al., 2012), and the semiconductor industry (Mack, 2008; Moore, 1995) to supply chips for mobile devices, the automotive industry, and medical equipment. These examples are key motivators to continue the innovation of manufacturing systems, which in turn contributes to the global economy, trade and creates many employment opportunities (Lieder and Rashid, 2016; Santacreu and Zhu, 2018).

Maintaining a good market position in the manufacturing industry requires keeping up with the increasing performance demands, while at the same time aiming for low-cost and large-scale production. Consider the key innovative examples in Figure 1.1, ranging from inexpensive high-volume production of IoT devices in a roll-to-roll print process (a) (Phung et al., 2021), to a vast network of thousands of interconnected satellites for future optical data communication (b) (Saathof et al., 2019), to a prototype multi-agent magnetically levitated platform to parallelize qualification of semiconductor components through atomic force microscopy (AFM) (c) requiring extremely high-accuracy as shown in the scanned AFM image (d) of a mask for the production of integrated circuits (IC) (Holz et al., 2019; Kramer et al., 2019; Sadeghian et al., 2017). These examples

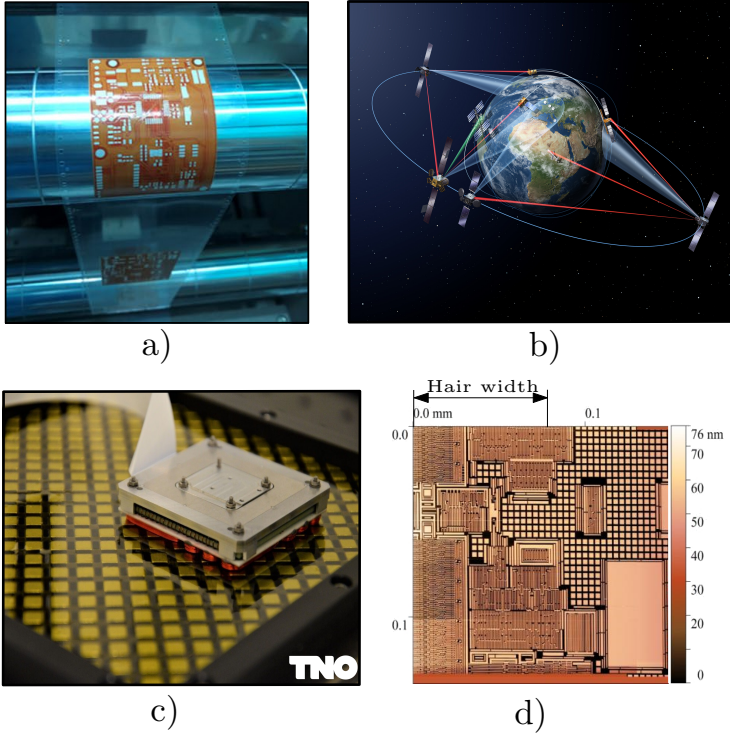


Figure 1.1: Examples of innovative (future) industrial applications: a) printed IoT circuit on the web of a roll-to-roll printing device (source: Phung et al. (2021)); b) a worldwide network for laser satellite communication empowering future high-speed data transfer (source: European Space Agency); c) magnetically levitated platform to carry a sample for atomic force microscopy developed at TNO; and d) an AFM image of a mask for extreme ultra-violet (EUV) lithography (source: Holz et al. (2019)).

illustrate the increasing demands for the manufacturing industry in terms of performance and cost.

To obtain high accuracy and throughput while aiming for inexpensive system designs for future systems requires developing new intelligent control technologies that are sufficiently capable. Bridging the gap between research in the field of control technologies and the current state-of-the-art in the manufacturing industry is key. More specifically, the development of intelligent and user-friendly control approaches in combination with low-cost and efficient control specific hardware (Akesson and Goossens, 2011; Haghi et al., 2020), can have a large impact on the manufacturing industry, by enabling SMEs to grow, innovate and improve sustainability while operational costs and the time-to-market will

decrease and efficiency improves (Čech et al., 2019).

The development of user-friendly control approaches for future manufacturing systems is investigated in this thesis that is a part of the European project I-MECH (Čech et al., 2019). The goal of I-MECH is the development of building blocks that are widely applicable in, e.g., the medical industry and the semiconductor industry. These building blocks are envisioned to embed the latest academic developments, are readily applicable, configurable, and scalable for rapid development of high-performance manufacturing systems. In the following section, the expected demands for future systems and the implications for new control techniques are investigated.

1.2 Increasing demands for future systems

A crucial aspect of future manufacturing systems is the mechatronic positioning system that facilitates fast and accurate positioning of for example high-speed printing of small IoT devices in Figure 1.1. It is expected that the increasing performance demands and the desire for cost-effective systems lead to more system complexity from a control point of view, i.e., achieving high-performance with ‘low-tech’ mechatronic designs requires complex control solutions. This imposes a challenge for the development of new control techniques for future systems, which is illustrated next by investigating the key mechatronic innovations that drive the future applications in Figure 1.1.

1.2.1 Demands for the printing industry

The first example in Figure 1.1 aims at high-speed and low-cost production of IoT devices through printing but is also relevant for, e.g., 3D printing or additive manufacturing (Park et al., 2021). These printing technologies require fast transportation of media, e.g., paper, foil, or glass, up to several meters per second, with an accuracy of a few micrometers. Recently, the generic substrate carrier in Figure 1.2 (a) is developed that attaches media by means of vacuum to a perforated steel transport belt. The steel belt eliminates elasticity and deformations of the substrate that hamper performance in traditional approaches. For accurate positioning of the belt a new type of roller with actuated segments is developed (Beltman et al., 2012a), see Figure 1.2 (a). This illustrates that performance demands lead to additional complexity, i.e., the steel belt and segmented rollers, which are more complex from a control perspective. As a result, disturbances appear that are introduced by imperfections in the belt and rollers which are periodic with one rotation and potentially non-periodic in time (Blanken et al., 2020a).

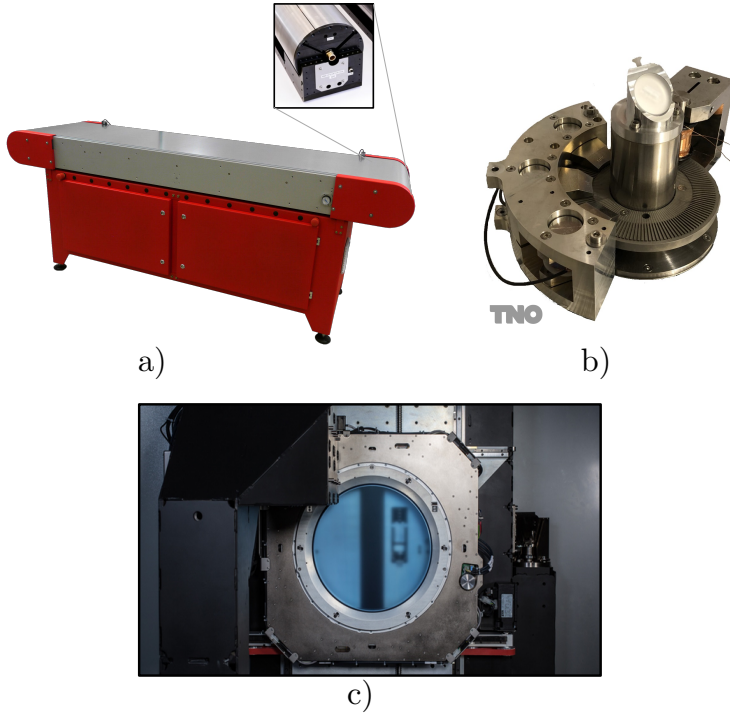


Figure 1.2: Industrial examples of mechatronic systems a) Generic substrate carrier for the printing industry developed at Sioux Technologies, b) coarse pointing assembly that is part of the optical path for laser satellite communication developed at TNO, and c) a wafer-stage setup for the semiconductor industry developed at Nexperia.

1.2.2 Demands for free-space optical communication

The second example in Figure 1.1 illustrates free-space optical communication (FSO) being a key enabling technology for, e.g., scientific earth observations, 5G and 6G connectivity, and the IoT, which rely heavily on high-throughput and secure data communication with worldwide coverage (Saathof et al., 2019). FSO communication requires a transmitter that points a laser beam towards a receiver over a very long distance, up to tens of thousands of kilometers, see Figure 1.1 (b). To generate worldwide coverage, many satellites and ground terminals must be equipped with the coarse pointing assembly in Figure 1.2, which is designed for tracking of the moving laser beam with up to micro-radians positioning accuracy (Kramer et al., 2020). Due to size, weight, and cost being critical aspects for space applications the CPA uses a switched reluctance motor concept that is inexpensive to manufacture but does introduce additional complexity for con-

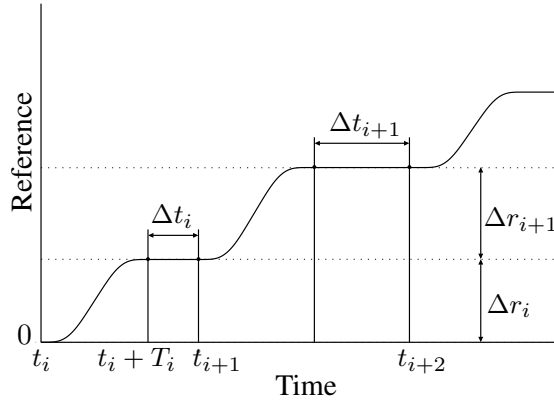


Figure 1.3: Example of a typical point-to-point reference with a varying step-size Δr_i and varying time in-between subsequent tasks Δt_i .

troller design, e.g., in the form of a torque ripple (Miller, 2001; Schmidt et al., 2020). This is yet another example where high-volume and low-cost demands result in systems that require more complexity from a control aspect.

1.2.3 Demands for semiconductor manufacturing

The final example is related to the IC manufacturing process that consists of many steps from exposing a silicon wafer with ultra-violet light, to slicing of the wafer (dicing), and taking ICs from the wafer to a semiconductor package. This latter step is done with the wafer-stage setup in Figure 1.2 (c), which performs this step 72.000 times per hour with several micro-meters positioning accuracy. Control of this setup is challenging; first, the high-throughput and positioning accuracy requires a lighter and stiffer mechanical design that remains to be cost-efficient, leading to more pronounced flexible dynamics. Moreover, the use of inexpensive spindle drives to move the stage introduces additional disturbances. Second, since the pick-and-place process consists of many steps, including, picking up, quality inspection, and positioning, high task flexibility is required, see the typical reference in Figure 1.3, where the step size Δr_i and the time in-between references Δt_i varies arbitrarily. These complexity and flexibility requirements impose a challenge for control techniques.

1.2.4 Resulting control demands for future systems

In summary, the general observation is that future control solutions must cope with cost-effective mechatronic designs while at the same time enabling high accuracy, throughput, and task flexibility. Due to cost and performance considerations, more intelligent control solutions become a favorable means over more

expensive system designs or exotic materials. This leads to more complexity from a control design perspective. More specifically, controllers must cope with, e.g., periodic disturbances (Blanken et al., 2020b; Steinbuch et al., 2007), motor commutations disturbances (Huo et al., 2019), thermal deformations (Evers et al., 2019; Veldman et al., 2019), and manufacturing imperfections that become more pronounced. The next section briefly outlines traditional control approaches which run into their limitations for future systems.

1.3 Current state-of-the-art in control

The performance of mechatronic systems is for a substantial part determined by a well-designed control algorithm in combination with an electro-mechanical design. This section outlines traditional control approaches for mechatronic systems and their limitations, both fundamental and in view of future mechatronic systems.

1.3.1 Control in mechatronic systems

Traditional mechatronic systems consist of a mechanical design, sensors, actuators (Schmidt et al., 2020), and a motion control algorithm that runs on a dedicated hardware platform (Valencia et al., 2015). The control algorithm reads sensor data, e.g., a position or temperature, to provide a control signal to the actuators, e.g., a force or torque, such that a degree of performance is obtained. This cycle is typically performed several thousand times per second. In industrial systems, controllers are implemented in discrete time by sampling the sensor data and providing a command at each sample.

1.3.2 Traditional motion control

The traditional control setting is schematically depicted in Figure 1.4, where the system P is to be controlled that consists of sensors, actuators, and system dynamics. The system is assumed approximately linear time-invariant (LTI) by design, however, due to cost-effective mechatronic designs, there are additional unknown disturbances d acting on the system P , e.g., repeating disturbances or a torque ripple, as shown in the examples in Section 1.2. The system output y contains the sensor measurements, e.g., a position or temperature, and the input u is the control signal, e.g., a force, voltage, or torque. The signal r is the reference to be tracked that is assumed to be known and η is assumed to be random, e.g., sensor noise or quantization errors.

The controller typically consists of a feedback controller C_{fb} that uses sensor data y , and the feedforward controller C_{ff} that only uses the reference signal r . The control goal is to attenuate the influence of the external disturbances d , the

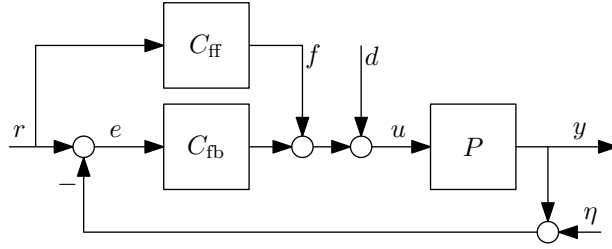


Figure 1.4: Traditional control setup.

reference r , and measurement noise η on the measured error, which is given by

$$e = \underbrace{S(1 - PC_{\text{ff}})}_{\text{Feedforward}} r - \underbrace{SP}_{\text{Feedback}} d - \underbrace{S}_{\text{Feedback}} \eta, \quad (1.1)$$

where $S = (1 + PC_{\text{fb}})^{-1}$ is the sensitivity function¹ (Franklin et al., 2015; Steinbuch and Norg, 1998; Visioli, 2006). If the feedforward controller is designed as $C_{\text{ff}} = P^{-1}$ then errors induced by the reference will be compensated. Moreover, if the feedback controller is designed such that $S = 0$ then the total error becomes zero. Hence, the main role of the feedforward controller is to compensate for the known reference and the feedback controller compensates for unknown disturbances. However, the ideal situation where $C_{\text{ff}} = P^{-1}$ and $S = 0$ cannot be achieved due to theoretical and practical constraints as outlined next. The aim of this thesis is to relax some of these constraints.

1.3.3 Feedback control

A traditional design approach for the feedback control and its limitations in view of future systems are briefly addressed.

Traditional feedback control: The feedback controller is ideally designed such that $S \ll 1$ in (1.1). Design can be done in the time domain, e.g., with a proportional-integral-derivative (PID) controller, by performance criteria such as overshoot, settling time and steady-state error (Visioli, 2006). More “sophisticated” is tuning in the frequency domain by creating a large controller gain $C_{\text{fb}} \gg 1$ such that $S \ll 1$ at the frequencies where a disturbance is present. The basic principle of generating a large controller gain is referred to as high-gain feedback. An important constraint for all feedback approaches is causality, which

¹For the ease of notation in (1.1) there is a slight abuse of notation, i.e., it concerns linear systems which may contain convolutions or multiplications in the time-domain, frequency-domain, Laplace-domain or Z-domain that are intentionally not denoted explicitly. In the remaining chapters, the notation is defined in detail.

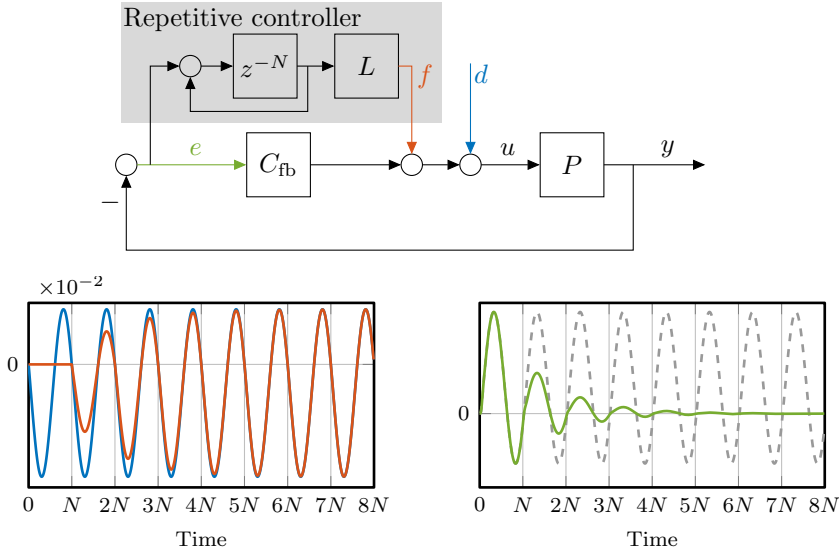


Figure 1.5: Add-on type repetitive controller in the traditional feedback loop for suppression of the unknown periodic disturbance d (top). The disturbance d (—), that is periodic with N samples, is learned by the RC output, see minus f in (—), that converges to the unknown disturbances (bottom left). The periodic part of the error (—) asymptotically converges to zero, compared to not using RC (—) where the periodic component remains (bottom right).

imposes a restriction on minimizing S known as the waterbed effect² (Skogestad and Postlethwaite, 2007), such that $S = 0$ for all frequencies is not possible. This implies that rejection of certain disturbances comes at the cost of amplification of other signals

Repetitive control for repeating disturbances: The unknown disturbance d and the error e often exhibit structure, e.g., repeating in the time domain. Repetitive control (RC) is an add-on type controller (Hara et al., 1988; Longman, 2010) that enables learning of an unknown periodic disturbance from data. A typical RC is schematically depicted in Figure 1.5, it can asymptotically reject the repeating part of the error. To reject the unknown periodic disturbance, RC relies on the internal model principle³ (Francis and Wonham, 1976) by learning

²The waterbed effect describes the constraint that reducing the sensitivity function $|S(e^{j\omega})| < 1$ for some frequencies ω directly implies that $|S(e^{j\omega})| > 1$ for other frequencies.

³According to the internal model principle, a model of the disturbance generating system must be present in a stable feedback loop for asymptotic rejection.

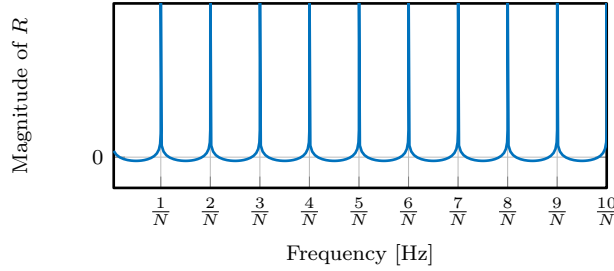


Figure 1.6: Example of the magnitude $|R(z = e^{-j2\pi f})|$ as a function of the frequency f Hertz. The RC has infinite gain at the fundamental frequency $f = \frac{1}{N}$ Hertz and its harmonics.

a model of the disturbance d that has time-domain period N ,

$$d(k + N) = d(k), \quad (1.2)$$

in a time-domain memory with a learning filter L . The memory is constructed from an N -sample delay z^{-N} in a positive feedback loop, the resulting feedback error is given by

$$e = - \underbrace{PS}_{\text{feedback}} \underbrace{(I + PSR)^{-1}}_{\text{RC}} d, \quad \text{with} \quad R = \frac{Lz^{-N}}{1 - z^{-N}}, \quad (1.3)$$

which shows that R gives an additional tuning knob N for disturbances d with the periodic property (1.2). Hence, the repetitive controller R is LTI filter that exploits time-domain periodicity which has the same limitations as the feedback controller C_{fb} . Essentially, RC generates high-gain feedback, similar to C_{fb} at the fundamental frequency $1/N$ and all of its harmonics as shown in Figure 1.6.

An illustrative example of RC is shown in Figure 1.5 where during the first period the RC buffer is filled with error data and the output f remains zero. In the consequent periods, the buffer content is filtered with L and injected in the feedback loop to compensate for the disturbance. By appropriate design of N and L this leads to asymptotic rejection of disturbances with period N , see Figure 1.5.

A crucial observation is that repetitive control can solely compensate for time-periodic disturbances. This is rather restrictive for future systems as the substrate carrier in Figure 1.2, where disturbances are expected that repeat each roller rotation instead of the time domain. Due to the waterbed effect, RC may even amplify such time-domain non-periodic disturbances, and it is yet unable to exploit the underlying periodicity in other domains.

1.3.4 Feedforward control

Feedforward control for mechatronic systems can yield a substantial performance improvement, with respect to feedback control, since the reference is known in advance. A traditional design approach and the limitation of feedforward are outlined next.

Model-based feedforward: It follows from (1.1) that for attenuation of the reference, the feedforward controller ideally is a perfect model of the system inverse $C_{\text{ff}} = \hat{P}^{-1}$. The model \hat{P}^{-1} can be obtained with either physical modeling or through experimental identification based on data. Typically, both approaches will yield an approximate model $\hat{P}^{-1} \approx P^{-1}$, which directly limits the performance (Devasia, 2002). Alternatively, the feedforward controller can be parameterized as $C_{\text{ff}}(\theta)$ with feedforward parameters θ that can be tuned to approximate the system inverse. In practice, mechatronic systems are often designed to exhibit dominant rigid-body behavior such that $y = \frac{1}{ms^2}u$ and the ideal feedforward is given by $f = m\ddot{r}$ also known as acceleration feedforward (Boerlage et al., 2003).

Future system dynamics are becoming more complex, potentially including position dependence and manufacturing imperfections. In this situation, obtaining a good system inverse model or tuning of feedforward parameters has to be performed on an inefficient machine-specific basis and becomes more time-consuming.

Learning feedforward: Feedforward control can benefit from learning from the abundance of data that is available and exploiting repetitive motion tasks. This can potentially result in extreme performance improvement even if the case of a non-perfect model (Devasia, 2002; Oomen, 2018). In iterative learning control (ILC) (Arimoto et al., 1984; Bristow et al., 2006) the repetitiveness of motion tasks is exploited to learn a compensation signal from error data that compensates for the exact same tasks. These learning approaches can significantly improve feedforward controller design by removing manual tuning. However, learning approaches require that motion tasks are exactly repeating and that each task starts from the same initial condition, which is quite restrictive in view of the applications in Figure 1.2.

To conclude, the traditional approaches for feedforward and feedback control, including learning extensions that exploit repetitiveness of tasks or disturbances, are not directly applicable to future mechatronic systems, this is further investigated in the following section.

1.4 Towards intelligent mechatronics through learning

Learning from data can improve performance and the design of intelligent control algorithms for future systems, examples include, Altın et al. (2017); Blanken et al. (2020a); Bolder et al. (2017); Reinders et al. (2020). This section investigates what requirements appear for future intelligent control approaches. Subsequently, the challenges that are investigated in this thesis are presented.

1.4.1 Challenges for learning control in future systems

Although learning approaches such as RC and ILC are highly promising, the complexity that appears for control of future mechatronic systems requires new intelligent control techniques. The aim of this thesis is to exploit learning from data to improve disturbance attenuation and reference tracking for future systems while at the same aiming for user-friendly design. The expected requirements for control are outlined next, after the following motivating examples inspired by the systems in Figure 1.2.

First, due to the increasing complexity it is expected that systems will have disturbances with complex underlying structures. More specifically, these disturbances can be repeating in other domains than the time domain, e.g., as in the substrate carrier where a disturbance repeats every roller or belt rotation. Consider the following motivational example.

Motivating example: Spatially periodic disturbance

Spatially periodic disturbances appear in the substrate carrier system in Figure 1.2 induced by roller imperfections and the segments in the rollers. This case is simulated in the simplified setting in Figure 1.7 of a roller with an eccentricity.

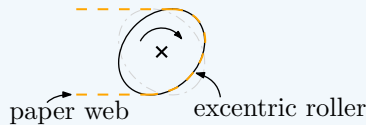


Figure 1.7: Imperfect roller with an excentricity introducing a disturbance that repeats every roller roation.

If the roller position increases linearly (—) as in Figure 1.8 (top left), then the disturbance is periodic in the time-domain (top right). If the velocity of the roller varies (—) (bottom left), then the resulting disturbance is non-periodic in the time domain (—) (bottom right).

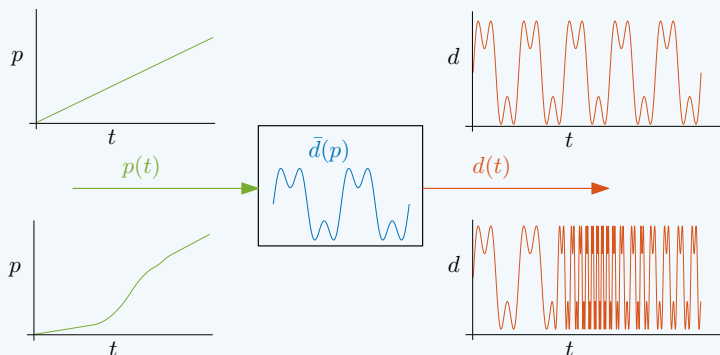


Figure 1.8: Spatial-periodic disturbance $\bar{d}(p)$ (—) (center) and the resulting time-domain disturbances $d(t)$ (—) for a constant (top) and varying (bottom) roller velocity (—).

This example shows that a spatially periodic disturbance, such as the segmented roller, leads to a disturbance that is non-periodic in the time-domain whereas the underlying spatial disturbance is periodic. Moreover, situations may appear where disturbances originate from multiple underlying causes, e.g., the magnetically levitated platform in Figure 1.1 as shown in the second motivating example.

Motivating example: Multi-dimensional disturbance

The magnetically levitated platform in Figure 1.9 is subject to a 2-dimensional commutation disturbance induced by the permanent magnet array, this disturbance repeats in the x and y position and the period is the magnet pitch (Kramer et al., 2019; Van den Braembussche et al., 1998). A simplified version of this disturbance is shown on the right in Figure 1.10.

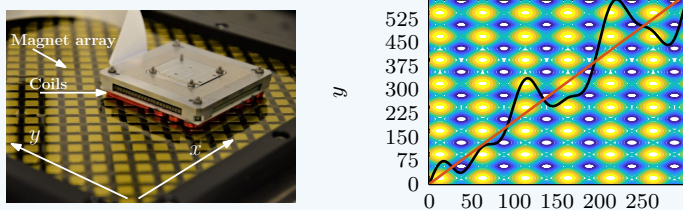


Figure 1.9: Magnetically levitated platform and an illustration of the disturbance on the platform in the (x, y) plane due to commutation.

Two example trajectories of the platform are given in (—) which is a

straight line and (—) that varies. The resulting disturbances are given in the bottom plot as a function of time. It appears that (—) is periodic whereas (—) is non-periodic in time, while the underlying 2-dimensional spatial disturbance is periodic.

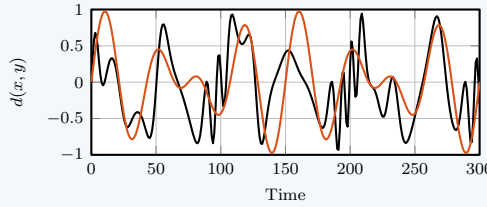


Figure 1.10: The commutation disturbance for two trajectories (—) and (—) of the platform as a function of time.

Hence, the type of disturbances that appear in these examples cannot be attenuated with traditional approaches such as RC, see Section 1.3.3. This imposes a major opportunity for controller design to exploit the underlying periodicity by learning from data while the time-domain signal is non-periodic.

Second, the complexity of control for mechatronic systems is rapidly increasing, i.e., more dynamic modes and flexible behavior, slight variations over systems due to imperfections, and position dependence are encountered. At the same time, the flexibility of motion tasks is increasing as in the wafer-stage application in Figure 1.3, where reference tasks are not exactly repeating nor resetting. This leads to a situation where traditional approaches for tuning of feedforward controllers are cumbersome, e.g., due to variations between systems tuning has to be performed on an inefficient machine-specific basis, and the modeling complexity significantly increases for model-based feedforward. Moreover, learning approaches such as ILC require repeating tasks, which is too restrictive for future systems.

These challenges for the rejection of unknown disturbances with complex underlying structures as well as feedforward controller tuning for tracking of varying reference with complex systems dynamics, are translated to generic requirement for control. The aim is to utilize the vast amount of data that is available and incorporate structure in multiple domains, e.g., time, position, or commutation angle domain, to improve control performance and automate tuning for future systems.

1.4.2 Control requirements for intelligent mechatronics

Inspired by the applications in Figure 1.1 and given the previous motivating examples, it is expected that intelligent controllers for future mechatronic systems in general need to comply with the following high-level requirements.

R1: Rejection of complex disturbances;

R1.1 that are periodic or multi-periodic, i.e., a sum of multiple-periodic signals, in the time domain;

R1.2 that can be non-periodic in the time-domain but are periodic in, e.g., the commutation-angle domain or roller-position domain;

R1.3 can be periodic in multiple uncorrelated domains at the same time; and

R1.4 with a computationally attractive approach for industrial applications.

R2: Feedforward tuning for flexible motion tasks;

R2.1 for systems with complex dynamics, e.g., pronounced flexible dynamic behavior, system to system variations;

R2.2 with a user friendly or automatic tuning approach; and

R2.3 to enable fast performance improvement, e.g., learning within a task, for a wide variety of motion tasks.

1.4.3 Research challenges

The challenges for intelligent control of future mechatronic systems are outlined in view of the requirements R1 for disturbance rejection and R2 for reference tracking.

Generic internal models for disturbance rejection: Repetitive control is a promising approach that incorporates prior disturbance knowledge, i.e., time-domain periodicity, to estimate the unknown periodic disturbance from error data (Steinbuch et al., 2007). As a response to the increasing complexity of these disturbance structures, a variety of extensions have been developed for RC. Robust RC is developed for uncertain period times or noise (Chen and Tomizuka, 2013; Pipeleers et al., 2008; Steinbuch et al., 2007), adaptive RC that can cope with period-time variations (Chen and Tomizuka, 2011; Hillerstrom, 1996; Kurniawan et al., 2014), RC for spatially periodic disturbances (Chen and Yang, 2007; Huo et al., 2019), and RC for multi-period disturbances (Blanken et al., 2020a). These approaches make modifications to the traditional RC internal model tailored towards a specific goal. Despite these extensions, a generic internal model for RC that can elegantly cope with the type of disturbances in R1, i.e., periodicity in on or multiple underlying domains, is not yet developed. This imposes the need for a generic and intelligent internal model for RC which constitutes the first research challenge in this thesis.

Research challenge R1: Develop a generic internal model that exploits underlying periodic structures of disturbances, e.g., in the commutation-angle domain, position domain or multiple-domain, to compensate for disturbances that are repeating but not necessarily in time.

Feedforward tuning for flexible motion tasks: High task flexibility and performance in addition to user-friendly or automatic tuning are required for feedforward control of future systems (R2). The task flexibility and performance trade-off among different existing feedforward approaches is visually shown in Figure 1.11 and outlined next. Model-based feedforward (○) is applicable to a wide range of references, and the performance is directly related to the inverse modeling (Van Zundert and Oomen, 2018) quality. Due to the increasing model complexity (R2.1) the initial modeling effort or tuning step is becoming increasingly difficult and time-consuming for future complex systems which hampers performance.

Alternatively, ILC (✕) is a non-parametric approach (■) that exploits repeating tasks to learn a feedforward signal from task-to-task (■) that compensates for the exact same reference (Altin et al., 2017; Blanken and Oomen, 2019; Boeren et al., 2016; Bristow et al., 2006; de Rozario et al., 2019). This approach can yield extreme performance by learning from previous tasks, but has low task flexibility. To improve task flexibility for ILC the feedforward signal can be parameterized (■) as $f = C_{ff}(\theta)r$ such that it adapts when the reference changes (◆) (Bolder and Oomen, 2015; Hoelzle et al., 2011; van de Wijdeven and Bosgra, 2010). If tasks are resetting, then ILC can be used to learn the parameters θ for each subsequent task, i.e., updating from task to task. The same can be done by directly parameterizing the inverse system model and learning the coefficients from task to task (Blanken and Oomen, 2020; Boeren et al., 2018).

Batch-wise approaches update feedforward parameters on a task-to-task basis and rely on the resetting behavior of motion tasks, i.e., it is required that the system resets to the same initial conditions at the start of each task. In contrast, manually tuned model-based feedforward can be applied to any reference at the cost of moderate performance.

The aim is to combine the benefits of model-based feedforward (high task flexibility) and learning from data (high performance) in view of R2. Also enable fast performance improvement (R2.3), e.g., by learning within a task, continuous updating as in adaptive feedforward control is highly promising (□) in Figure 1.11. Adaptive updating is expected to enable learning for a wide range of non-resetting references and complex system dynamics, which constitutes the second research challenge of this thesis.

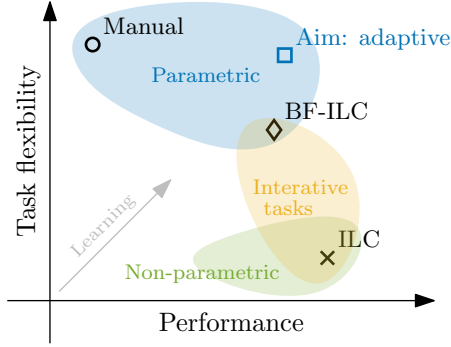


Figure 1.11: Task flexibility and performance trade-offs in a variety of common feedforward approaches, ranging from non-parametric (■) approaches, to approaches that exploit resetting tasks (■), and parametric approaches that allow for a wider range of (resetting) tasks (■).

Research challenge R2: Develop a framework for automatic on-line estimation of feedforward controller parameters using data for immediate performance improvement for a wide variety of potentially non-resetting and varying references.

1.5 Estimation techniques for learning

In this section, basic fundamentals from the field of parameter estimation and machine learning that are required for the contributions in this thesis are outlined, after which the contributions are stated.

1.5.1 Parametric estimation in control

In view of requirement R2, the feedforward controller is preferably parameterized for flexibility and it is required to estimate the parameters from data. A simple way of parametric estimation that is used in control, e.g., for feedforward (Boeren et al., 2016; Bolder and Oomen, 2015; Fujimoto and Yao, 2005) and identification (Ljung, 1999) is outlined next.

Consider a given measured signal y that contains observations of an unknown function or system $f(x)$,

$$y = f(x) + \text{noise} \quad \text{with} \quad f(x) = \sum_{i=1}^n \varphi_i^\top(x) \theta_i^0 = \phi(x)^\top \theta_0, \quad (1.4)$$

parameterized with the basis functions $\phi(x) = [\varphi_1(x) \ \varphi_2(x) \ \dots \ \varphi_n(x)]^\top \in \mathbb{R}^n$. Then, given a set of N measurement points $Y \in \mathbb{R}^N$ and inputs $X \in \mathbb{R}^N$ an estimate of the true unknown parameters $\theta_0 \in \mathbb{R}^n$ can be obtained with the following optimization problem that minimizes the difference between the approximated model and the data

$$\hat{\theta} = \min_{\theta} \|Y - \Phi\theta\|^2, \quad (1.5)$$

with $\Phi = [\phi^\top(1) \ \dots \ \phi^\top(N)]^\top \in \mathbb{R}^{N \times n}$ which is a linear least-squares problem, with analytic solution

$$\hat{\theta} = (\Phi^\top \Phi)^{-1} \Phi^\top Y, \quad (1.6)$$

that can also be computed recursively for on-line estimation (Åström and Wittenmark, 2013), and in (Butler, 2012) for on-line parameter estimation for feed-forward. Moreover, under specific conditions on Φ and Y , the estimator (1.5) recovers the true unknown parameters $\theta \rightarrow \theta_0$ as $N \rightarrow \infty$ (Söderström and Stoica, 1989).

Example: Least squares regression

Suppose a true static function, (—) in Figure 1.12, given by

$$f(x) = 5 \sin(x) + 4 \cos(3x) - 2 \sin(6x) - \sin(9x),$$

that is a sum of $n = 4$ sinusoids with unknown true parameters $\theta_0 = [5, 4, -2, -1]^\top$ from which $N = 14$ samples are available. An estimate is computed with a perfect model (—) and with only two basis functions

$$\Phi = [\sin(x) \ \cos(3x)],$$

in (—) in Figure 1.12. As a result (—) does not fully recover the true function due to imperfect model knowledge, whereas (---) does.

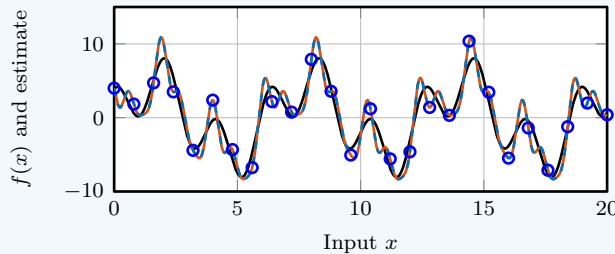


Figure 1.12: Least squares regression example.

This illustrative example is a simple case of LS regression to estimate a function parameters from data, which is also applicable to estimate system parameters, see (Boeren et al., 2014) as further exploited in this thesis. The advantage of LS is that the problem has an analytic solution, however, the choice of basis function directly influences the model quality. Next, an alternative non-parametric estimation technique is presented that does not require explicit knowledge of the individual basis functions.

1.5.2 Non-parametric Gaussian process estimation

Estimation of unknown disturbances with underlying periodic structures from data is required in view of R1. Next, a non-parametric estimator is presented that enables estimation of a continuous function from data by incorporating structure with high-level properties such as smoothness and periodicity, instead of explicit basis functions as in the previous section.

Machine learning and in particular Gaussian process (GP) regression is suitable to estimate and predict unknown functions or models based on large amounts of data by introducing prior knowledge. With the available data and the efficiency of hardware implementations, GPs are gaining more attention in control (Schölkopf et al., 2002) recently, e.g., for identification of linear (Lataire and Chen, 2016; Pillonetto and De Nicolao, 2010) and non-linear systems (Mazzoleni et al., 2020), modeling of non-causal inverses for feedforward control (Blanken and Oomen, 2020), and for function estimation in iterative learning control (Poot et al., 2021). In this thesis, Gaussian processes play an important role to learn unknown disturbances from data and periodic prior knowledge to make RC more generally applicable.

A Gaussian process (GP) is a distribution over functions denoted by

$$f(x) \sim \mathcal{GP}(m(x), \kappa(x, x')), \quad (1.7)$$

that is completely determined by its mean function and covariance function, given by

$$m(x) = \mathbb{E}[f(x)] \quad (1.8)$$

$$\kappa(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))], \quad (1.9)$$

respectively, see, e.g., Murphy (2012); Williams and Rasmussen (2006). The mean and covariance functions specify prior knowledge about the unknown stochastic function $f(x)$, e.g., smoothness or periodicity. It can be shown that some covariance functions, referred to as non-degenerate, require an infinite sum of basis functions to be exactly recovered, see Mercer's theorem (Williams and Rasmussen, 2006, §4.3). GP regression allows to define a potentially infinite set of basis functions through high-level properties such as periodicity with the prior distribution (1.7) instead of individual basis functions.

GP regression is performed with (noisy) data (x, y) of an unknown function $f(x)$, i.e.,

$$y = f(x) + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2), \quad (1.10)$$

and a *prior distribution* (1.7), to compute a *posterior distribution* at an arbitrary test point x^* . This is possible by assuming that the data (x, y) and any test point x^* have a joint Gaussian distribution

$$\begin{bmatrix} y \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{xx} + \sigma_n^2 I_N & K_{xx^*} \\ K_{xx^*}^\top & K_{x^*x^*} \end{bmatrix}\right), \quad (1.11)$$

where K_{ab} is the covariance function κ evaluated at all combinations (a, b) and referred to as a kernel matrix. The posterior mean and variance are given by

$$\mu(x^*) = K_{xx^*}^\top (K_{xx} + \sigma_n^2 I_N)^{-1} y \quad (1.12)$$

$$\Sigma(x^*) = K_{x^*x^*} - K_{x^*x}^\top (K_{xx} + \sigma_n^2 I_N)^{-1} K_{xx^*} \quad (1.13)$$

which is continuous in x^* . GP regression is graphically illustrated by means of the following example.

Example: Gaussian process regression

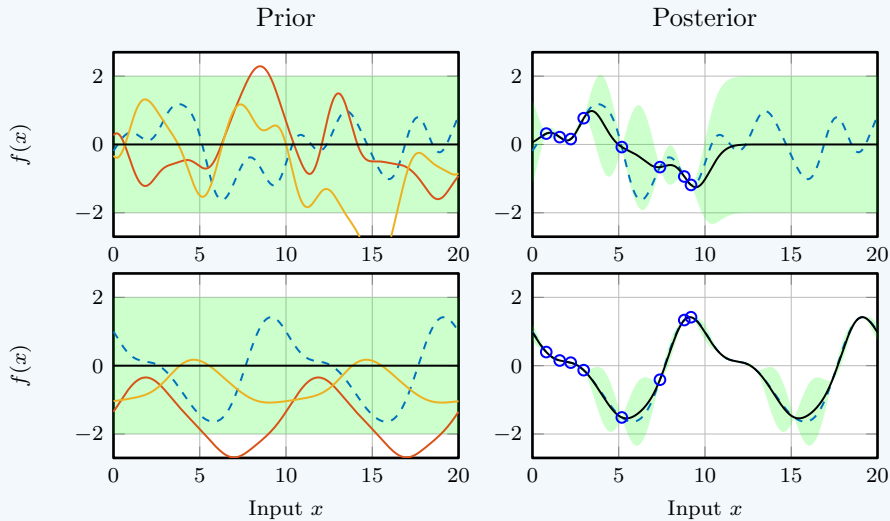


Figure 1.13: Random samples (—) and (—) from the prior distribution (left), and the posterior mean (—) and variance (■) (right). Obtained with Gaussian process regression based on the data (○) and true function (---).

Figure 1.13 shows random samples (—) and (—) taken from two different zero-mean prior distributions, the first one (top left) contains samples from smooth prior and the second one (bottom left) shows samples from smooth and periodic prior with period 10 [-]. These samples comply with the prior, i.e., they are smooth and with periodicity respectively. GP regression is performed with 8 random samples from the true function (--) and the prior distributions. The resulting posterior mean (—) in the figure on the right is a continuous function that interpolates in-between the data points, the variance (■) provides a confidence interval. With the periodic prior the data is extrapolated periodically beyond the first period.

The example shows that the posterior mean is a continuous function and the variance provides a confidence interval. Moreover, the choice of a covariance function has a large influence on the posterior, i.e., introducing periodicity can yield a much better posterior estimate compared to only including smoothness, in the case that the true function is indeed periodic.

In summary, GP regression is a promising approach that enables learning an unknown function from data if the explicit basis functions are unknown but high-level properties are known. Moreover, LS regression is suitable for on-line estimation due to its analytic solution and efficient computation. The use of these estimators in control is further investigated in this thesis.

1.6 Contributions and approach

In this section, the contributions in this thesis are outlined. These are split into *learning for disturbance rejection* and *feedforward tuning for flexible motion tasks* in line with the requirements R1 and R2 in Section 1.4.2.

1.6.1 Contribution I: Learning for disturbance rejection

Repetitive control can result in asymptotic rejection of disturbances that are unknown but periodic in the time-domain with a known period, see Section 1.3.3. As the complexity of systems is increasing, so is the underlying structure of disturbances. This could lead to periodicity in position rather than time or in multiple underlying domains, as shown in previous motivating examples. The internal models for RC are currently not generic enough to compensate for these disturbances because they only consider time-domain periodicity, whereas the principle behind RC is highly suitable for a wider range of disturbances. Hence, the first main contribution of this thesis is to develop an intelligent internal model for RC that allows for including more generic structures or prior knowledge.

Generic GP-based internal model for RC: Applying traditional temporal RC to the disturbances in requirement R1 can lead to slow learning in the case that multi-period disturbances appear, or even a deterioration of performance if the disturbance is non-periodic in time (Blanken et al., 2020a). To address these limitations in RC, several extensions of RC are developed, however, there is no generic approach for these situations. Motivated by the lack of a generic approach, the aim is to introduce a new internal model for RC that can cope with a wide variety of (non-standard) situations, including, multi-period disturbances, disturbances with a period that is not an integer number of samples, and situations that require noise or period time robustness. This leads to a new, user-friendly, and systematic design framework for RC that can recover several existing approaches. The recent developments of GP regression in estimation and control are very promising to estimate continuous function from data and prior. The application of GP as an internal model in RC is not yet explored. This leads to the first contribution in this thesis.

Contribution C1: A generic internal model for temporal RC to attenuate a wide range of disturbance, by employing Gaussian processes to incorporate prior knowledge resulting in a user-friendly and systematic design approach.

Compensating spatially periodic disturbances: Due to the repetitive nature of systems or system components, e.g., rotating electro motors or belts and rollers, disturbances appear that are fundamentally repeating in the position-domain rather than in the time domain. This may lead to a situation where the disturbance is non-periodic in time whereas it remains periodic in position as outlined in earlier motivating examples. Compensation of spatially periodic disturbances through RC that are non-periodic in time requires a new type of internal model. Moreover, due to equidistant sampling in the time-domain, the disturbances become non-equidistantly sampled in the position domain due to velocity variations. This implies that for compensation of spatial disturbances the traditional RC buffer, which consists of discrete values, must be replaced by a continuous spatially periodic function based on non-equidistant data.

Existing approaches appeared where either the memory-loop size or the sample time are adaptively changed, however, there is no systematic approach for interpolation and extrapolation due to the non-equidistant spatial data. The presented GP-based internal model for RC in C1 can be extended systematically to also embed spatially periodic disturbances. The GP-based internal disturbance model can be constructed in the position-domain by using smooth and periodic prior knowledge resulting in a continuous function. This constitutes the second contribution of this thesis.

Contribution C2: A spatially periodic Gaussian process based internal model for RC that can learn a continuous spatial disturbance model from non-equidistant observations for rejection of spatial disturbances that are potentially non-periodic in time.

This spatial GP-based RC approach is experimentally validated on the substrate carrier system in Figure 1.2.

Compensating multi-dimensional disturbances: Contributions C1 and C2 focus on new internal models for temporal periodic and spatial periodic disturbances respectively. In line with requirement R1.3, it may appear that a disturbance is repeating in multiple domains such as in the motivating example in Section 1.4.2. Compensation of these multi-dimensional disturbances can be done through RC by exploiting the periodicity in all of the underlying domains in a multi-dimensional internal model. A useful property of the covariance functions for GP regression is that new covariance functions can be constructed by combining multiple covariance functions, e.g., through multiplication or addition. This allows defining multi-dimensional covariance functions with structures in multiple domains which can be exploited to reject multi-dimensional disturbances through RC. This constitutes the third contribution of this thesis.

Contribution C3: A framework for the rejection of multi-dimensional disturbance through a multi-dimensional GP-based buffer design for RC, that exploits the underlying periodicities in each of the domains.

Efficient GP-based learning for torque ripple compensation: The coarse pointing assembly for FSO communication in Figure 1.2 is an example where a torque ripple disturbance appears due to commutation. The torque ripple disturbance is periodic in the commutation-angle domain but is non-periodic in time if the velocity varies. Hence, compensation with traditional RC is not possible. Since the torque ripple is caused by imperfect commutation, which is static, a static compensation function in the commutation-angle domain is sufficient for compensation at arbitrary velocity. This motivates the fourth contribution of this thesis, on how to automatically learn a spatially periodic static compensation function from data for torque ripple compensation at arbitrary velocity. This is possible by using RC (C1) at constant velocity and consequently learning a spatial GP-based function for compensation at arbitrary velocity. Moreover, this also allows avoiding the relatively high computational demand of spatial GP-based RC (C2), which is favorable for space applications. This constitutes the fourth contribution of this thesis.

Table 1.1: Overview of contributions in part I.

Requirements	R1.1	R1.2	R1.3	R1.4
Contribution 1	✓	-	-	✓
Contribution 2	✓	✓	-	✓
Contribution 3	✓	-	✓	-
Contribution 4	✓	✓	-	✓

Contribution C4: A framework for torque ripple compensation through repetitive control and a Gaussian-process-based compensation mechanism that is computationally efficient.

An overview of the contributions in part I the relation with requirement R1 is given in Table 1.1.

1.6.2 Contribution II: Feedforward tuning for flexible motion tasks

Increasing task flexibility while maintaining high-performance is key for feedforward control of future systems, see R2 and the overview in Figure 1.11. On-line estimation for adaptive feedforward control (Åström and Wittenmark, 2013) is a promising approach to achieve this. Several adaptive feedforward schemes have appeared in literature, for an overview see Åström and Wittenmark (2013), including a direct approach where an estimation algorithm estimates the feedforward parameters using data while operating in closed loop. Examples of adaptive approaches include Åström and Wittenmark (2013); Butler (2012); Fujimoto (2009). These adaptive approaches are applicable to a wide range of motion tasks similar to model-based feedforward, see Figure 1.11. However, a direct estimation can result in a closed-loop estimation problem (Söderström and Stoica, 1989), or non-convexity (Boeren et al., 2018), potentially resulting in severe performance degradation or slow learning.

A key motivator for the second contribution of this thesis is the wafer-stage application in Figure 1.2 that performs varying and non-resetting point-to-point motion tasks as in Figure 1.3. Here, performance is particularly relevant at the final position, and the error may be larger during the motion. Moreover, the tasks are non-resetting such that batch-wise approaches are not applicable. To track resetting point-to-point tasks a combination of an input-shaper and feedforward control as depicted in Figure 1.14 is presented in Boeren et al. (2014), where the parameters are optimized in a batch-wise setting with a convex LS optimization problem.

The second main contribution of this thesis is the optimal and on-line tuning of feedforward controller parameters using data to remove flexibility limitations

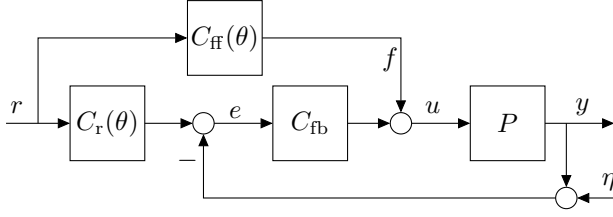


Figure 1.14: Input shaper and feedforward controller setup.

in batch-wise approaches. On-line learning has two key advantages. First, there is a fast performance improvement which can also be within a task, second, it removes the need for references that are resetting to the same initial condition every time, i.e., for continuously varying references. Finally, for systems that exhibit varying dynamics, e.g., due to wear or position-dependencies, the on-line learning approach can update parameters quickly on the basis of data.

From batch-wise to on-line feedforward tuning and its limitations: It is shown in Boeren et al. (2014) that the introduction of an input shaper allows formulating a convex optimization problem with an analytic expression for the feedforward parameters for batch-wise tuning. In the first contribution of this part, the limitations of these batch-wise approaches in an on-line setting are identified, i.e., in an on-line setting, a closed-loop estimation problem can appear if noise is present (Söderström and Stoica, 1989; van den Hof and Schrama, 1994). Moreover, a solution is provided to mitigate the effect of noise on the feedforward parameters to enable both batch-to-batch as well as on-line feedforward parameter tuning. This constitutes the fifth contribution of this thesis.

Contribution C5: The limitation of existing batch-wise feedforward approaches in an on-line setting are identified in the presence of noise, including a practical solution to tune feedforward parameters for tracking of varying reference tasks.

Optimal on-line feedforward estimation: Estimating feedforward coefficient from data in an on-line setting in the presence of measurement noise can yield degraded positioning performance as investigated in Contribution 5. A known result is that instrumental variable (IV) estimators can remove this bias that appears in closed-loop identification (Gilson et al., 2011; Söderström and Stoica, 2002). In Boeren et al. (2018) IV in combination with repeating tasks to optimize feedforward parameters in a batch-wise setting. The aim of this final contribution is to develop an optimal on-line parameter estimator for feedforward to obtain high-performance for a large class of point-to-point references.

This constitutes the sixth contribution of this thesis.

Contribution C6: An optimal instrumental variable based estimator for on-line estimation of feedforward parameters to enable tracking of non-resetting and varying point-to-point references.

1.7 Organization of the thesis

The outline of this thesis is presented in this section. All chapters represent one contribution, are self-contained, and can be read independently. A schematic overview of the thesis is given in Figure 1.15 indicating the main contribution and the experimental setup or motivating application that is used in the chapter. The chapters are based on the following (combinations of) peer-reviewed publications in the same order as they appear in the thesis.

Learning for disturbance rejection.

Ch.2: Mooren, N., Witvoet, G., and Oomen, T. (2022d). Gaussian process repetitive control: Beyond periodic internal models through kernels. *Automatica*, 140:110273

Ch.3: Mooren, N., Witvoet, G., and Oomen, T. (2022a). Gaussian Process Repetitive Control with Application to an Industrial Substrate Carrier Systems with Spatial Disturbances. (*Submitted for journal publication*)

Ch.4: Mooren, N., Witvoet, G., and Oomen, T. (2022c). A Gaussian Process Approach to Multiple Internal Models in Repetitive Control. In *IEEE 17th International Conference on Advanced Motion Control*

Ch.5: Mooren, N., van Meer, M., Witvoet, G., and Oomen, T. (2022b). Compensating Torque Ripples in Coarse Pointing Mechanism for Free-Space Optical Communication: A Gaussian Process Repetitive Control Approach. (*In preparation for journal submission*)

Feedforward tuning for flexible motion tasks.

Ch.6: Mooren, N., Witvoet, G., and Oomen, T. (2019a). Feedforward motion control: From batch-to-batch learning to online parameter estimation. In *2019 American Control Conference (ACC)*. IEEE, pages 947–952, and Mooren, N., Witvoet, G., and Oomen, T. (2019b). From Batch-to-Batch to online learning control: experimental motion control case study. *IFAC-PapersOnLine*, 52(15):406–411

- Ch.7: Mooren, N., Witvoet, G., and Oomen, T. (2022e). On-line Instrumental Variable-Based Feedforward Tuning for Non-Resetting Motion Tasks. *(Submitted for journal publication)*

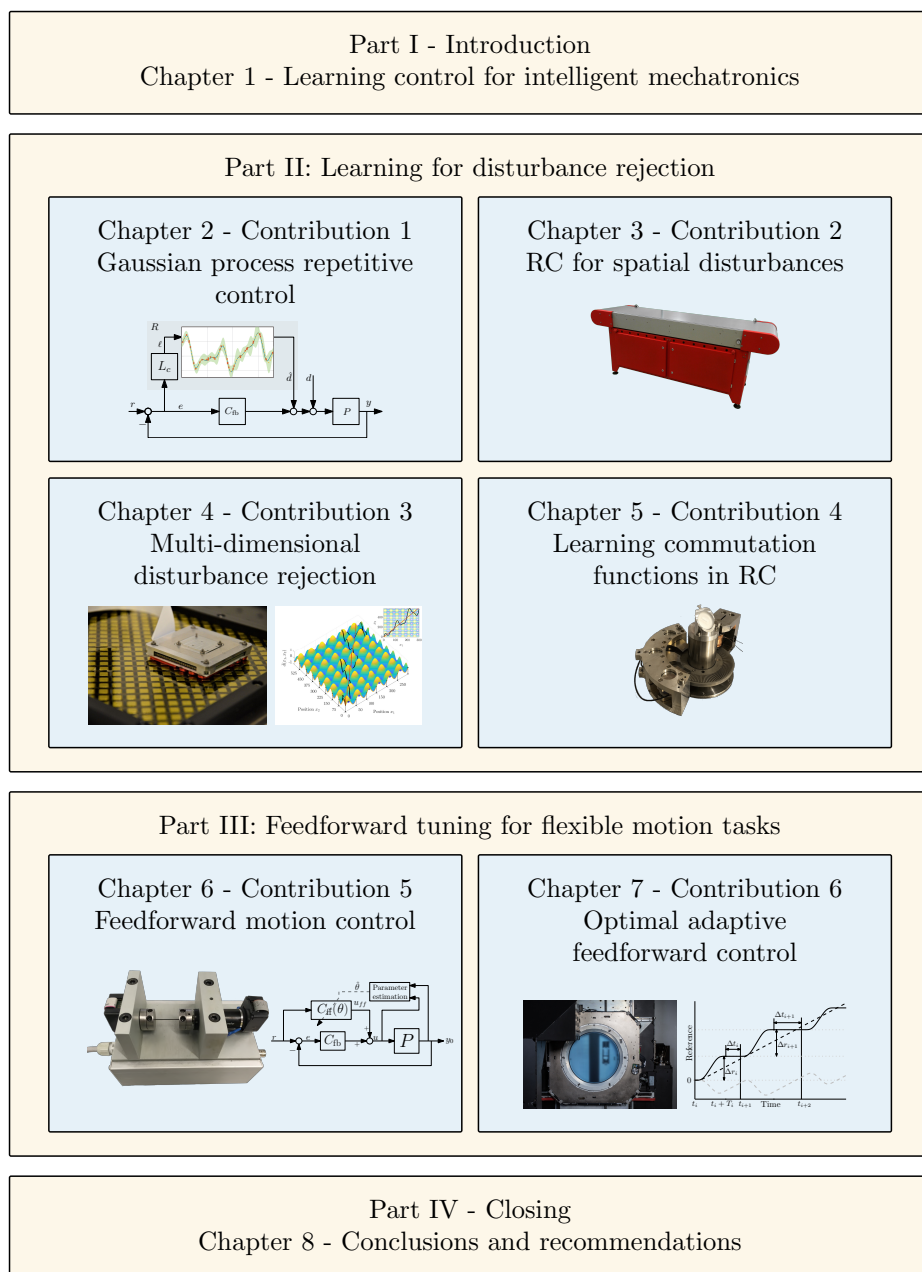
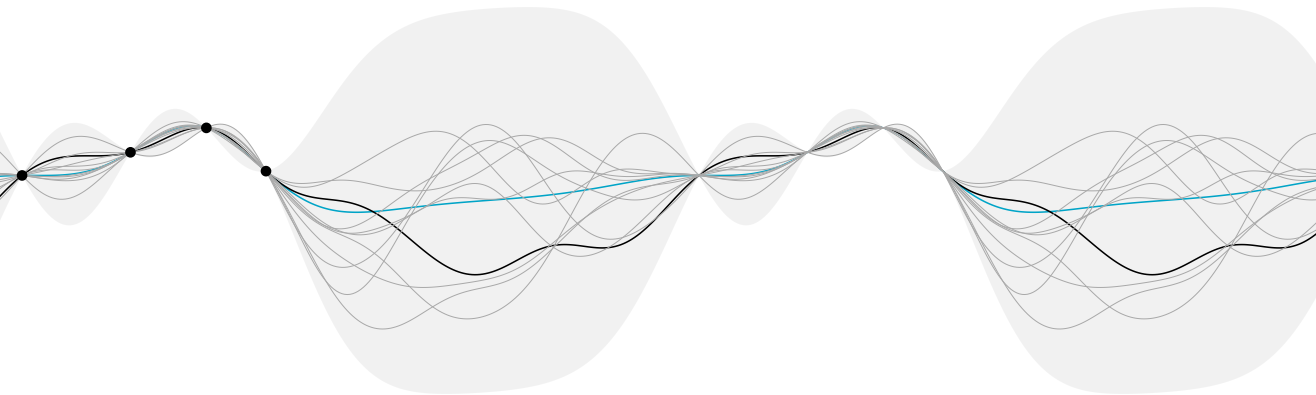


Figure 1.15: Schematic overview of the thesis.

Part II

Learning for disturbance rejection



CHAPTER

2

Gaussian Process Repetitive Control: Beyond Periodic Internal Models through Kernels¹

Abstract: Repetitive control enables the exact compensation of periodic disturbances if the internal model is appropriately selected. The aim of this chapter is to develop a novel synthesis technique for repetitive control (RC) based on a new more general internal model. By employing a Gaussian process internal model, asymptotic rejection is obtained for a wide range of disturbances through appropriate selection of a kernel. The implementation is a simple linear time-invariant (LTI) filter that is automatically synthesized through this kernel. The result is a user-friendly design approach based on a limited number of intuitive design variables, such as smoothness and periodicity. The approach naturally extends to reject multi-period and non-periodic disturbances, exiting approaches are recovered as special cases, and a case study shows that it outperforms traditional RC in both convergence speed and steady-state error.

¹The results in this chapter constitute Contribution 1 of this thesis. The chapter is based on Mooren, N. et al. (2022d). Gaussian process repetitive control: Beyond periodic internal models through kernels. *Automatica*, 140:110273.

2.1 Introduction

Repetitive control (RC) can effectively improve positioning performance for systems that have dominant repeating errors, examples include Merry et al. (2011); Shan and Leang (2012). Asymptotic rejection of repeating disturbances in RC is enabled by the internal model principle (Francis and Wonham, 1976). In particular, a disturbance model is specified as a time-domain memory loop, such that an error that is periodic with the same period can be fully compensated (Goodwin and Sin, 2014; Hara et al., 1988).

Repetitive control is only applicable to periodic signals with a known period due to the traditional delay-based buffer as an internal disturbance model. A key assumption to achieve good performance is that the delay size matches the known period of the disturbance. As a result, RC is very sensitive to small variations in the disturbance period and non-periodic disturbances are even amplified (Steinbuch, 2002). This limits achievable performance in practice, e.g., if the disturbance period is uncertain, or does not fit into the delay size which is an integer multiple of the sample time. In addition, many practical applications have multiple periodic components in the error. If multiple periodic disturbances occur, then their sum may have a very large common multiple, or can even be non-periodic if there is no common multiple, i.e., a situation where traditional RC memory loops are not directly applicable.

Several modifications have been made to the memory loop in RC to improve robustness and performance. In Steinbuch et al. (2007) robustness for small variations in the period time is addressed by incorporating multiple memory loops referred to as higher-order RC (HORC). This results in a trade-off between period uncertainty and sensitivity to non-periodic disturbances, which can be tuned optimally as shown in Pipeleers et al. (2008). In Cao and Ledwich (2002) an approach is presented for disturbance periods that are not an integer multiple of the sample time through interpolation. In Chang et al. (1998) and Zhou et al. (2007) extensions of RC are presented to learn multi-period disturbances by connecting multiple RCs, that each address a single period, in different configurations. However, the design of multi-period RC requires a sequential design procedure to take the interaction between different RCs into account, at the expense of increased complexity in the design procedure, as shown in Blanken et al. (2020a). Moreover, the above approaches are extensions or combinations of the traditional delay-based non-parametric memory loop tailored towards specific refinements instead of generic approaches.

Parametric internal models for RC enable rejection of a wider class of disturbances, e.g., matched basis functions and adaptive RC approaches in Cuiyan et al. (2004); Pérez-Arancibia et al. (2010); Shi et al. (2014). In this approach, a set of basis functions is defined by selecting all relevant frequencies in the error, subsequently, the corresponding coefficients are learned. This allows to learn multi-period disturbances and non-periodic disturbances, but it requires each

specific frequency and its harmonics to be selected a priori.

In view of generic internal models for RC, recent developments in kernel-based approaches such as Gaussian Process (GP) regression have shown to be promising, general results include, identification and control of LTI systems (Pillonetto and De Nicolao, 2010; Scampicchio et al., 2019), non-linear systems (Berkenkamp et al., 2016; Mazzoleni et al., 2020). GP regression is a non-parametric approach that allows learning a wide range of functions, more specifically a distribution over functions is learned, by specifying prior knowledge in the sense of a kernel function through hyperparameters (Murphy, 2012; Williams and Rasmussen, 2006). Gaussian processes are utilized in RC for the compensation of spatially periodic disturbances in Chapter 3. Here, GP regression is employed with a periodic kernel to learn a continuous function from the non-equidistant observations, which is periodic in the spatial domain and potentially non-periodic in the time domain. In contrast to parametric internal models for RC, where basis functions have to be selected explicitly, the GP is a non-parametric approach that only requires a periodic kernel function with a few intuitive tuning parameters. However, the further use of GPs in time-domain RC is not yet explored and the computational complexity of GPs hampers the practical implementation.

Although recently substantial improvements have been made to the robustness and applicability of RC, a unified internal model for RC that covers a wide range of disturbances is not yet available. The aim of this chapter is to present a generic internal model for RC that efficiently uses Gaussian Processes to enable the rejection of a wide variety of disturbances, including, single-period, multi-period, and non-periodic disturbances, by specifying disturbance properties in a kernel function. By learning a continuous function, the disturbance period is not restricted to be an integer multiple of the sample time allowing for rational disturbance periods, which is different in, e.g., Hara et al. (1988); Nagahara and Yamamoto (2016). The following contributions are identified:

- C1 a generic RC design framework and computationally inexpensive internal disturbance model using GP is presented, including prior selection, LTI representation, stability analysis, and a design procedure (Section 3 and 4);
- C2 performance and robustness analysis is performed, providing new insights for RC synthesis from a kernel-based perspective (Section 5); and
- C3 implementation aspects that improve learning within the first period are presented (Section 6).

Several existing approaches are recovered as special cases of the presented framework, and a generic case study is performed to validate the approach.

The chapter is outlined as follows. In Section 2.2, the disturbance attenuation problem and considered class of disturbances are introduced. In Section 2.3, the Gaussian-process-based RC (GPRC) is developed, including LTI case and

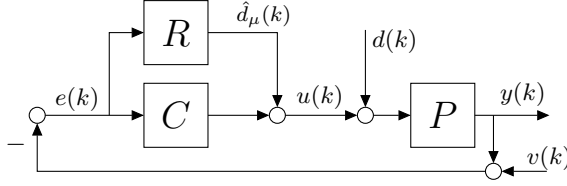


Figure 2.1: Control setting with multi-period disturbance $d(k)$ with $k \in \mathbb{N}$ and repetitive controller R .

2

stability conditions (C1). In Section 2.4, design of GPRC and provides design procedure (C1). In Section 2.5, performance and robustness of GPRC is analyzed, and existing methods are recovered as special cases (C2). In Section 2.6, learning in the first period is improved (C3). In Section 2.7, a case study validates the developed approach and Section 2.8 presents conclusions and ongoing research.

2.2 Problem formulation

2.2.1 Control setting

The considered problem is depicted in Figure 2.1, where P is a discrete-time linear time-invariant (LTI) system, C is a stabilizing feedback controller, and R is an add-on type repetitive controller (RC) that is specified in the forthcoming sections. The goal is to reject the input disturbance $d(k)$ with $k \in \mathbb{Z}_{\geq 0}$, where $d(k)$ is a sampled version of a continuous disturbance $d^c(t)$ with $t \in \mathbb{R}$, i.e., $d(k) = d^c(kT_s)$. Without loss of generality the sample time is scaled to $T_s = 1$. Furthermore, noise v is present that follows an independent, identically distributed (i.i.d.) Gaussian distribution with zero mean.

Definition 2.1. *The control goal is to asymptotically reject the disturbance-induced error $e_d(k)$, given by $e(k)$ in Figure 2.1 for $v(k) = 0$, i.e.,*

$$\lim_{k \rightarrow \infty} e_d(k) = 0 \quad (2.1)$$

by designing R . In the case that R is LTI, then

$$e_d = - \underbrace{P(I + PC)^{-1}}_{S_P} \underbrace{(I + S_P R)^{-1}}_{S_R} d \quad (2.2)$$

where S_R is the modifying sensitivity, that is a measure for the performance improvement through R , and S_P is the process sensitivity when $R = 0$.

Asymptotic rejection for a wide range of disturbances is obtained through a generic internal disturbance model in R which is investigated next.

Remark 2.2. *The RC configuration in R is slightly different from the traditional one, e.g., as in Steinbuch et al. (2007). If R is linear, these are equivalent due to the commutative property of linear systems. The presented one has particular advantages in view of the GP prior as the RC output is equal to the disturbance in an ideal setting.*

2.2.2 Internal model control

The internal model principle states that asymptotic disturbance rejection is obtained by including a model of the disturbance generating system in a stable feedback loop, see, e.g., Francis and Wonham (1976). By the final value theorem (Schiff, 1999), it can be shown that a constant disturbance $d(k) = 1$ with \mathcal{Z} -transform $(1 - z^{-1})^{-1}$, is asymptotically rejected with a factor $(1 - z^{-1})^{-1}$ in the open-loop PC . For periodic disturbances with period $T \in \mathbb{N}$, a model of the disturbance generating system consists of a delay-based buffer z^{-N} , with $N = T$, in a feedback loop, i.e.,

$$R^{\text{conv}}(z) = \frac{z^{-N}}{1 - z^{-N}} = \frac{1}{z^N - 1}, \quad (2.3)$$

which is the most simple form of conventional RC, that it is often employed with a learning filter for stability, to asymptotically attenuate any disturbance with period time T , see, e.g., Goodwin and Sin (2014). However, disturbances with a rational period time $T \in \mathbb{R}$, as illustrated in Figure 2.2, do not fit in these traditional buffers and require additional interpolation.

The following general class of disturbances is considered in this chapter.

Definition 2.3. *The continuous-time disturbance is defined as*

$$d^c(t) = \sum_{i=1}^{n_d} d_i^c(t), \quad (2.4)$$

which is a multi-period disturbance consisting of $n_d \in \mathbb{N}$ periodic scalar-valued signals $d_i^c(t) \in \mathbb{R}$ that are smooth and satisfy

$$d_i^c(t) = d_i^c(t - \beta T_i), \quad (2.5)$$

with $\beta \in \mathbb{Z}$, and $T_i \in \mathbb{R}$ is the period time of the i^{th} component. Moreover, the frequency content of the disturbance signal is contained below the Nyquist frequency, i.e., π , to avoid aliasing.

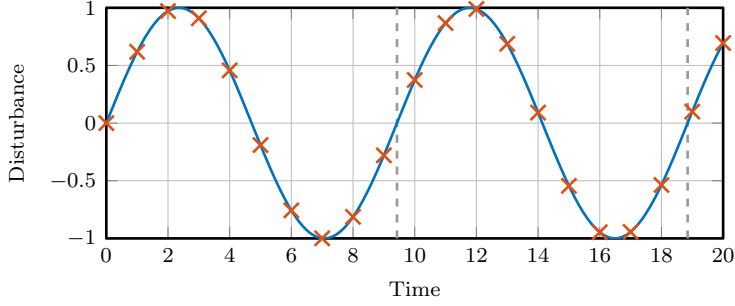


Figure 2.2: Example: a continuous time disturbance $d^c(t)$ (—) with period $T = 3\pi$ from which discrete samples $d(k)$ (×) with sample frequency 1 Hz are taken, i.e., the discrete time sequence $d(k)$ is non-periodic for all k while the continuous time signal $d^c(t)$ is periodic with the period time $T \in \mathbb{R}$.

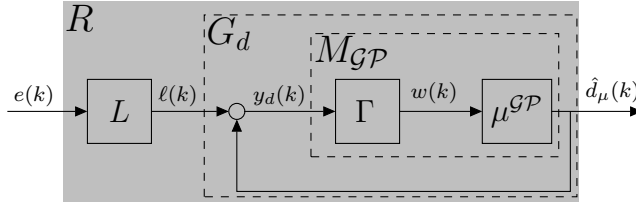


Figure 2.3: Gaussian process repetitive controller R with Gaussian-process-based buffer $M_{\mathcal{GP}} = \mu^{\mathcal{GP}}\Gamma$ and internal model of the disturbance generating system G_d .

The disturbance (2.4) is a single-period disturbance if $n_d = 1$ or a multi-period disturbance with $n_d > 1$; in the latter case $d^c(t)$ is either periodic with a period equal to the least common multiple (LCM)

$$T = \text{lcm}\{T_1, T_2, \dots, T_{n_d}\} \in \mathbb{R}, \quad (2.6)$$

or is non-periodic if there is no least common multiple.

Existing extensions of traditional internal models for RC that cover multi-period disturbances lead to a complicated design procedure due to interaction between different RCs, see, e.g., Blanken et al. (2020a). Alternatively, the buffer size N in (2.3) can be chosen equal to the common multiple (2.6) yielding slow learning performance if T is very large. Yet, a generic internal model for the class of disturbances in Definition 2.3 is not available.

2.2.3 Gaussian process RC setup

The RC structure that is presented in this chapter is shown in Figure 2.3, where L is a learning filter and the proposed GP-based internal model of the disturbance generating system is given by G_d with $M_{\mathcal{GP}} = \mu^{\mathcal{GP}}\Gamma$ the GP-based memory. Moreover, $\Gamma \in \mathcal{RH}_{\infty}^{N \times 1}(z)$ is a delay line that accumulates the past $N \in \mathbb{N}$ samples of its input $y_d \in \mathbb{R}$, i.e.,

$$\Gamma := \begin{cases} x(k+1) &= Ax(k) + By_d(k) \\ w(k) &= Cx(k) + Dy_d(k), \end{cases} \quad (2.7)$$

where $x \in \mathbb{R}^N$ is the state, and

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] = \left[\begin{array}{cc|c} 0 & 0 & 1 \\ I_{N-1} & 0 & 0 \\ 0 & 0 & 1 \\ I_{N-1} & 0 & 0 \end{array} \right], \quad (2.8)$$

which results in the vector valued signal $w(k) \in \mathbb{R}^N$. Finally, $\mu^{\mathcal{GP}} \in \mathbb{R}^{1 \times N}$ is a vector of, possibly time-varying, coefficient that are designed and formally introduced in the forthcoming sections.

2.2.4 Problem definition

In this chapter, a systematic design approach for the repetitive controller R is presented, by developing a generic Gaussian-process-based internal disturbance model for the disturbances in Definition 2.3, see Figure 2.3. A Gaussian process (GP) specifies disturbance properties through a kernel function and hyperparameters, which enables to model a wide range of disturbances as in Definition 2.3. The following requirements are addressed:

- R1 asymptotic rejection for a wide range of disturbances, i.e., periodic, multi-period, and non-periodic disturbances, in the setting in Figure 2.1, and
- R2 a user-friendly approach for synthesizing R by specifying disturbance properties, such as periodicity and smoothness, through a kernel function.

A framework that utilizes GP-based internal models in RC to cover both R1 and R2 is presented.

2.3 Gaussian process buffer in repetitive control

In this section, the generalized Gaussian process repetitive control (GPRC) framework to synthesize the repetitive controller R is introduced. The GPRC

setup is further outlined in Section 2.3.1, after which the GP internal model is presented in Section 2.3.2. Conditions under which GPRC is LTI and non-conservative stability conditions are provided in Section 2.3.3 and 2.3.4 respectively, constituting contribution C1.

2.3.1 Gaussian process repetitive control setup

The GP-based repetitive controller R in Figure 2.3 contains the GP-based disturbance model G_d that is designed using GP-regression to generate a continuous model of the true disturbance d^c . A sample of \hat{d}^c is parameterized as

$$\hat{d}_\mu(k) = \mu_k^{\mathcal{GP}} w(k) \quad (2.9)$$

where $\mu_k^{\mathcal{GP}} \in \mathbb{R}^{1 \times N}$ are, in general, time-varying coefficients that follow from GP regression elaborated in detail in Section 2.3.2. Moreover, in Section 2.3.3 mild conditions are provided under which $\mu^{\mathcal{GP}}$ is time invariant.

The data used for GP-regression is given by the noisy data samples in

$$w(k) = [y_d(k) \quad y_d(k-1) \quad \dots \quad y_d(k-N+1)]^\top \quad (2.10)$$

to estimate a continuous function \hat{d}^c of the true disturbances d^c for compensation. To compose the data set for GP regression, define the vector with corresponding time instances

$$X(k) = [t(k) \quad t(k-1) \quad \dots \quad t(k-N+1)]^\top, \quad (2.11)$$

constituting the data set $\mathcal{D}_N(k) = (w(k), X(k))$ that contains N pairs (y_d, t) of observations. At each sample k the data $\mathcal{D}_N(k)$ is used to perform GP regression resulting in the coefficients $\mu_k^{\mathcal{GP}}$ as shown next.

Remark 2.4. *Note that all the past data can be used for GP regression, i.e., all samples $y_d(k')$ with $k' \in \{1, 2, \dots, k\}$ at sample k such that $w(k) \in \mathbb{R}^k$. However, here N is fixed analog to traditional RC approaches, generalization to larger buffers is conceptually straightforward, for example, using online GP regression (Bijl et al., 2017; Umlauft et al., 2020).*

2.3.2 Gaussian process disturbance model

The compensation signal (2.9) with coefficients $\mu_k^{\mathcal{GP}}$ is an estimate of the disturbance that is obtained using data and prior knowledge through GP regression. In this section, it is shown how GP regression is used to model the disturbance and consequently synthesize these coefficients.

First, consider the prior disturbance model \hat{d}^c given by a GP

$$\hat{d}^c(t) \sim \mathcal{GP}(m(t), \kappa(t, t')), \quad (2.12)$$

that is a distribution over functions which is completely determined by its prior mean function $m(t)$ and prior covariance function $\kappa(t, t') : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^{n \times m}$ with n and m the size of t and t' respectively. The choice of a covariance function depends on the disturbance properties, e.g., periodicity, which is investigated in detail in Section 2.4.2 by taking the additive structure in (2.4) into account. For presentation purposes, $m(t) = 0$, the results can easily be extended for non-zero mean function, see, e.g., Murphy (2012). Next, it is shown how the prior knowledge (2.12) and the data \mathcal{D}_N is used to compute $\mu_k^{\mathcal{GP}}$ in (2.9).

The data set \mathcal{D}_N contains noisy observations of the model $\hat{d}^c(t)$ in (2.12), i.e.,

$$w(k) = \begin{bmatrix} \hat{d}^c(t(k)) \\ \hat{d}^c(t(k-1)) \\ \vdots \\ \hat{d}^c(t(k-N+1)) \end{bmatrix} + \epsilon, \quad (2.13)$$

where $\epsilon \sim \mathcal{N}(0_N, \sigma_n^2 I_N)$, with 0_N a matrix of zeros of size $N \times N$, that follows an independent, identically distributed (i.i.d.) Gaussian distribution with zero mean and variance σ_n^2 as a result of the noise v .

Predictions of the disturbance model for compensation can be made at arbitrary $X_* \in \mathbb{R}$, denoted by $\hat{d}^c(X_*) = \hat{d}_*$, based on the data \mathcal{D}_N and prior (2.12). Moreover, for the application in RC, predictions are made at the current time, i.e., the test point becomes $X_* = t(k) \in \mathbb{Z}_{\geq 0}$ since $T_s = 1$. The joint prior distribution

$$\begin{bmatrix} w \\ \hat{d}_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K + \sigma_n^2 I_N & K_* \\ K_*^\top & K_{**} \end{bmatrix} \right), \quad (2.14)$$

defines the correlation between the data $w(k)$ and the test point X_* , where $K = \kappa(X, X) \in \mathbb{R}^{N \times N}$ is the covariance function κ evaluated at all pairs of (X, X) , and similarly for $K_* = \kappa(X, X_*) \in \mathbb{R}^N$ and $K_{**} = \kappa(X_*, X_*) \in \mathbb{R}$. From (2.14) it follows that the predictive posterior distribution at the test point X_* becomes $p(\hat{d}_* | \mathcal{D}_N, X_*) = \mathcal{N}(\hat{d}_\mu, \Sigma)$ where

$$\hat{d}_\mu(k) = K_*^\top (K + \sigma_n^2 I_N)^{-1} w(k), \quad (2.15a)$$

$$\Sigma(k) = K_{**} - K_*^\top (K + \sigma_n^2 I_N)^{-1} K_*, \quad (2.15b)$$

are the mean and variance respectively, see, e.g., (Murphy, 2012, Chapter 4.3). The posterior mean \hat{d}_μ is equal to the maximum a posteriori (MAP) estimate, and is used for compensation, yielding that the coefficients $\mu_k^{\mathcal{GP}}$ in (2.9) are given by

$$\mu_k^{\mathcal{GP}} = K_*^\top (K + \sigma_n^2 I_N)^{-1}. \quad (2.16)$$

By performing GP regression (2.15) at each sample, updated coefficients $\mu_k^{\mathcal{GP}}$ are obtained through (2.16) for compensation. In contrast to traditional RC with internal disturbance model (2.3), GPRC enables compensation within the first period. Furthermore, by using a GP function estimator a more general setting is established in which also multi-period and non-periodic disturbances can be captured with suitable prior, as shown in Section 2.5.

2.3.3 LTI representation of GPRC

In this section, conditions are presented under which the coefficients $\mu^{\mathcal{GP}}$ in (2.16) are time invariant, rendering the repetitive controller in Figure 2.3 to be LTI.

Assumption 2.5. *Consider the following assumptions on the covariance function κ and training data set \mathcal{D}_N ;*

A1 the covariance function κ in (2.12) is a stationary function, i.e., a function of the relative difference $\tau(k) = t(k) - t'(k)$, see, e.g., (Williams and Rasmussen, 2006, p.82);

A2 the vector $X(k) \in \mathbb{R}^N$ in (2.11) contains equidistantly sampled time instances with N fixed; and

A3 the test point $X_(k) = t(k + \alpha)$ with $\alpha \in \mathbb{Z}$ constant.*

Theorem 2.6. *Under Assumption 2.5, the repetitive controller R in Figure 2.3 is LTI and given by*

$$R = \frac{M_{\mathcal{GP}}L}{1 - M_{\mathcal{GP}}}, \quad (2.17)$$

where the GP buffer $M_{\mathcal{GP}}$ is a finite impulse response (FIR) filter

$$M_{\mathcal{GP}}(z) = \mu^{\mathcal{GP}} \Gamma(z) = \sum_{i=0}^{N-1} \mu_i^{\mathcal{GP}} z^{-i}, \quad (2.18)$$

with time-invariant coefficients $\mu^{\mathcal{GP}}$.

Proof. If $\mu^{\mathcal{GP}}$ in (2.16) is time-invariant under Assumption 2.5, then R in Figure 2.3 is LTI and of the form (2.17). Hence, it is shown that (2.16) is time-invariant under A1-A3. First, K is obtained by evaluating the kernel function κ at all combinations of $(X(k), X(k))$ with $X(k)$ in (2.11), these combinations

are given by

$$\begin{aligned}\tau(k) &= X(k)1_N^\top - 1_N X(k) \\ &= \begin{bmatrix} t_k - t_k & t_k - t_{k-1} & \dots & t_k - t_{k-N+1} \\ t_{k-1} - t_k & t_{k-1} - t_{k-1} & & \vdots \\ \vdots & & \ddots & \\ t_{k-N+1} - t_k & \dots & & t_{k-N+1} - t_{k-N+1} \end{bmatrix}\end{aligned}$$

which is Toeplitz, $1_N \in \mathbb{R}^N$ is a matrix of ones, and $t_{k-i} = t(k-i)$. Second, from assumption A2 it follows that $\tau(k) = \tau(j) \forall (k, j) \in \mathbb{Z}$. Similarly for K_* that is obtained by evaluating κ at all pairs $(X(k), X_*)$ given by

$$\bar{\tau}(k) = X^\top - 1_N^\top X_* = [X_* - t_k \quad \dots \quad X_* - t_{k-N+1}]$$

which, under assumptions A3, satisfy that $\bar{\tau}(k) = \bar{\tau}(j) \forall (k, j) \in \mathbb{Z}$. Third, under assumption A1, the kernel matrices K and K_* are a function of $\kappa(\tau)$ and $\kappa(\bar{\tau})$ respectively. Since, τ and $\bar{\tau}$ are time-invariant, so are K and K_* , as a result, rendering (2.16) time invariant, such that R is of the form (2.17) which completes the proof. \square

Consequently, the RC output (2.9) is given by the following FIR operation

$$\hat{d}_\mu(k) = \sum_{i=0}^{N-1} \mu_i^{\mathcal{GP}} y_d(k-i). \quad (2.19)$$

with fixed coefficient $\mu^{\mathcal{GP}}$ that follow from (2.16). In addition, the internal disturbance model is now also LTI and given by

$$G_d = \frac{M_{\mathcal{GP}}}{1 - M_{\mathcal{GP}}}. \quad (2.20)$$

As a result, synthesis of a generalized, possibly multi-period, RC reduces to the selection of a covariance function κ . By evaluating (2.16), this framework then facilitates the construction of appropriate FIR coefficients $\mu^{\mathcal{GP}}$, through which it enables efficient implementation of GPs in RC, allowing for larger flexibility, and offers superior performance in the first period due to continuous updating.

2.3.4 Stability analysis

In this section, the stability of GPRC with LTI repetitive controller R in (2.17) is analyzed in the setting in Figure 2.1, resulting in non-conservative stability conditions.

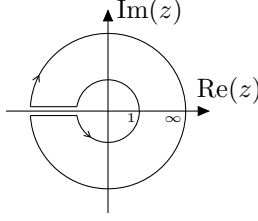


Figure 2.4: Nyquist contour D with inner radius 1, outer radius infinity, and the parallel lines infinitely close to the real axis.

2

Theorem 2.7. Consider Figure 2.1 with repetitive controller (2.17) in Theorem 2.6, a specified kernel function κ and a buffer size N . Suppose all poles of S_P and L are in the open unit disk, and the feedback loop in Figure 2.1 is asymptotically stable, then the closed-loop is stable if and only if the image of $-M_{GP}(z)(1 - S_P(z)L(z))$;

- makes no encirclements around the point -1 , and
- does not pass through the point -1 ,

as z traverses the Nyquist contour D in Figure 2.4, see, Skogestad and Postlethwaite (2007).

Proof. The setting in Figure 2.1 is stable if and only if $S_P \in \mathcal{RH}_\infty$ and $S_R = (1 + S_P R)^{-1} \in \mathcal{RH}_\infty$. First, S_P is proper and stable by the assumption in Theorem 2.7. Second, substituting R (2.17) in S_R (2.2) gives

$$S_R = \frac{1 - M_{GP}}{1 - M_{GP}(1 - S_P L)}, \quad (2.21)$$

the Nyquist theorem, see, e.g., (Skogestad and Postlethwaite, 2007, Theorem 4.14), states that S_R is stable if and only if the image of $-M_{GP}(z)(1 - S_P(z)L(z))$ i) encircles the point -1 in anti-clockwise direction P_{ol} times, and ii) does not pass through the point -1 as z traverses the D , where P_{ol} is the number of poles of $-M_{GP}(1 - S_P L)$ inside D .

It remains to show that $P_{ol} = 0$, i.e., there are no unstable poles in $-M(1 - S_P L)$. This holds true since M_{GP} is a FIR filter, see Theorem 2.6, with all poles in the origin, and, S_P and L are stable by the assumptions in Theorem 2.7, which completes the proof. \square

Theorem 2.7 provides a non-conservative condition to check stability given M_{GP} in (2.18) that contains the GP buffer. If the resulting closed-loop is unstable, e.g., due to modeling errors, the following slightly more conservative frequency-domain condition is provided to tune R for stability.

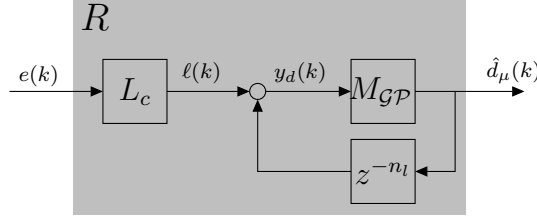


Figure 2.5: Practical RC implementation with a non-causal learning filter L with causal equivalent L_c and preview z^{n_l} .

Corollary 2.8. *Theorem 2.7 is satisfied if*

$$M_{GP}(e^{j\omega}) (1 - S_P(e^{j\omega})L(e^{j\omega})) < 1, \quad (2.22)$$

for all $\omega \in [0, \pi]$.

Corollary 2.8 yields that the closed-loop is stable if i) a perfect model is available, i.e., $LS_P^{-1} = 1$, or ii) if model errors appear $L \neq S_P^{-1}$ then M_{GP} in (2.18) must be designed to act as a robustness filter and stabilize the closed-loop S_R , which is further addressed in Section 2.5.

2.4 Design methodology for Gaussian process RC

In this section, design guidelines are presented for the learning filter and selection of suitable prior knowledge through the covariance function κ for the class of disturbances in Definition 2.3. Finally, a procedure to implement GPRC is provided.

2.4.1 Learning filter design

The learning filter L in the repetitive controller (2.17) is present for stability, i.e., from Theorem 2.7 it follows that by designing L as

$$L = S_P^{-1}, \quad (2.23)$$

then $(1 - S_PL)$ renders zero satisfying Theorem 2.7 regardless of M_{GP} .

Direct inversion of S_P may lead to an unstable or non-causal inverse, e.g., if P contains non-minimum phase zeros. By employing finite preview a bounded approximate inverse of S_P can be obtained, e.g., using Zero-Phase-Error-Tracking-Control (ZPETC) (Tomizuka, 1987; van Zundert and Oomen, 2017) yielding L

of the form

$$L = L_c z^{n_l} \approx S_P^{-1} \quad (2.24)$$

where L_c is causal and z^{n_l} with $n_l \leq N$ is a possible finite preview.

A practical implementation for the non-causal L filter (2.24) is presented in Figure 2.5, where the error is filtered with the causal part L_c yielding

$$y_d(k) = -L_c S_P(d(k) + \hat{d}_\mu(k)) + q^{-n_l} \hat{d}_\mu(k), \quad (2.25)$$

$$= -d(k - n_l), \quad (2.26)$$

where q is the forward time-shift operator, to be a delayed version of the disturbance with n_l samples. This delay is compensated by a preview in the memory M_{GP} , i.e., the test point becomes $X_* = t(k + n_l)$, to implement the non-causal part of L .

Remark 2.9. Note that $X_* = t(k + n_l)$ is an estimate of \hat{d}^c at $t(k + n_l)$ being n_l samples in the future. This is possible by introducing smoothness in the GP prior as shown later.

2.4.2 Prior selection

In this section, a suitable covariance function κ in (2.12) that specifies prior knowledge for the class of disturbances in Definition 2.3 is presented.

The additive structure in Definition 2.3 is imposed on the disturbance model (2.12) by parameterizing it as a sum of n_d periodic functions with periods T_i , i.e.,

$$\hat{d}^c(t) = \sum_{i=1}^{n_d} \hat{d}_i^c(t), \text{ with } \hat{d}_i^c(t) \sim \mathcal{GP}(0, \kappa_i(t, t')), \quad (2.27)$$

where \hat{d}_i^c are samples from n_d independent GPs with periodic covariance function κ_i . Hence, $\hat{d}^c(t)$ in (2.27) is referred to as an additive GP, see, e.g., Durrande et al. (2011), with an additive covariance function

$$\kappa(t, t') = \sum_{i=1}^{n_d} \kappa_i(t, t'), \quad (2.28)$$

that is simply the sum of the individual covariance functions κ_i . The periodic covariance function κ_i is of the form

$$\kappa_i(t, t') = \sigma_{f,i}^2 \exp \left(\frac{-2 \sin^2 \left(\frac{\pi(t-t')}{T_i} \right)}{l_i^2} \right), \quad (2.29)$$

with hyperparameters $\Theta_i = \{T_i, l_i, \sigma_{f,i}\}$ where

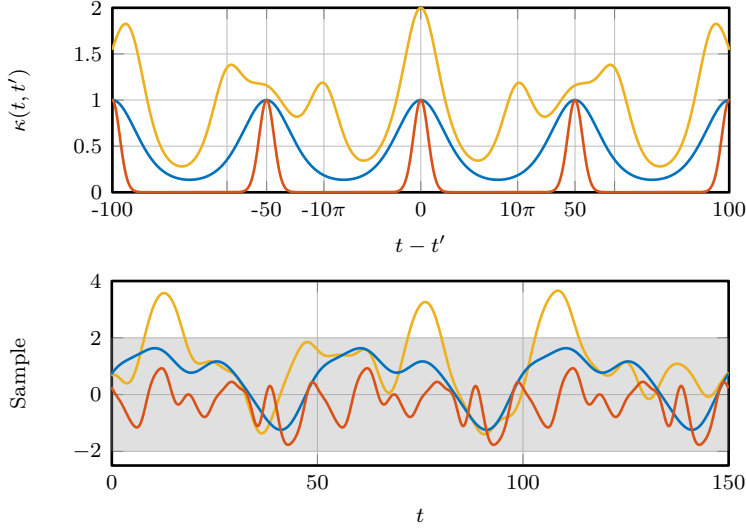


Figure 2.6: Example: Top plot shows three examples of periodic covariance functions $\kappa_i(t, t')$ as a function of $t - t'$ with hyperparameters $T_1 = 50$, $l_1 = \sigma_{f,i} = 1$ in (—) and with $l_2 = 0.25$ in (—). This shows that if $t - t'$ is close to zero or the period T_i , then t and t' are highly correlated. Also the sum of two periodic kernels with periods $T_1 = 50$ and $T_2 = 10\pi$ is shown (—) yielding a non-periodic kernel function. The bottom plot shows random samples taken from the distributions $\mathcal{N}(0, \kappa_i(t, t'))$, these samples both periodic and have more (—) or less (—) smoothness, or become non-periodic (—) with a non-periodic kernel that is the sum of two periodic kernels.

- $T_i \in \mathbb{R}$ is the period of the i^{th} component;
- $l_i \in \mathbb{R}$ is the smoothness of \hat{d}_i^c , i.e., choosing l large implies less higher harmonics and vice versa; and
- $\sigma_{f,i} \in \mathbb{R}$ is a gain relative to the other components and the noise variance σ_n^2 .

The periodic covariance function (2.29) is often encountered in literature, see, e.g., (Williams and Rasmussen, 2006, Chapter 4.2), (Duvenaud, 2014, Chapter 2.2). Note that κ is non-periodic if there is no least common multiple as in (2.6) for non-periodic disturbances. An example of the periodic kernel function κ_i , and a non-periodic kernel that is a sum of two periodic kernels, including random samples taken from the prior distributions $\mathcal{N}(0, \kappa_i(t, t'))$ are shown in Figure 2.6. This allows to capture both period and non-periodic disturbances in the GP-based internal disturbance model.

In GPRC with kernel function (2.28) the disturbance period is included through the hyperparameter T_i that may be rational, in contrast to traditional RC, allowing to reject disturbances with a rational period time. The number of components n_d specifies if the disturbance is single-period ($n_d = 1$) or multi-period ($n_d \geq 2$) and can be determined with for example a power spectral density (PSD) estimate of a measured error signal, where n_d equals the number of fundamental frequencies. Finally, if the periods T_i do not have a common multiple, then the resulting kernel function is non-periodic. Hence, the prior (2.28) is flexible and can be tuned with only a limited number of intuitive hyperparameters.

2.4.3 Design procedure

The following procedure summarizes the design steps that are required to implement GP-based RC.

Procedure 2.10 (GPRC design).

Given a measured frequency response function (FRF) $\hat{S}_P(e^{j\omega})$ and a parametric model S_P , perform;

1. Invert \hat{S}_P to obtain L_c and non-causal part z^{n_l} with $n_l \geq 0$ in (2.24), e.g., using ZPETC.
2. Determine n_d in (2.27), e.g., using a PSD estimate of the error. Then, set $i = 1$ and repeat the following.
 - a) Choose the period T_i , smoothness l_i and gain $\sigma_{f,i}$ for κ_i in (2.29).
 - b) until $i = n_d$, set $i \rightarrow i + 1$ and repeat step 2a.
3. Choose a buffer size $N \in \mathbb{N}$, e.g., a good starting point is $N \geq \sum_{i=1}^{n_d} T_i$ which yields sufficient design freedom, although smaller buffer sizes are possible with appropriate prior, see Remark 2.11.
4. Define

$$X = \begin{bmatrix} N & N-1 & \dots & 1 \end{bmatrix}^\top, \quad (2.30a)$$

$$X_* = N + n_l, \quad (2.30b)$$

and evaluate κ in (2.28) for $\kappa(X, X)$ and $\kappa(X, X_*)$ to obtain K and K_* respectively.

5. Compute FIR coefficient $\mu^{\mathcal{GP}}$ in (2.16) and verify stability with \hat{S}_P using Theorem 2.7 or Corollary 2.8 (Remark 2.12).
-

Remark 2.11. To model a periodic signal with period T at least T independent parameters are required. By including a correlation through smoothness ($l > 0$) or periodicity in the kernel, a smaller buffer size $N < T$ can be used in practice.

Remark 2.12. The FIR filter $M_{\mathcal{GP}}$ influences stability if $L \neq S_P^{-1}$, it is shown in Section 2.5 that increasing smoothness yield more robustness for modeling errors.

2.5 Performance and robustness

The generic GPRC framework introduced in the previous sections is further analyzed, i.e., it is shown under which conditions traditional RC and Higher-order RC (HORC) (Steinbuch et al., 2007) are recovered as a special case of GPRC. Furthermore, by a suitable kernel choice GPRC improves robustness for period variations or reduces the sensitivity with respect to noise similar to Steinbuch et al. (2007). Furthermore, GPRC applied to multi-period disturbances and disturbances with a rational period time is analyzed.

2.5.1 Recovering traditional RC

GPRC recovers traditional RC for a specific type of prior, i.e., a periodic kernel without smoothness. In traditional RC the buffer $M_T = z^{-(N-n_l)}$ is a pure delay, hence, the output is simply a delayed version of the input. This is recovered in GP-based RC as follows.

Theorem 2.13. In the setting in Figure 2.5 and under the conditions in Theorem 2.6, then with $N = T \in \mathbb{N}$, a periodic kernel (2.28) where $n_d = 1$ and $T = N$, $\sigma_n^2 = 0$, $\sigma_f^2 = 1$, and $l \rightarrow 0$, the memory

$$M_{\mathcal{GP}} = z^{-(N-n_l)}, \quad (2.31)$$

recovers traditional RC.

Proof. To show that $M_{\mathcal{GP}}(z) = z^{-(N-n_l)}$, note that this is equivalent to showing that $\hat{d}_\mu(k) = y_d(k - N + n_l)$. The output $\hat{d}_\mu(k) = \mu^{\mathcal{GP}} w(k)$ with $w(k)$ in (2.10), hence by showing that the vector

$$\mu^{\mathcal{GP}} = [0_{N-n_l-1} \quad 1 \quad 0_{n_l-1}]^\top \in \mathbb{R}^N \quad (2.32)$$

implies that $M = z^{-(N-n_l)}$. Substitute $\sigma_n^2 = 0$ and $\sigma_f^2 = 1$, then in the limit case the kernel function (2.29) is of the form

$$\lim_{l \rightarrow 0} \kappa = \lim_{l \rightarrow 0} \exp \left(\frac{a(k)}{l^2} \right) = \begin{cases} 0 & \text{if } a(k) \neq 0 \\ 1 & \text{if } a(k) = 0 \end{cases}$$

where $a(k) = -2 \sin^2 \left(\frac{\pi \tau(k)}{T_i} \right) = 0 \forall \tau(k) = \beta T_i$ with $\tau(k) = X(k) - X'(k)$ and $\beta \in \mathbb{Z}$. With $X(k)$ in (2.11) this leads to $K = I_N$ and $\lim_{\ell_i \rightarrow 0} K_*^\top K^{-1}$ is of the form (2.32) which completes the proof. \square

Hence, by setting smoothness to zero and the kernel period limited to an integer, the traditional RC memory is recovered. Next, it is shown that GPRC is not limited to disturbances that have an integer period time through introducing smoothness.

Remark 2.14. *Theorem 2.13 shows that setting $l \rightarrow 0$ recovers traditional RC, which does not take inter-sample behavior into account (Nagahara and Yamamoto, 2016). In the following subsections, the smoothness $l > 0$ resulting in a smooth and continuous disturbance estimate, also in-between the discrete data point, i.e., the inter-sample modeling error is reduced. In Nagahara and Yamamoto (2016), sampled-data signal reconstruction is employed to generate a continuous-time disturbance model that explicitly takes inter-sample behavior into account in RC.*

2.5.2 GPRC for discrete-time non-periodic disturbances

Traditional RC is not applicable to rational period times as in Definition 2.3 with $T \in \mathbb{R}$ which are non-periodic in discrete time, for these disturbances additional interpolation is required, see, e.g., Cao and Ledwich (2002). In contrast, it is shown that GPRC can suppress disturbances that have a rational period time.

In GPRC the disturbance period is specified through the kernel function (2.29) where $T_i \in \mathbb{R}$, and is not necessarily related to the buffer size $N \in \mathbb{N}$ as in traditional RC. It is shown in Theorem 2.13 that if $l \rightarrow 0$ and T is an integer, then $M_{\mathcal{GP}}$ is a pure delay such that \hat{d}_μ in (2.19) depends solely on $y_d(k - T)$. In the case that T is rational then $y(k - T)$ is not directly available, i.e., it is in-between two samples, but it is estimated from the available inputs using a smoothness $l > 0$ also estimating the disturbance in-between samples. Hence, smoothness enables interpolation for disturbances with a rational period time as shown in the following example.

Example 2.15. *Consider the problem of rejecting a disturbance with a rational period time $T_d = 10.5$ samples, for which the kernel (2.29) with $T = T_d$, $l = 10$ and $\sigma_f = 1$ is designed. The resulting modifying sensitivity S_R and the FIR coefficients $\mu^{\mathcal{GP}}$ are shown in Figure 2.7. As a comparison, a traditional RC with $N = 11$ is also provided.*

The modifying sensitivity S_R shows that GPRC (—) attenuates the disturbance at the fundamental frequency $1/T_d$ and its harmonics, through combining

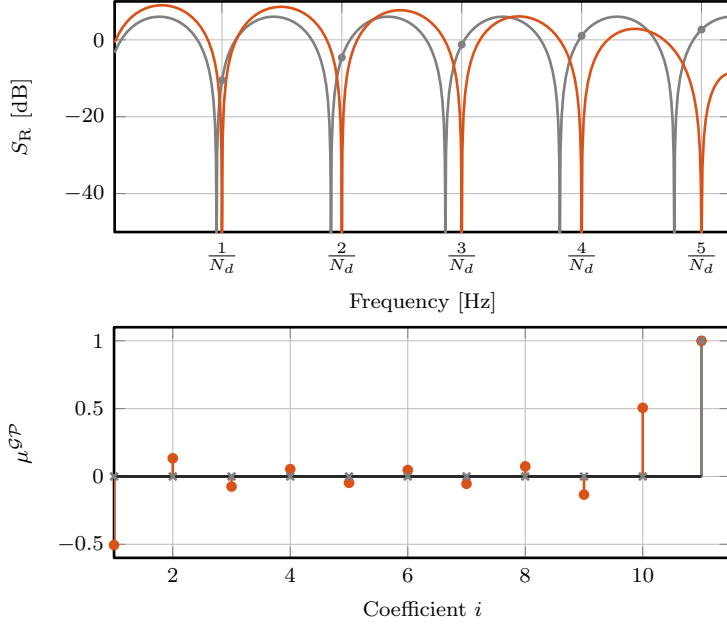


Figure 2.7: Example 2.15: Modifying sensitivity function S_R (top plot) and impulse response of M_{GP} (bottom plot) for GPRC without smoothness and $T = 11$ (—), and with smoothness with $T = 10.5$ (—). Including smoothness yield that many FIR coefficients μ^{GP} are non-zero (●) for automatic interpolation, which enables suppression at $1/N_d$ and higher harmonics, whereas traditional RC performance in much worse (×).

the available inputs $w(k)$ to estimate $\hat{d}^c(k - T_d)$ as in (2.19) with coefficients μ^{GP} (●), yielding automatic interpolation. In contrast, it is evident that traditional RC (—) attenuates the disturbance at the wrong frequency $1/N$ which may even amplify the actual disturbance indicated by (●).

2.5.3 Recovering HORC

GPRC can improve the robustness of RC with respect to noise or uncertain period times similar to HORC, where $p \in \mathbb{N}$ buffers z^{-N} are combined, see, e.g., Pipeleers et al. (2008); Steinbuch et al. (2007). Next, it is shown that HORC is a special case of GPRC, while at the same time the GP framework allows for substantially larger design freedom for the HORC controller. Consider the following Lemma and Theorem that provides conditions under which HORC is recovered, after which two examples illustrate the extended design freedom.

Lemma 2.16. *Consider GPRC under Assumption 2.5, then for all $(i, j) \in$*

$\{1, 2, \dots, N\}$ and $i \neq j$ the kernel matrix $K(i, j) = 0$ if and only if its inverse $K^{-1}(i, j) = 0$.

Proof. Under Assumption 2.5 the matrix $K \in \mathbb{R}^{N \times N}$ is square and symmetric $K = K^\top$. Decompose K as $U\Sigma U^\top$ where $\Sigma \in \mathbb{R}^{N \times N}$ is a diagonal matrix with singular values and $U = [u_1^\top \ u_2^\top \ \dots \ u_N^\top]^\top \in \mathbb{R}^{N \times N}$ is unitary such that the row-vectors $u_i \in \mathbb{R}^{1 \times N}$ are orthogonal, i.e., $\langle u_i, u_j \rangle = \delta_{ij} \ \forall i, j$ where δ_{ij} is the Kronecker delta and $\langle \cdot, \cdot \rangle$ is the inner product defined over $\text{span}\{u_1, u_2, \dots, u_N\}$. Furthermore, $U^{-1} = U^\top$ yielding that the inverse $K^{-1} = (U^\top \Sigma U)^{-1} = U \Sigma^{-1} U^\top$.

To show that $K(i, j) = u_i \Sigma u_j^\top = 0$ if and only if $K^{-1}(i, j) = u_i \Sigma^{-1} u_j^\top = 0$ the following property must hold

$$u_i \in \text{Ker}(\Sigma u_j^\top) \Leftrightarrow u_i \in \text{Ker}(\Sigma^{-1} u_j^\top) \quad (2.33)$$

for all $(i, j) \in \{1, 2, \dots, N\}$ except for $i = j$, which holds true since U is unitary and by using that $\text{Ker}(u_j) = \text{Ker}(\Sigma u_j)$, see, e.g., Bernstein (2009, p.115), which completes the proof. \square

Lemma 2.17. *In the setting in Figure 2.5 and under Assumption 2.5, then with the kernel (2.29) where $l \rightarrow 0$, $N = pT \in \mathbb{N}$ and $p \in \mathbb{N}$, the FIR filter $M_{\mathcal{GP}}$ in (2.18) is of the form*

$$M_{\mathcal{GP}} = \sum_{i=1}^p w_i z^{-(iN-n_l)} \quad (2.34)$$

with weights $w_i \in \mathbb{R}$.

Proof. If $l \rightarrow 0$ and $N = pT$ then with the same reasoning as in Theorem 2.13 it can be shown that $K = \kappa(X, X)$ has non-zero values on the diagonal, all N^{th} off-diagonals and is zero elsewhere, i.e.,

$$K = \begin{cases} K(i, i + kN) \neq 0, \\ K(i + kN, i) \neq 0, \\ 0 \text{ elsewhere,} \end{cases} \quad (2.35)$$

with $k, i \in \mathbb{N}$ which is the same structure as $(K + \sigma_n^2)^{-1}$ using Lemma 2.16. Furthermore, the vector $K_*(i) \neq 0$ for $i = N - n_l + 1$ and zero elsewhere. Then, $\mu^{\mathcal{GP}}$ in (2.16) is of the form

$$\mu^{\mathcal{GP}} = [0_{N-n_l+1} \ w_1 \ 0_{N-1} \ w_2 \ \dots \ 0_{N-1} \ w_N], \quad (2.36)$$

which implies that $M_{\mathcal{GP}}$ is equal to (2.34) which completes the proof. \square

Lemma 2.17 shows that GPRC recovers the same structure as HORC, with weights w_i for $i = 1, 2, \dots, p$ that depend on the kernel and hyperparameters. The following Sections 2.5.3.1 and 2.5.3.2 illustrate that noise-robust RC and period-time robust RC in Steinbuch et al. (2007) are closely recovered with a suitable kernel function.

2.5.3.1 GPs for period-time robust RC: A form of HOCR improves robustness for uncertain period times, which is recovered by GPRC through a locally periodic kernel, that allows for slight variations in the disturbance estimate and is given by

$$\kappa_{LP}(t, t') = \exp\left(-\frac{(t - t')^2}{2l_s^2}\right) \kappa(t, t'), \quad (2.37)$$

where κ in (2.29) is the periodic kernel and l_s the local smoothness. The following example shows that noise robust HOCR is closely recovered by HOCR with a locally periodic kernel.

Example 2.18. A GPRC is designed with a buffer size $N = 3T$ where $T = 20$ and a locally periodic kernel (2.37) with hyperparameters $T = 20$, $\sigma_f = 1$, $l \rightarrow 0$, $l_s = 225$ and $\sigma_n^2 = 10^{-6}$ yielding weights

$$(w_1, w_2, w_2) = (2.93, -2.92, 0.98) \quad (2.38)$$

in Lemma 2.17 that closely resemble the weights obtained in Steinbuch et al. (2007). The modifying sensitivity S_R is shown in Figure 2.8 for GPRC (—) and HOCR (--) which are almost identical and significantly improve disturbance rejection for a wide range compared to traditional RC (- -).

Hence, GPRC closely recovers period-time robust RC in Steinbuch et al. (2007) using a suitable kernel function with a specific smoothness.

2.5.3.2 GPs for noise robust RC: GPRC can improve noise robustness with respect to traditional RC by using smoothness $l > 0$ in a periodic kernel, even outperforming noise-robust HOCR with a smaller buffer size. Noise robust RC as in Steinbuch et al. (2007) is recovered using a periodic kernel without smoothness $l \rightarrow 0$. This is illustrated in the following example.

Example 2.19. A GPRC is designed using the periodic kernel (2.29) without smoothness $l \rightarrow 0$ and $T = 20$, $\sigma_f = 1$ and $\sigma_n = 10^{-7}$. The buffer Γ contains $N = 3T$ samples. This results in the weights

$$(w_1, w_2, w_2) = (0.48, 0.33, 0.19), \quad (2.39)$$

as in Lemma (2.17), that closely resemble the weights for noise-robust HOCR in Steinbuch et al. (2007). Moreover, with the same periodic kernel where now smoothness is included $l = 100$ and the buffer size is reduces to $N = T$ samples, then noise robust HOCR is outperformed (—).

The resulting modifying sensitivities are shown in Figure 2.8 where noise-robust GPRC without smoothness (—) recovers noise-robust HOCR (--). By employing the extended design freedom in GPRC, i.e., using smoothness, then, even with a smaller buffer size (—), it outperforms HOCR due to averaging over potentially all N samples.

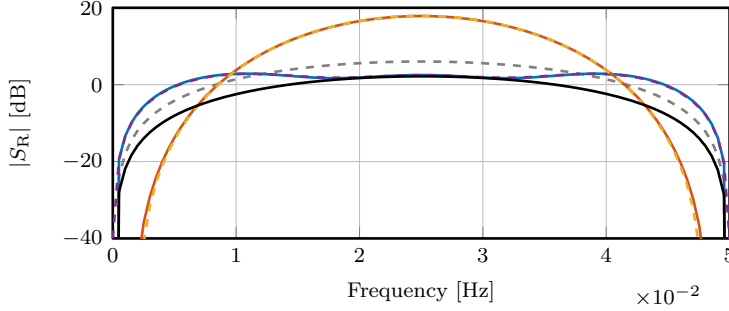


Figure 2.8: Modifying sensitivity S_R with traditional RC (--) as baseline. HORC in Steinbuch et al. (2007) for noise robustness (- -) with $N = 3T$ is recovered by GPRC a periodic kernel (—). Also HORC for period variations (- -) is recovered with a locally periodic kernel (—). Introducing smoothness (—) outperforms noise-robust HORC with a smaller buffer size $N = T$.

Examples 2.19 and 2.18 show that HORC is recovered without smoothness and an appropriate kernel, furthermore, introducing smoothness yields additional design freedom to improve noise robustness with a much smaller buffer size than HORC. However, including smoothness also leads to less disturbance attenuation at high frequencies as shown in Figure 2.9.

2.5.4 GPs for multi-period RC

The periodic kernel (2.28) in Section 2.4.2 also enables rejection of multi-period disturbances. Using a multi-period kernel GPRC suppresses the disturbance at specific frequencies instead of all harmonics of the common multiple, resulting in less amplification of non-periodic errors similar to Blanken et al. (2020a) or Griño and Costa-Castelló (2005) where only odd frequencies are rejected. This is illustrated in the following example.

Example 2.20. *The modifying sensitivity with a multi-period kernel where $n_d = 2$, $T_1 = 20$, $T_2 = 15$, and $l \rightarrow 0$ is shown in Figure 2.10 for buffer size $N = T_1 + T_2 = 35$ samples (—) and for a larger buffer size $N = \text{lcm}(T_1, T_2) = 60$ (—).*

Figure 2.10 shows that only disturbances with fundamental frequencies $1/T_1$ and $1/T_2$ are suppressed, compared to traditional RC with $N = 60$ samples (—) that yield unnecessary disturbance suppression at $1/T$ and harmonics. The FIR coefficients μ^{GP} in (2.18) are given in Figure 2.11, which are non-zero at the multiples of T_1 and T_2 and the difference between both.

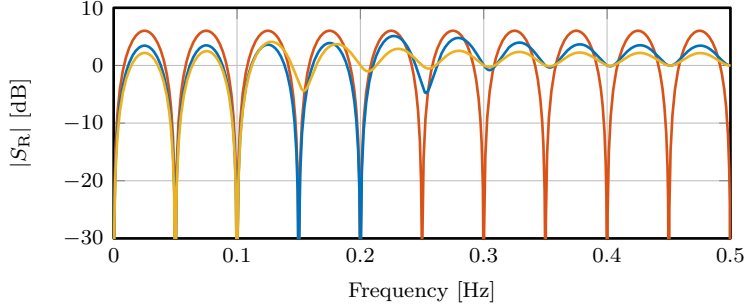


Figure 2.9: Modifying sensitivity with GPRC for a periodic kernel with three different levels of smoothness $l = 10^{-6}$ (—), $l = 10$ (—) and $l = 100$ (—), showing that smoothness improves robustness for noise and reduces disturbance rejection at higher frequencies.

Example 2.20 illustrates that by only introducing disturbance suppression where this is required, less amplification of noise at intermediate frequencies is obtained, due to Bode's Sensitivity integral.

Remark 2.21. *Example 2.20 shows that a buffer size of $N = T_1 + T_2$, as in Remark 2.11 is sufficient to suppress the disturbance, with more data the robustness with respect to noise is improved by averaging out over multiple samples.*

Remark 2.22. *If uncertain period times, noise and multi-period disturbances appear at the same time, then a sum of locally periodic kernels K_{LP} in (2.37) can be used. In this case, l_s acts tuning parameter for the trade-off between noise robustness or period-time uncertainty. Specifically, if l_s is large, then κ_{LP} has more emphasis on noise robustness, i.e., $\lim_{l_s \rightarrow \infty} \kappa_{LP} = \kappa$, and if l_s is small then period uncertainties are more taken into account.*

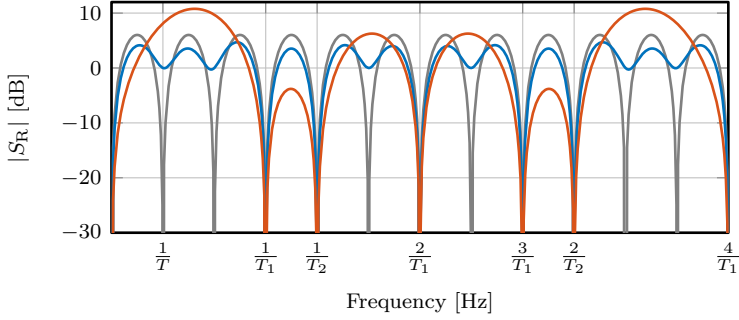


Figure 2.10: Modifying sensitivity S_R for multi-period GPRC with $n_d = 2$, $T_1 = 20$ and $T_2 = 15$ samples with buffer size $N_1 = T_1 + T_2 = 35$ samples (—) and $N_2 = \text{lcm}(T_1, T_2) = 60$ samples (—). As a comparison, the traditional RC with $N = 60$ is also shown (—).

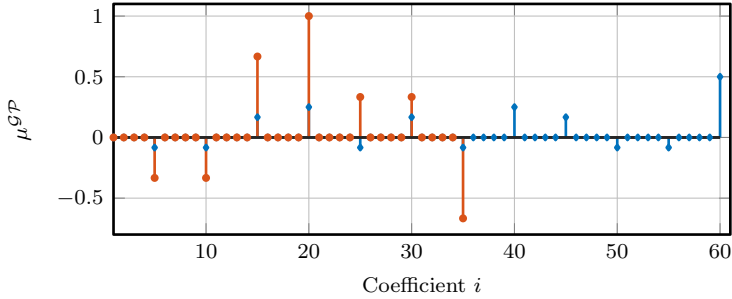


Figure 2.11: FIR coefficients $\mu^{\mathcal{GP}}$ with a multi-period kernel where $n_d = 2$, $T_1 = 20$, $T_2 = 15$, yielding non-zero FIR coefficients $\mu^{\mathcal{GP}}$ at T_1 , T_2 and the difference between them for $N_1 = 35$ (○) and with $N_2 = 60$ (◇).

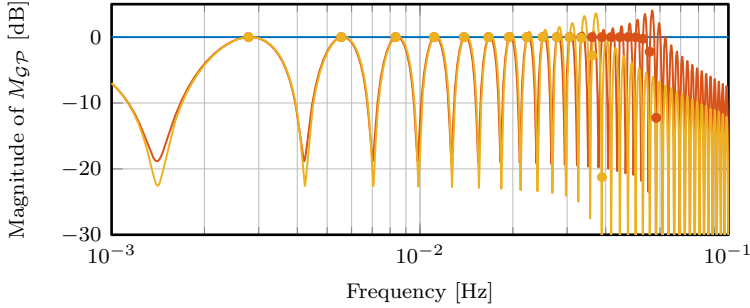


Figure 2.12: Magnitude of $M_{\mathcal{GP}}$ for a periodic kernel with $n_d = 1$ and $T = 360$ and three settings of smoothness, i.e., $l = 10^{-3}$ (—), $l = 0.5$ (—) and $l = 1$ (—). The markers (●) and (●) indicate fundamental frequencies $f = \frac{1}{T}$ and harmonics (top plot). This shows that smoothness leads to a low-pass characteristic in $M_{\mathcal{GP}}$, which in turn increases robustness.

2.5.5 Robustness for model errors

Robustness for model errors in RC is often improved by designing a robustness filter Q , typically a low-pass filter, that is placed in series with the buffer $M_{\mathcal{GP}}$. Next, it is shown that robustness is naturally included in GPRC by increasing smoothness. Theorem 2.7 provides a non-conservative stability condition where $M_{\mathcal{GP}}$ has a similar role as the traditional Q filter in RC, see, e.g., Hara et al. (1988); Steinbuch (2002).

If smoothness $l \rightarrow 0$ and $n_d = 1$, then by Theorem 2.13 $M_{\mathcal{GP}} = z^{-(N-n_l)}$ which has magnitude $|M_{\mathcal{GP}}(e^{j\omega})| = 1 \forall \omega$, see (—) in Figure 2.12. To improve robustness, the buffer $M_{\mathcal{GP}}(e^{j\omega}) < 1$ for the frequencies where model errors are present. In Figure 2.12 $M_{\mathcal{GP}}$ is given for $l = 1$ (—) and $l = 0.5$ (—) resulting in a low-pass characteristic which increases robustness for high-frequency modeling errors.

From an intuitive point of view, higher smoothness yields a smoother disturbances estimate \hat{d}^c , and thereby less high-frequency content in the RC output \hat{d}_μ . Hence, learning is limited in the high-frequency range, i.e., where the model is not reliable, having a similar effect as a Q filter in traditional approaches. Hence, smoothness also imposes an upper bound on the frequency content of the disturbance that can be learned.

Remark 2.23. The markers (●) and (●) in Figure 2.12 indicate the magnitude of $M_{\mathcal{GP}}(e^{j\omega})$ for $\omega = \frac{2\pi}{T}$ and its harmonics. In between these frequencies the magnitude of $\mu^{\mathcal{GP}}\Gamma$ is small, hence disturbances at those frequencies are filtered out. Note that $M_{\mathcal{GP}} > 1$ for some frequencies, which is allowed as long as Theorem 2.7 is satisfied.

Remark 2.24. *GPRC can be extended with a robustness filter Q if desired, i.e., $M_{GP}(z) \rightarrow Q(z)M_{GP}(z)$ to satisfy the stability condition (2.22).*

2.6 Implementation aspects: dealing with initial conditions

The previous sections establish an LTI framework for RC synthesis using GPs. In this section, performance in the first N samples is improved even further by taking into account the initial conditions of the buffer Γ , which may limit performance in the LTI case. Two solutions are provided to avoid this.

2.6.1 Limitations of the LTI case

The problem that arises in the LTI case is that the initial condition of the buffer Γ , which is zero by default, appears as observations of the disturbance in the training data set \mathcal{D}_N during the first N samples. Performing GP regression with these incorrect observations gives a worse disturbance estimate. After the first N samples, the initial condition of Γ disappears from the buffer. To improve GPRC in the first N samples, the following two solutions are provided.

2.6.2 Discarding observations

A simple solution is to discard the first N observations from the data set \mathcal{D}_N that correspond with the initial conditions of the memory Γ . This is done by introducing a time-varying selection matrix Ξ_k such that $w(k) = \Xi_k \Gamma \in \mathbb{R}^{\bar{N}(t)}$ where

$$\Xi_k = \begin{bmatrix} I_{\bar{N}(t)} & 0_{\bar{N}(t) \times (N - \bar{N}(t))} \end{bmatrix} \in \mathbb{R}^{\bar{N}(t) \times N} \quad (2.40)$$

with $\bar{N}(t) \leq N$ the time-varying number of samples that are used for GP regression. After N samples $\bar{N}(t) = N$ thus $\Xi_k = I_N$ such that the LTI case in Theorem 2.6 is recovered.

Note that this approach requires computing (2.16) at each sample during the first N samples, which is computationally demanding. Therefore, an alternative solution is introduced next.

2.6.3 Time-varying kernel to improve learning

A second solution is to choose a sufficiently high noise variance $\sigma_r^2 \gg \sigma_n^2$ for the undesired inputs such that these are reflected less in the RC output. This can

be done by modifying the matrix $(K + \sigma_n^2 I_N)$ in (2.16), by replacing the diagonal matrix with noise variances $\sigma_n^2 I_N$ with the following time-varying diagonal matrix

$$K_v^k = S_k \sigma_r^2 + (I_N - S_k) \sigma_n^2 \quad (2.41)$$

where

$$S_k = \begin{bmatrix} 0_{\Delta} & 0 \\ 0 & I_{\bar{N}-\Delta} \end{bmatrix} \quad (2.42)$$

is a selection matrix in which

$$\Delta = \begin{cases} 0 & \text{if } k \leq n_l, \\ N & \text{if } k - n_l \geq N, \\ k - n_l & \text{otherwise,} \end{cases} \quad (2.43)$$

such that after N samples $S_k = 0_N$ and the LTI case is recovered.

The time-varying matrix K_v^k is diagonal with noise variance σ_r^2 for GP inputs that correspond to the initial condition of Γ , and the variance is σ_n^2 for the GP inputs that represent the disturbance. In this way, the observations with a large variance have negligible influence on the posterior mean (2.15a), resulting in a significant improvement in convergence during the first N samples if smoothness is included as shown in Section 2.7.

Remark 2.25. *Both solutions lead to a time-varying system in the first N samples and are equivalent to the LTI repetitive controller in Theorem 2.6 after N samples.*

2.7 Generic case study

In this section, a simulation case study is performed in the most general case, i.e., a multi-period disturbance with a rational period-time, in presence of noise, and a very large common multiple such that traditional RC methods are not directly applicable. In addition, the effect of increasing smoothness and model uncertainties is illustrated.

2.7.1 System and disturbance

The case study is performed in the setting in Figure 2.1, where P is a discrete-time second order mass-spring-damper system given by

$$P(z) = \frac{0.05(z+1)}{z^2 - 1.99z + 0.99}, \quad (2.44)$$

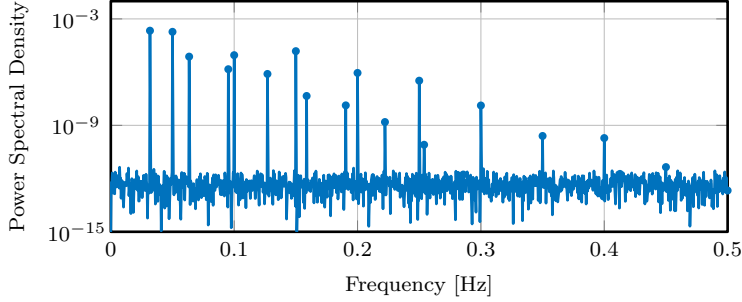


Figure 2.13: Power spectral density of the multi-period disturbance, containing two fundamental periods $T_1 = 20$ samples and $T_2 = 31.5$ samples.

see Figure 2.16, for which a stabilizing PD controller is designed

$$C(z) = \frac{5.0047(z+1)(z-0.8104)}{(z-0.5171)(z+0.02961)}, \quad (2.45)$$

resulting in a 0.1 Hz bandwidth.

A multi-period disturbance $d^c(t)$ is present that contains two fundamental periods $T_1 = 20$ samples and $T_2 = 31.5$ samples such that the common multiple is very large $T = 1260$ samples, also i.i.d. Gaussian distributed noise with $\sigma_n = 10^{-3}$ is added to the disturbance. A PSD of the disturbance is depicted in Figure 2.13.

Remark 2.26. *GPRC is also applicable if there is no common multiple, i.e., to non-periodic disturbances. In this case study, a large common multiple is chosen to compare GPRC steady-state performance with traditional RC as a benchmark.*

2.7.2 GPRC design

The learning filter L is designed as (2.23) using ZPETC resulting in a causal filter L_c and a non-causal part with $n_l = 1$ sample preview.

The corresponding hyperparameters with the multi-period kernel (2.28) are $n_d = 2$, $T_1 = 20$, $T_2 = 31.5$, including two different levels of smoothness $l_{1,2} = 1$ and $l_{1,2} = 3$, and $\sigma_f = 1$ and $\sigma_n = 10^{-3}$ are kept constant. The buffer Γ is implemented with two buffer sizes $N_1 = T_1 + T_2 = 52$ samples and $N_2 = 2N_1 = 104$ samples to illustrate the effect of including more data.

Simulations are conducted with PD control only, PD with LTI GPRC in Theorem 2.6 with the varying prior variance (2.41) in the first N samples where $\sigma_r = 10^3$. As a comparison, traditional RC with a buffer size $N = 1260$ is also implemented.

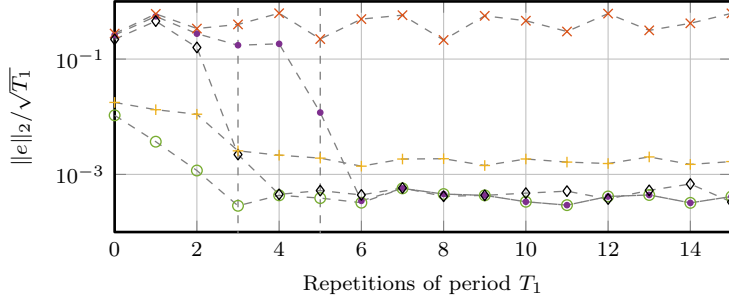


Figure 2.14: Error 2-norm as function of T_1 with PD (\times). GPRC with $N_1 = 52$ in (\diamond) that converges after $3T_1$, and with buffer size $N_1 = 104$ (\bullet) GPRC converges after $5T_1$. With a time-varying prior noise variance (2.41) the error (\circ) convergence is much faster ($l = 1$), converges to a larger error with a larger smoothness ($l = 3$) ($+$).

2.7.3 Results

The 2-norm of the error computed over the fundamental period T_1 is shown in Figure 2.14, steady-state performance is analyzed using the power spectral density (PSD) and cumulative power spectrum (CPS) of the converged error, see Figure 2.15. The following observations can be made.

- The contribution of the disturbance, i.e., the peaks in the PSD in Figure 2.15 of the error without RC (---), is fully rejected by GPRC (---), that has a buffer size ($N_1 = 52$) much smaller than the period time of the disturbance ($T = 1260$). The GPRC error-norm (\diamond) in Figure 2.14 significantly drops after $3T_1 \approx 52$ samples when sufficient observations are available (Remark 2.11) and the initial condition of Γ vanished from the buffer.
- Increasing the buffer size to $N_2 = 104 \ll T$ reduces the amplification of noise compared with buffer size N_1 (Section 2.5.3.2), yielding a lower cumulative error (---) than with N_1 (---), see Figure 2.15. As a consequence of the larger buffer size the initial condition of Γ vanishes after N_2 samples yielding slower convergence as shown by the error-norm (\bullet) in Figure 2.14 that drops after $5T_1 \approx 104$ samples.
- Convergence in the first N_2 samples is significantly improved by dealing with the initial conditions using a time-varying kernel (2.41) (\circ) compared with the LTI case (\bullet), see Figure 2.14. After N_2 samples both methods have the same error.
- Increasing smoothness from $l_{1,2} = 1$ to $l_{1,2} = 3$ essentially cuts-off learning

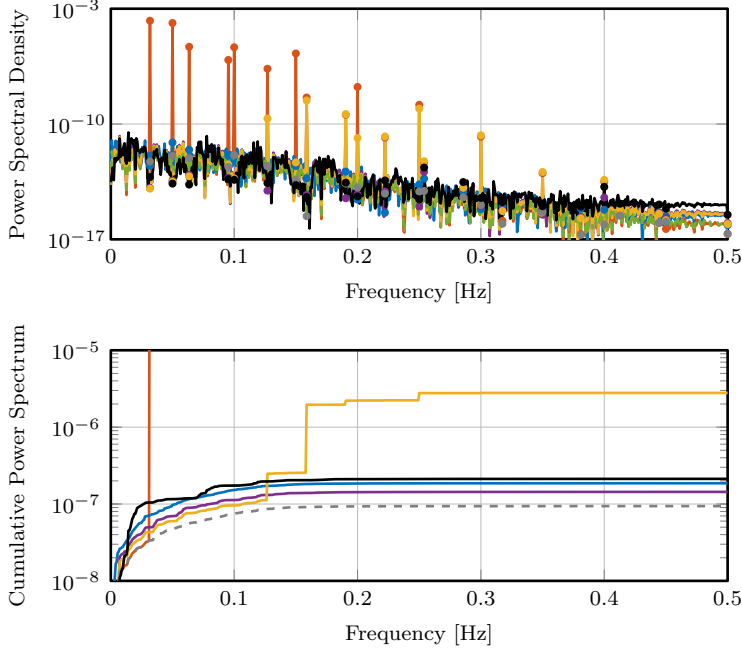


Figure 2.15: Power spectral density and cumulative power spectrum of the converged error with PD (—), GPRC with buffer size $N_1 = 52$ and $l_{1,2} = 1$ as a baseline (—) that fully rejects the disturbance. In addition, with buffer size $N_2 = 104$ (—) noise is amplified less, and with too much smoothness $l_{1,2} = 3$ (—) the higher frequency harmonics are not rejected sufficiently. As a comparison traditional RC with a very large buffer size $N = \text{lcm}(T_1, T_2) = 1260$ samples (—) is outperformed by GPRC and the lower bound is given by the noise induced error (—).

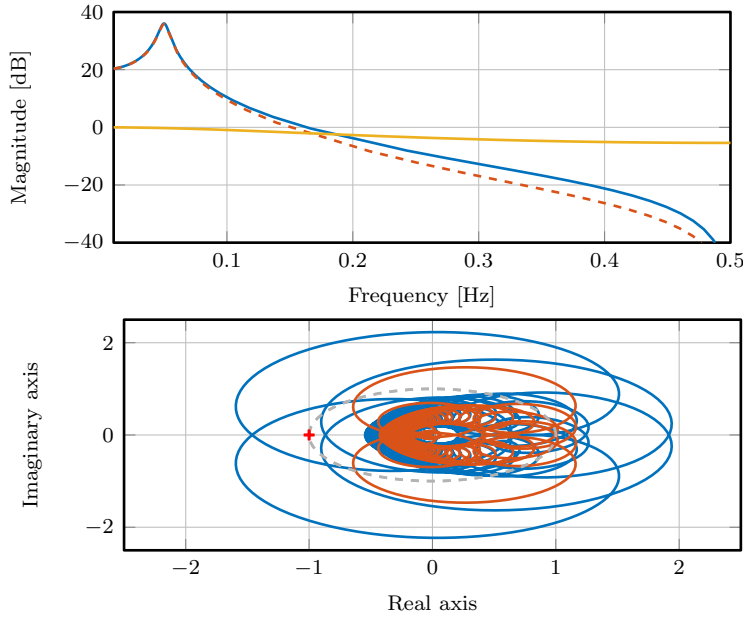


Figure 2.16: Top plot shows the true system P (—), a perturbed model (--) that is used for RC as well as the model mismatch (—) that is significant (up to $> 100\%$) beyond 0.1 Hz. The bottom plot shows the Nyquist stability test for smoothness $l_1 = l_2 = 10$ (—) that is unstable, by increasing the smoothness to $l_1 = 50$ and $l_2 = 10$ the RC is stabilized (—) since there are no encirclements of the point $-1 + 0i$.

in the high-frequency range. This reduces disturbance attenuation at high-frequencies as shown by (—●—) in Figure 2.15, and by (+) in Figure 2.14.

- A perturbed model with a significant mismatch in the high-frequency range is also used to compute the learning filter, see Figure 2.16. Increasing the smoothness allows to deal with this model mismatch, as discussed in Section 2.5.5, i.e., satisfy the stability condition in Theorem 2.7 as shown in the Nyquist plot in Figure 2.16. The effect of increasing smoothness is discussed in the previous topic. This effect is similar to using a low-pass robustness filter in traditional RC.
- Finally, GPRC (—) can outperform traditional RC (—) both in convergence as in steady-state error while using a significantly smaller buffer.

These observations confirm that GPRC is applicable to the general case, i.e., a multi-period disturbance with a rational period time and a very large common multiple, while using only a small buffer size and a limited number of intuitive design variables. Note that for this example T_1 and T_2 are selected such that a comparison with traditional RC can be made, i.e., a large common multiple exists. However, in general, the existence of a common multiple is not a restriction and GPRC is readily applicable to non-periodic disturbances as well. A major advantage of GPRC is a new way of designing the repetitive controller that, as shown here, naturally extends to for example multi-period disturbances.

2.8 Conclusions

A generic repetitive control framework for asymptotic rejection of single-period, and multi-period disturbances, with potentially rational period times, is enabled through a Gaussian process (GP) based internal model. The presented GP-based approach also enables compensation within the first period, in contrast to many existing RC approaches. The disturbance is modeled using GP regression, which is a non-parametric approach that combines data with prior knowledge. Prior knowledge is included in the form of a kernel function with periodicity and smoothness, which allows modeling a wide range of disturbances by specifying intuitive tuning parameters. It appears that under mild assumptions the GP-based RC approach is LTI and more specifically given by an FIR filter, such that it is computationally inexpensive, stability conditions can be provided and several existing approaches are recovered as a special case. Moreover, applying GP-based RC for non-linear systems is conceptually possible following the developments in this chapter by reformulating the stability conditions for the non-linear case which is a part of future research. Ongoing work focuses on utilizing the posterior variance of the disturbance model to improve robustness against model errors and incorrect prior.

CHAPTER

3

Gaussian Process Repetitive Control with Application to an Industrial Substrate Carrier System with Spatial Disturbances¹

Abstract: Repetitive control (RC) can perfectly attenuate disturbances that are periodic in the time domain. The aim of this chapter is to develop an RC approach, that compensates for disturbances that are time-domain non-periodic but are repeating in the position domain. The developed position-domain buffer consists of a Gaussian Process (GP) which is learned using appropriate dynamic filters. This approach estimates position-domain disturbances resulting in perfect compensation. The method is successfully applied to a substrate carrier system, demonstrating performance robustness against time-domain non-periodic disturbances, which are amplified by traditional RC.

¹The results in this chapter constitute Contribution 2 of this thesis. The chapter is based on Mooren, N. et al. (2022a). Gaussian Process Repetitive Control with Application to an Industrial Substrate Carrier Systems with Spatial Disturbances. (*Submitted for journal publication*), and preliminary results appeared in Mooren, N. et al. (2020b). Gaussian process repetitive control for suppressing spatial disturbances. *IFAC-PapersOnLine*, 53(2):1487–1492.

3.1 Introduction

Spatially periodic disturbances, including cogging, imbalances, eccentricity, and commutation errors, are encountered in many mechatronic applications (Chen and Chiu, 2008; Huo et al., 2019). These often appear non-periodic in the time domain, while these are reproducible in the spatial domain, e.g., with the (angular-)position (Li, 2015; Sun, 2004). In particular, these spatial disturbances appear non-periodic in the time domain for general tasks. In the case of repeating tasks, these appear periodic, which is a highly restrictive situation.

Repetitive control (RC) enables perfect attenuation of periodic disturbances through the internal model principle (IMP). The IMP (Francis and Wonham, 1976) states that a model of the disturbance generating system must be present in the stable feedback loop to enable asymptotic rejection. In traditional RC, a disturbance model is generated in a time-based fixed memory loop, allowing to attenuate disturbances that have a fixed time-domain period, see, e.g., Hara et al. (1988); Longman (2010); Tomizuka (2008). Only in the case of periodic tasks, i.e., if the position-domain disturbances appear periodic in time, RC is effective.

Repetitive control is not effective for disturbances that have a varying period or appear non-periodic in the temporal domain, due to inadequate temporal buffer (Hara et al., 1988; Steinbuch et al., 2007). In the case of non-periodic tasks, the resulting performance can be significantly degraded. Hence, a traditional time-domain memory loop is not effective for spatially periodic disturbances in the case of non-periodic tasks.

High-order RC (Pipeleers et al., 2008; Steinbuch et al., 2007) has been developed to increase flexibility for disturbances with uncertain period times by placing multiple delay lines in series. This allows optimizing a trade-off between variations in the period time on one hand, and robustness against non-repeating errors on the other hand. In Blanken et al. (2020a), multi-period RC is designed that allows for multiple disturbances with different periods, where, for time-domain non-periodic disturbances, a sum of periodic signals is selected a priori. Furthermore, a variety of adaptive repetitive control schemes, including, Chen and Tomizuka (2011); Hillerstrom (1996); Kurniawan et al. (2014); Manayathara et al. (1996) are developed, exploiting the adaptation of RC parameters, such as the sample time or buffer length, to cope with spatial disturbances. However, practical applications mainly operate at a fixed sample rate. While the above approaches also improve the performance for spatial disturbances with small task variations, they do not generalize to arbitrary task variations.

Spatial disturbances can be modeled efficiently in the position domain, where they appear periodic, in contrast to existing approaches where a time-based memory is employed in RC. Spatial disturbance models have been developed for RC, see, e.g., Chen and Yang (2007); Huo et al. (2019), where the disturbance is modeled as a non-linear potentially time-varying parametric model. Alterna-

tively, in Cao and Ledwich (2002); Mooren et al. (2020a); Yao et al. (2013) a discrete buffer is presented that contains position information to model the spatial disturbance. These discrete buffers require additional interpolation to deal with the inherently non-equidistant data points in the spatial domain. Hence, both spatial RC approaches require additional modeling effort or interpolation, which complicates their practical implementation.

Although improvements have been made to traditional RC to cover wider ranges of disturbances, including spatial disturbances, a systematic approach to generate spatial buffers efficiently with the non-equidistant observations in the spatial domain is not yet available. The aim of this chapter is to present a spatial RC approach that can compensate for position-domain disturbances by using a new type of spatial memory.

Beyond typical buffers in RC, i.e., by a discrete-time or continuous-time delay, recent developments have enabled modeling function from data, e.g., as a Gaussian Process (GP). The key idea in Gaussian Processes (GP) regression is to estimate a distribution over functions from data and prior knowledge, see, e.g., Murphy (2012); Williams and Rasmussen (2006) for a complete overview. This enables efficient interpolation of continuous disturbance models from finite measurement points (Bijl et al., 2015; Snelson and Ghahramani, 2007). The aim is to employ a GP-based disturbance model in a spatial memory loop for RC.

The main contribution in this chapter is a continuous GP-based spatial disturbance model in RC to enable attenuation of spatial disturbance for arbitrary task variations. This includes the following sub-contributions.

- C1 Developing a novel continuous spatial disturbance model, efficiently utilizing non-equidistant observations in combination with suitable prior knowledge, based on Gaussian Processes (GP).
- C2 Implementation of the GP-based disturbance model in a memory loop for spatial RC, including stability analysis, learning filter design, and prior selection.
- C3 A computationally efficient implementation of the GP-based spatial memory loop.
- C4 A simulation case study, with full GP, sparse GP and traditional RC.
- C5 Experimental validation on a substrate carrier system.

This chapter is outlined as follows. In Section 3.2, the problem setting and spatial disturbance are introduced. In Section 3.3, a continuous function of the spatial disturbance is identified as a Gaussian Process with suitable prior (C1). In Section 3.4, the spatial repetitive control framework is presented, including stability analysis, learning filter design, and integration of the spatial model in a spatial memory loop (C2). In Section 3.5, a computationally efficient implementation of GP regression is provided (C3). Finally, in sections 3.6 and 3.7,

simulations and experimental validation are carried out (C4-C5), and conclusions are presented in Section 3.8.

3.2 Problem formulation

In this section, an industrial relevant motivational application is presented, and the considered type of disturbance is defined. Furthermore, the control setting and spatial disturbance rejection problem is defined.

3.2.1 Application motivation

Disturbances that are repeating in the position domain appear in many industrial applications, e.g., due to imbalances or imperfections in rotary systems, or non-perfect commutation leading to position-dependent disturbances in positioning systems such as a wafer stage. In this chapter, the industrial substrate carrier in Figure 3.1 with schematic representation in Figure 3.2 is considered, and used for experimental validation.

The aim of the substrate carrier setup is to accurately position the point-of-interest on the substrate with respect to a print unit. This is enabled by fixating the substrate to a steel belt using a vacuum, consequently, the position of the steel belt is controlled in the three degrees-of-freedom (DoF) r_x , r_y and ϕ_z , see Beltman et al. (2012b); Beltman et al. (2014) for details. Positioning of the belt is performed with two actuated rollers for the longitudinal direction r_y , these rollers also contain 3 segments that are actuated in the lateral direction to control r_x and ϕ_z , see Figure 3.1 and 3.2. The position of the belt is measured with an encoder in r_y , and two sensors on the side of the belt measure r_x and ϕ_z . The focus in the remainder of this chapter is on the positioning performance in the lateral direction r_x .

The rotational nature of this system and slight imperfections in the rollers and its segments, induce a position-dependent disturbance that repeats every roller rotation. Depending on the belt velocity, which may vary over time, the disturbance can appear periodic or non-periodic in the time domain. The aim of this chapter is to reject the spatial disturbance, independent of roller velocity variations. Furthermore, the roller position in the experiment setup is measured very accurate and fast which is also considered in the remainder of the chapter.

3.2.2 Control problem

The control configuration is depicted in Figure 3.3, where the substrate carrier is denoted by P and assumed to be a Single-Input Single-Output (SISO) stable Linear Time Invariant (LTI) system, and C is a stabilizing feedback controller. The reference to be tracked is $r(t)$, which is zero in the industrial setup, and $y(t)$ is the position output corresponding to ϕ_z in the setup. A roller induced input

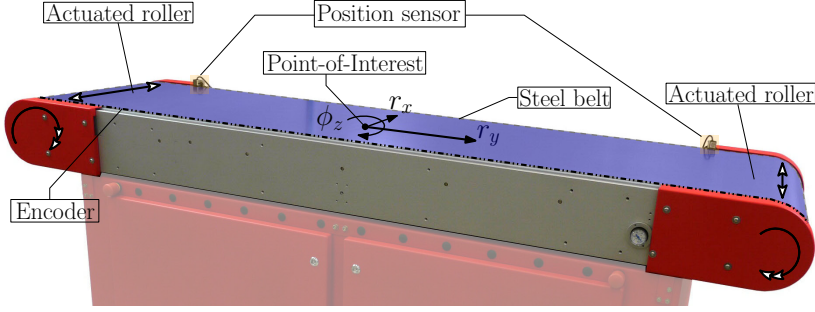


Figure 3.1: Industrial substrate carrier setup where to aim is to position the point-of-interest subject to spatially periodic disturbances. The belt is controlled using actuated rollers.

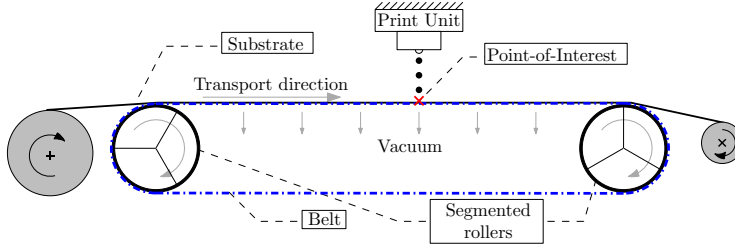


Figure 3.2: Schematic representation of the industrial substrate carrier setup.

disturbance $d(t)$ is present, that is generated by an unknown static position-domain mapping $\bar{d}(p) \in \mathbb{R}$, driven by an exogenous and known position signal $p(t)$, i.e., the roller position in the considered experimental setup. Note that the position signal $p(t)$, and the system output $y(t)$ are not related in this setting. Furthermore, $\eta(t)$ represents measurement noise that is independent and identically distributed Gaussian zero mean with variance σ_η^2 . Throughout the chapter, $t \in \mathbb{R}$ is continuous-time, $k \in \mathbb{Z}$ is discrete-time, and the notation $\bar{\cdot}$ indicates a spatial signal. The aim is formulated in the following definition.

Definition 3.1. *Define the measured tracking error*

$$e_m(t) = r(t) - y(t) - \eta(t), \quad (3.1)$$

in the presence of a position-domain disturbance $d(t)$, reference $r(t)$ and measurement noise $\eta(t)$. Assume that, the frequency content of $r(t)$ and $d(t)$ are significantly different from the frequency content of $\eta(t)$, such that the aim is to minimize the disturbance-induced and reference-induced errors.

The disturbance- and reference-induced errors can be written as

$$e = S(r - Pd) = e_r + e_d, \quad (3.2)$$

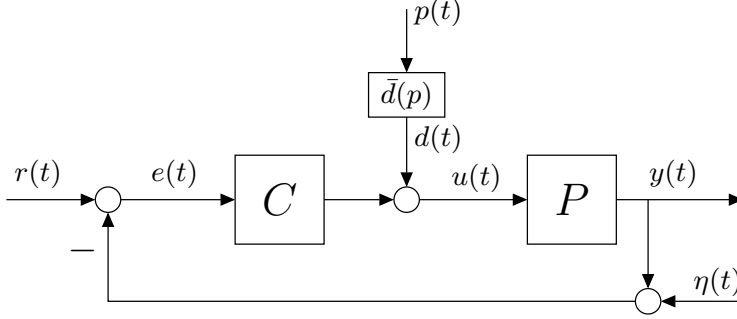


Figure 3.3: Spatial repetitive control problem with input disturbance $d(t)$.

3

with $S = \frac{1}{1+PC}$ the sensitivity function, and e_d and e_r the disturbance induced error and reference induced error respectively. This chapter focuses on the former, by making the following assumption.

Assumption 3.2. *The reference induced error $e_r(t)$ is negligibly small after a certain (transient) time t_0 , i.e.,*

$$|e_r(t)| \leq \beta \quad \forall t \geq t_0 \quad (3.3)$$

with $\beta \in \mathbb{R}$ a sufficiently small constant. This can be obtained by a suitable design choice of C , and possible feedforward control, i.e., $C_{\text{ff}} = P^{-1}$ and $u_{\text{ff}} = C_{\text{ff}}r$ that is added to the input $u(t)$, see, e.g., Oomen (2019).

Hence, when $t \geq t_0$ we assume that $e(t) = e_d(t)$, so that the control goal reduces to the attenuation of the spatially periodic disturbance, as is outlined next.

3.2.3 Spatial disturbance

The disturbance $d(t)$ acting on the system may appear periodic or non-periodic in time. However, it has an equivalent counterpart $\bar{d}(p)$ that is repeating in the spatial domain. Hence, the roller induced disturbance in the substrate carrier system is modeled as a static but unknown function of position, defined as follows.

Definition 3.3. *The disturbance $d(t)$ is composed from the exogenous and known position signal $p(t)$, and the unknown static spatial disturbance function $\bar{d}(p)$, i.e.,*

$$d(t) = \bar{d}(p(t)) \quad (3.4)$$

where $\bar{d}(p)$ is periodic in the spatial domain, i.e.,

$$\bar{d}(p) = \bar{d}(p + n \cdot p_{\text{per}}), \text{ for } n \in \mathbb{N} \quad (3.5)$$

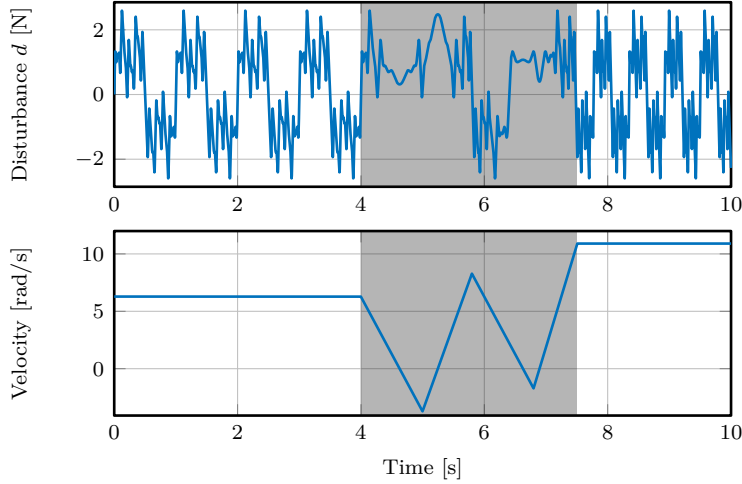


Figure 3.4: Simulation example of a position-domain disturbance (top plot), with corresponding velocity of $p(t)$ (bottom plot) as function of time. The gray area indicates where the velocity changes, which causes the disturbance to become non-periodic in the time domain.

with $p_{\text{per}} \in \mathbb{R}$ the spatial period.

Remark 3.4. For this specific application in RC, the spatial disturbance function \bar{d} is assumed to be periodic. However, this is not required in the general case.

Remark 3.5. In contrast to other existing approaches (Sun, 2004; Sun et al., 2009), the exogenous position signal $p(t)$ is not equal to the plant output; also, the spatial disturbance is a static function of position, i.e., not a dynamical model, which removes the need to convert the system dynamics to the position domain, resulting in a time-varying system.

The spatial disturbance leads to two major challenges. First, if $p(t)$ is periodic then $d(t)$ is periodic. However, if $p(t)$ is non-periodic, then $d(t)$ is in general also non-periodic. This implies that non-periodicity of $p(t)$ leads to $d(t)$ being non-periodic in the time domain. For instance, if $\dot{p}(t)$ is (piecewise) constant, then $d(t)$ is (piecewise) periodic in time, and, if $\dot{p}(t)$ varies then $d(t)$ is non-periodic in time, see Figure 3.4. Second, samples of the position-domain disturbance are in the general case non-equidistant in the spatial domain. The sampled position signal is denoted by $p(t_k)$ where $t_k = kT_s$ with T_s the sample time and $k \in \mathbb{N}$, which are equidistant in time. As a consequence, the corresponding spatial

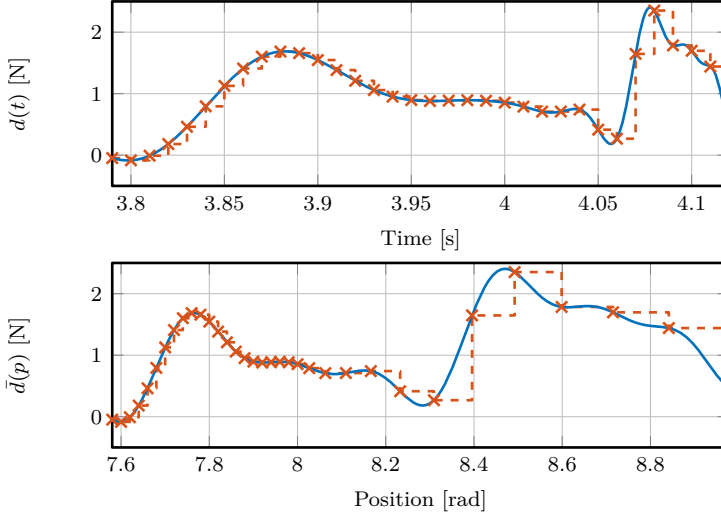


Figure 3.5: Simulation example: the top plot shows a continuous disturbance as a function of time $d(t)$ (—) from which equidistant samples (×) are taken with a fixed sampling frequency. The bottom plot shows the same disturbance as a function of position yielding a different shape, where, due to velocity variations, the same samples that were equidistant in time are now non-equidistant in position (×).

samples are non-equidistant, i.e.,

$$p(t_k) - p(t_{k+1}) \neq p(t_i) - p(t_{i+1}) \text{ for some } i, k \in \mathbb{N} \quad (3.6)$$

which implies that observations of $\bar{d}(p)$ are non-equidistant in the spatial domain; see Figure 3.5 where $d(t)$ and several samples that are equidistant in time are shown both in the time and position domain where they are non-equidistant. This leads to major challenges for writing to and reading from the buffer, which are both essential aspects in RC.

3.2.4 Problem definition

The aim of this chapter is to reject the spatially periodic disturbance invariant under velocity variations in $p(t)$. This is established in the spatial RC framework in Figure 3.6, where, analog to traditional RC, the internal model principle is applied by learning a continuous disturbance model through a time-domain filter. The crucial difference is that a disturbance model is learned in a spatial memory-loop instead of a temporal one in traditional RC. The disturbance model is learned from non-equidistant observations \bar{y}_d , and predicted at other

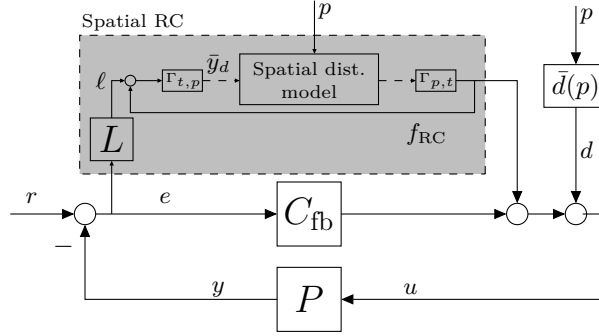


Figure 3.6: Spatial repetitive control framework, where (---) represents a position-domain signal and (—) is a time-domain signal.

positions for compensation. To enable learning in the spatial domain, transformations between time-domain signals and position-domain signals are used, and a suitable time-domain learning filter L is developed. Next, in Section 3.3 the spatial disturbance model is derived, and in Section 3.4 the spatial RC framework is designed and analyzed.

3.3 Gaussian process spatial disturbance model

In this section, a continuous model of the spatial disturbance is identified from non-equidistant observations by means of Gaussian Process regression, constituting C1.

3.3.1 Identifying a spatial disturbance model

According to the internal model principle, the spatial buffer in Figure 3.6 must contain a model of $\bar{d}(p)$, that is learned from N training samples $\bar{d}(p_i)$ at non-equidistant positions $p_i \in \mathbb{R}$ for $i = 1, 2, \dots, N$. At the same time, the RC provides a control action by evaluating the disturbance model at another position $p_j \in \mathbb{R}$, which is in general not equal to the training samples. This implies that interpolation or extrapolation is required to estimate $\bar{d}(p_j)$ from data $\bar{d}(p_i)$.

GP regression enables estimation of $\bar{d}(p)$ as a continuous function from non-equidistant training data and suitable prior knowledge, essentially automating the interpolation that is required in other spatial approaches (Yao et al., 2013). The resulting GP-based disturbance model is a distribution over functions, i.e., a collection of random variables in the position domain determined by a mean and variance, see Figure 3.7 for an illustrative example. Loosely speaking, GP regression is a projection of data on a set of basis functions, that do not need to be defined explicitly, which is a major advantage over parametric estimation

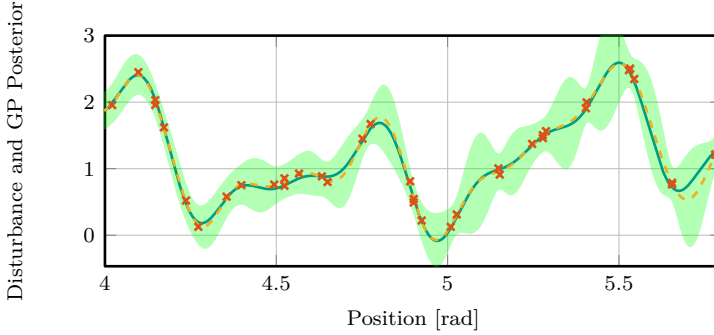


Figure 3.7: GP regression example, with the disturbance (—) from which noisy observations (×) are generated. With these observations the GP posterior mean (---), and variance () are generated.

methods. Instead, a class of a potentially infinite set of basis functions is defined in the form of a kernel function; it will become clear in Section 3.4 how to design the kernel function for spatial RC. Furthermore, GP regression can be split up in two key steps, a training step and a prediction step by performing inference, these are considered in the remainder of this section and used in the spatial RC framework in Section 3.4.

Remark 3.6. *In contrast to existing spatial RC approaches, the non-equidistant data is used to estimate a distribution over (periodic) function, whereas traditional approaches either require interpolation in a discrete buffer (Cao and Ledwich, 2002; Mooren et al., 2020a; Yao et al., 2013) or require to reformulate the systems in the spatial domain resulting in time-varying dynamics (Chen and Chiu, 2008; Sun, 2004). This chapter presents a systematic design approach for continuous buffers while also removing the need for complex time-varying models.*

Next, consider the disturbance function that is assumed to generate the data. Thereafter, a suitable disturbance estimation for spatial RC is formulated as a GP regression problem, see Section 3.4.

3.3.2 Spatial disturbance function and training data set

Observations of the spatial disturbance \bar{y}_d are subject to independent and identically distributed zero-mean Gaussian noise with variance σ_n^2 , i.e.,

$$\bar{y}_d(p_k) = f(p_k) + \epsilon, \text{ with } \epsilon \sim \mathcal{N}(0, \sigma_n^2), \quad (3.7)$$

where $p_k \in \mathbb{R}$ is the position at which observation $\bar{y}_d(p_k)$ is taken. Note that ϵ is the result of measurement noise η , filtered by the closed loop and learning

filter, which remains normally distributed. It will be shown later in Section 3.4 that $f(p_k)$ approximates the actual disturbance function, i.e., $f(p_k) = \bar{d}(p_k)$. The function f is parameterized as a linear combination of a potential infinite number of basis functions and parameters

$$f(p_k) = \phi(p_k)^\top w, \text{ with } w \sim \mathcal{N}(0, \Sigma_p) \quad (3.8)$$

where $\phi(p_k)$ is a vector of basis functions and $\Sigma_p = \mathbb{E}(ww^\top)$ is the covariance matrix of the weights w . Finally, the training data set containing noisy observations and corresponding positions is denoted as $\mathcal{D} = (X, \mathbf{y}_d)$, with

$$X = \begin{bmatrix} p_1 & p_2 & \dots & p_N \end{bmatrix}^\top \in \mathbb{R}^N$$

$$\mathbf{y}_d = \begin{bmatrix} \bar{y}_{d,1} & \bar{y}_{d,2} & \dots & \bar{y}_{d,N} \end{bmatrix}^\top \in \mathbb{R}^N$$

consists of N training positions and observations, is used to train the GP.

Next, the training data \mathcal{D} is used to identify a continuous function f that represents the spatial disturbance \bar{d} , and allows to make predictions outside the training set.

3.3.3 Gaussian Process regression

In this subsection, a GP regression problem is formulated to model the continuous disturbance function from data \mathcal{D} and prior knowledge, i.e., training of the GP, and to make prediction at new position p_* by performing inference, for further details, see Williams and Rasmussen (2006). This separation is used later in Section 3.4 to establish the spatial memory loop.

First, define a set of N_* test positions $\mathbf{p}_* \in \mathbb{R}^{N_*}$ with corresponding function values $\mathbf{f}_* = f(\mathbf{p}_*)$, and the training points \mathbf{y}_d . Next, assume that \mathbf{f}_* and \mathbf{y}_d have a joint Gaussian distribution $p(\mathbf{y}_d, \mathbf{f}_*)$ given by

$$\begin{bmatrix} \mathbf{y}_d \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{XX} + \sigma_n^2 I_N & K_{X*} \\ K_{X*}^\top & K_{**} \end{bmatrix} \right), \quad (3.9)$$

where $K_{XX} \in \mathbb{R}^{N \times N}$ denotes the matrix of covariances evaluated at all pairs of training positions in X and similar for $K_{X*} \in \mathbb{R}^{N \times N_*}$ and $K_{**} \in \mathbb{R}^{N_* \times N_*}$ which also include test positions with $*$. The covariance matrix K is selected by the user, see Section 3.4.5, and also known as the kernel matrix that expresses prior knowledge on the function f to be estimated.

Predictions of f at a test point p_* from training data \mathcal{D} , are given by the conditional posterior distribution of f^*

$$p(f_* | p_*, \mathcal{D}) = \mathcal{N} \left(\hat{d}_\mu, P_{\text{post}} \right), \quad (3.10)$$

where

$$\hat{d}_\mu = K_{X*}^\top (K_{XX} + \sigma_n^2 I_N)^{-1} \mathbf{y}_d \quad (3.11a)$$

$$P_{\text{post}} = K_{**} - K_{X*}^\top (K_{XX} + \sigma_n^2 I_N)^{-1} K_{X*} \quad (3.11b)$$

are the posterior predictive mean and covariance respectively. The GP mean can be computed efficiently for a single test point p_* as

$$\hat{d}_\mu(p_*) = \sum_{i=1}^N \alpha_i \kappa(p_i, p_*), \quad (3.12)$$

where $\kappa(p_i, p_*)$ is the kernel function evaluated at training points and test point, and

$$\alpha = (K_{XX} + \sigma_n^2 I_N)^{-1} \mathbf{y}_d = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_N]^\top \quad (3.13)$$

is essentially the link between the trained GP and the prediction step, as will become clear later.

The posterior mean, i.e., (3.11a) or (3.12), involves inverting an $N \times N$ matrix that solely depend on the training data \mathcal{D} . Due to a lack of structure, the computational complexity of this inversion scales cubically with N . Note that this inversion, i.e., training of the GP, can be done independently of the prediction step, such that new predictions are less time-consuming. In addition, in Section 3.5 a sparse GP approximation is presented that reduces the computational effort.

Remark 3.7. *Inversion of the matrix $(K_{XX} + \sigma_n^2 I) \in \mathbb{R}^{N \times N}$ can be done more efficiently. Instead of inverting the kernel matrix completely at every training step, the inverse can be computed using the Sherman-Morrison formula, essentially using rank 1 updates on the previous inverse, e.g., similar to the work in Bijl et al. (2015). This reduces the computational complexity to $\mathcal{O}(N^2)$ for each training update instead of $\mathcal{O}(N^3)$.*

Remark 3.8. *If the position signal $p(k)$ in the training data X is not known sufficiently accurate then input noise can be taken into account for GP regression, see, e.g., Bijl et al. (2017); McHutchon and Rasmussen (2011).*

3.3.4 Prior selection

GP regression relies on data and suitable prior knowledge, the latter is essential to extrapolate the estimated model beyond the training points, which is key in the spatial RC framework. The function \bar{f} in (3.8) is defined by a set of basis function ϕ and parameters w . Instead of selecting these basis functions explicitly, the prior distribution (3.9) contains a covariance function that specifies a *class* of basis functions. Consequently, the prior spans a potentially infinite set of basis

functions, without explicitly defining them, which is a result known as Mercer's Theorem, see, e.g., (Williams and Rasmussen, 2006, Section 4.3).

To show this, the covariance of two training observations of the disturbance is written as

$$\begin{aligned} \text{cov}(\bar{y}_d^i, \bar{y}_d^j) &= \mathbb{E}[(\psi(p_i)^\top w + \epsilon_i)(\psi(p_j)^\top w + \epsilon_j)^\top] \\ &= \phi^\top(p_i) \Sigma_p \phi(p_j) + \delta_{ij} \sigma_n^2 \end{aligned} \quad (3.14)$$

such that

$$\text{cov}(\mathbf{y}_d) = \Phi^\top \Sigma_p \Phi + \sigma_n^2 I_N \quad (3.15)$$

and the mean of a training observation

$$\mu(\bar{y}_d) = \mathbb{E}[\phi(p)^\top w + \epsilon] = \phi(p)^\top \mathbb{E}[w] \quad (3.16)$$

is assumed to be zero, i.e., $p(\mathbf{f}) \sim \mathcal{N}(0, \Phi^\top \Sigma \Phi)$ with $\mathbf{f} = f(X)$.

Remark 3.9. GP regression can easily be extended for non-zero mean in the case that a prior estimate of the disturbance function is known, see, e.g., Murphy (2012).

By comparing the covariance of the observations in (3.15) with the joint prior in (3.9), it follows that the kernel expresses prior on the function \mathbf{f} , i.e., $K_{XX} = \Phi^\top \Sigma \Phi$, such that the covariance matrix (3.15) becomes

$$\text{cov}(\mathbf{y}_d) = K_{XX} + \sigma_n^2 I_N. \quad (3.17)$$

Details on how to choose the kernel matrix for the application in RC follow in Section 3.4.5.

Remark 3.10. It follows from Mercer's Theorem that any symmetric positive definite kernel matrix can be mapped into an inner product $K(p, p') = \langle \psi(p), \psi(p') \rangle$, with $\phi(p) = \Sigma_p^{\frac{1}{2}} \psi(p)$ a potentially infinite sum of basis functions, see, e.g., Murphy (2012); Williams and Rasmussen (2006). This enables to perform inference with an infinite set of basis functions while the kernel matrix is of size $N \times N$.

This completes the GP-based disturbance modeling, in the following section, the disturbance model is incorporated in the spatial RC framework.

3.4 Spatial Gaussian process repetitive control

In this section, the spatial GP-based RC as briefly introduced in Section 3.2 is further analyzed (contribution C2). This includes stability analysis, suitable learning filter design, integration of GP-disturbance model in a memory loop, and prior selection for the GP. Finally, a procedure is provided to implement spatial GP-based RC.

3.4.1 Spatial repetitive controller

The spatial repetitive controller indicated in gray in Figure 3.6 is depicted in detail in Figure 3.8, where the solid lines represent time-domain signals, and dashed lines are position-domain signals. The GP-based disturbance model is included in two blocks, i.e., GP_y that represents training of the GP by computing (3.13) from data \mathcal{D} , and $GP(p)$ is the prediction step at test point $p^* = p$ to generate the RC output f_{RC} . The link between the training and prediction step is given by (3.13) and indicated by the dotted line. Furthermore, transitions between time and position domain are indicated by $\Gamma_{t,p}$ and $\Gamma_{p,t}$ respectively, which are defined as follows.

Definition 3.11. The mappings $\Gamma_{t,p}$ and $\Gamma_{p,t}$ map a time domain signal $x(k)$ at sample k to the spatial signal $\bar{x}(p)$ and vice versa, i.e.,

$$\Gamma_{t,p} : x(k) \mapsto \bar{x}(\bar{p}), \quad \bar{x}(\bar{p}) = x(k) \quad (3.18a)$$

$$\Gamma_{p,t} : \bar{x}(p) \mapsto x(k), \quad x(k) = \bar{x}(p) \quad (3.18b)$$

where \bar{p} is the position at which the observation $x(k)$ is stored in \mathcal{D} , and p is the current position at sample k .

A stable learning filter L is present that filters the error to obtain the learning signal ℓ that is used to update the GP. Since the disturbance will be suppressed over time, the error e and consequently also ℓ will converge to zero. Hence, the learning signal ℓ cannot be fed directly to the GP. A memory element in the form of a feedback loop, i.e., similar to traditional RC, see, e.g., Hara et al. (1988), is present in the spatial RC, such that the identified model remains present in the loop. This constitutes the update

$$y_d(k) = \ell(k) + f_{\text{RC}}(k), \quad (3.19)$$

where $\ell = Le$, and f_{RC} is the RC output at sample k . Using Definition 3.11, the spatial observation is generated by transforming y_d to the spatial domain, i.e.,

$$\bar{y}_d(p) = y_d(k) \quad (3.20)$$

essentially including $y_d(k)$ at position $p(k)$ in the training data set \mathcal{D} . The spatial RC output f_{RC} is generated by evaluating the GP at test position $p(k)$ and converting to the time domain as in Definition 3.11, i.e.,

$$f_{\text{RC}}(k) = \bar{\mu}(p) \quad (3.21)$$

with $\bar{\mu}(p)$ the mean of the GP at position p , i.e., (3.12) with $p_* = p$.

Remark 3.12. Note that in classical RC the learning filter can be placed either before or after the buffer because of the commutative property of SISO LTI filters. In spatial RC, the commutative property does no longer holds and the position of the learning filter L before the buffer determines the interpretation of the GP, which is of crucial importance.

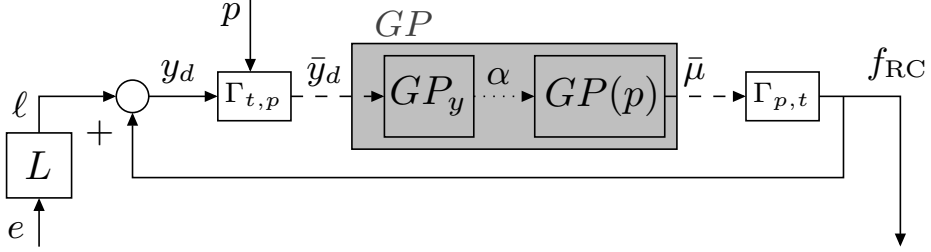


Figure 3.8: Spatial RC with GP-based memory loop, where (---) is a position-domain signal and (—) a time-domain signal (····).

In the remainder of this section, it is shown how to design L to ensure closed-loop stability, and such that \bar{y}_d represents an estimate of the disturbance, i.e., $\bar{y}_d(p) = \hat{d}(p)$.

3.4.2 Stability analysis

To analyze closed-loop stability with spatial RC, the input-output gain of the GP buffer is upper-bounded. This is a reasonable assumption since the GP will mainly take care of the interpolation/extrapolation of the data given a certain prior. Hence, if a perfect model is obtained, then the gain of the GP-buffer, i.e., from y_d to f_{RC} , is one. The mean of the GP is initialized as zero, hence if no data is available then the gain tends to zero.

Assumption 3.13. *The ℓ_2 -induced gain of the spatial GP-buffer is upper bounded as*

$$\|f_{RC}\|_{\ell_2} \leq \gamma \|y_d\|_{\ell_2}. \quad (3.22)$$

Assumption 3.13 implies that the output energy of the GP-buffer is less than or equal to γ times input energy, which is required to provide stability results for GP-based RC. In contrast to traditional RC where there is a delay z^{-N} that has a gain of one, the GP buffer depends on many variables, including the hyperparameters and the position p , and often $\gamma \neq 1$. Assumption 3.13 provides a condition to be tested on given a set of hyperparameters, rather than providing direct design guidelines for the hyperparameters, as there is not a direct relation between γ and the hyperparameters.

The following lemma is introduced before the main stability result is stated.

Lemma 3.14. *For the system interconnection in Figure 3.9, with*

$$\|y_1\|_{\ell_2} \leq \gamma_1 \|e_1\|_{\ell_2} \quad (3.23)$$

$$\|y_2\|_{\ell_2} \leq \gamma_2 \|e_2\|_{\ell_2} \quad (3.24)$$

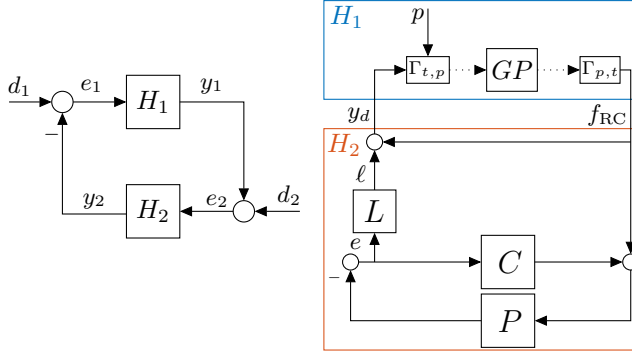


Figure 3.9: Standard feedback interconnection (left) and GP-based RC casted in the standard feedback interconnection (right).

and the inputs $\|d_1\|_{\ell_2}$, $\|d_2\|_{\ell_2} < \infty$, i.e., have finite ℓ_2 -induced norm. Then, for $\gamma_1\gamma_2 < 1$, the signals e_1 , e_1 , y_1 and y_2 have bounded ℓ_2 -induced norms, i.e., the interconnection is internally stable.

For a proof, see, e.g., Desoer and Vidyasagar (2009, Chapter 2).

Stability of the closed-loop is given by the following theorem.

Theorem 3.15. *The GP-based spatial RC is closed-loop stable under Assumption 3.13 if*

$$|1 - L(e^{j\omega})S(e^{j\omega})P(e^{j\omega})| < \frac{1}{\gamma} \quad \forall \omega \in [0, 2\pi]. \quad (3.25)$$

Proof. First, i) spatial RC framework is casted in the standard feedback interconnection as in Lemma 3.14, then ii) it is shown that if Theorem 3.15 is satisfied under Assumption 3.13, such that Lemma 3.14 is satisfied and the system is internally stable.

i) Set H_1 as the non-linear mapping from $y_d(k)$ to $f_{RC}(k)$, then H_2 is a linear system

$$y_d = H_2 f_{RC} - LSPd, \quad (3.26)$$

with $H_2 = 1 - LSP$, where LSP is stable since L is stable by design, and C stabilizes the feedback loop, i.e., SP is also stable.

ii) It remains to show that (3.25) satisfies Lemma 3.14 for some $\gamma_1\gamma_2 < 1$. Note that H_2 corresponds to the mapping $y_d \mapsto f_{RC}$ which is linear, thus the signal norm bound (3.24) is satisfied if

$$\|1 - LSP\|_{\infty} \leq \gamma_2 \quad (3.27)$$

for some $\gamma_2 < 1$ which is equivalent to (3.25) for SISO systems. From Assumption 3.13, it follows that $\gamma_1 \leq 1$, i.e., $\gamma_1\gamma_2 < 1$ and hence Lemma 3.14 is satisfied which completes the proof. \square

Condition (3.25) in Theorem 3.15 and additional robustness filter Q can be introduced at the output of the GP-based buffer, similar to traditional RC, see, e.g., Blanken et al. (2017b) for detailed design guidelines.

Remark 3.16. *Note that the stability condition in Theorem 3.15 can be tested on the basis of an identified frequency response function of the system, see, e.g., Blanken et al. (2020a).*

3.4.3 Learning filter design

From Theorem 3.15 it follows that if L is designed as the inverse of SP , then (3.25) is satisfied and closed-loop stability is obtained. In addition, recall from (3.2) that $d = -(SP)^{-1}e_d$, i.e., this learning filter design is also a sensible choice since the learning signal $\ell = Le = d$ generates an estimate of d to train the GP. Next, a procedure is outlined to design the learning filter.

Procedure 3.17 (Learning filter design).

1. Identify a parametric model \hat{P} of the system.
2. Compute an estimate of the process sensitivity

$$\widehat{SP} = \hat{P} (I + C_{fb}\hat{P})^{-1}. \quad (3.28)$$

3. Invert the process sensitivity estimate to obtain L

$$L = \widehat{SP}^{-1}. \quad (3.29)$$

Direct inversion of \widehat{SP} may lead to a non-causal or unstable inverse, i.e., if the plant contains non-minimum phase zeros or delays. By employing finite preview, a bounded inverse can be obtained, see, e.g., Van Zundert and Oomen (2018) and (Tomizuka, 1987), leading to

$$L = q^m L_c \quad (3.30)$$

where q is the forward time shift operator, L_c is the causal part of L , and n_l is the number of samples of preview in L .

In the following section, the RC scheme in Figure 3.8 is extended to implement the non-causal part of the learning filter.

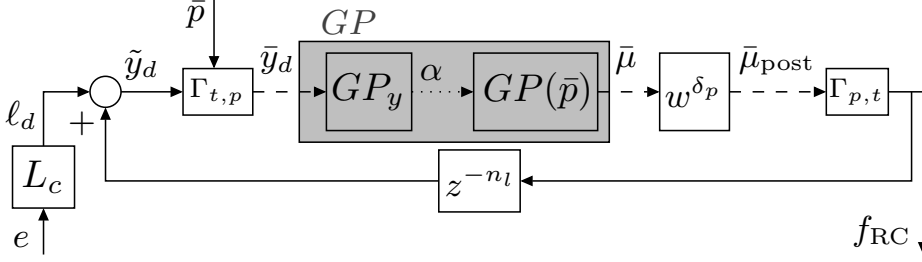


Figure 3.10: Spatial RC with GP-based memory loop including preview.

3.4.4 Incorporating spatial preview

In traditional RC, preview is incorporated in the time-domain buffer. This principle is extended towards spatial preview. A non-causal L filter can be implemented in spatial RC by filtering the error with the causal part of L , and implementing the non-causal part as a preview in the spatial memory.

To show this, define ℓ_d as the error filtered with the causal part of L

$$\ell_d(k) = L_c(q)e_d(k) = \hat{d}(k - n_l) \quad (3.31)$$

which corresponds to the disturbance estimate with n_l samples of delay. Substituting (3.31) in the update (3.19) gives

$$y_d(k) = q^{n_l} \ell_d(k) + f_{RC}(k), \quad (3.32)$$

which requires future values of ℓ_d . Multiplying both sides with q^{-n_l} gives the following causal update,

$$\tilde{y}_d(k) = \ell_d(k) + q^{-n_l} f_{RC}(k), \quad (3.33)$$

where $\tilde{y}_d(k) = q^{-n_l} y_d(k)$, this is schematically presented in the RC memory loop in Figure 3.10. Next, from (3.31), and by using that $q^{-n_l} f_{RC}(k) = \bar{\mu}(p(k - n_l))$, the observation \tilde{y}_d corresponds to the position $p(k - n_l)$. Hence, the spatial observation is \tilde{y}_d is included in \mathcal{D} at position $p(k - n_l)$, i.e.,

$$\bar{y}_d(\bar{p}) = \tilde{y}_d(k) \quad (3.34)$$

with $\bar{p} = p - \delta_p$ and $\delta_p(k) = p(k) - p(k - n_l)$ is a spatial shift.

Definition 3.18. The spatial forward shift operator is defined as

$$w^{\delta_p} \bar{x}(p) = \bar{x}(p + \delta_p), \quad (3.35)$$

To generate $f_{RC}(k)$, the GP is evaluated at the current position p by employing a spatial preview of δ_p to $\bar{\mu}(\bar{p})$ resulting in

$$\hat{d}_\mu(p) = w^{\delta_p} \bar{\mu}(\bar{p}) \quad (3.36)$$

as shown in Figure 3.10. Finally, the RC output becomes

$$f_{\text{RC}}(k) = \hat{d}_\mu(p). \quad (3.37)$$

The spatial forward shift allows to compensate for non-causal part of L , essentially, observations at sample k are stored in the GP at position $p(k - n_l)$ and used to perform inference at the test point $p(k)$. In contrast to traditional RC, the spatial preview δ_p is not limited by the buffer size N . The GP model can be evaluated at arbitrary position in the future, potentially resulting in a posterior mean equal to the prior mean (zero by default) if there is little correlation between the test point and the training data.

Finally, the spatial RC with and without spatial preview are identical, therefore the stability condition in Theorem 3.15 remains valid under Assumption 3.13 where now the preview is included.

3.4.5 Periodic Kernel Design

What remains is to select a suitable kernel function that represents the spatial disturbance prior knowledge for GP regression. The kernel function imposes prior knowledge on the disturbance function \bar{d} , as shown in (3.17) and (3.15).

According to Definition 3.3, the underlying spatial disturbance is smooth and periodic with period p_{per} . Hence, a periodic kernel, that reflects the class of smooth and periodic functions is suitable. Note that traditional kernels as often used in system identification approaches, see, e.g., Chen et al. (2012); Frigola et al. (2013); Jidling et al. (2018); Pillonetto et al. (2014), do not impose these type of priors.

The periodic kernel function is given by,

$$\kappa(p, p') = \sigma_f^2 \exp \left(\frac{-2 \sin^2 \left(\frac{\pi(p-p')}{\lambda} \right)}{l^2} \right) \quad (3.38)$$

with hyperparameters σ_f being a scaling, l the length-scale and λ is the period, see Figure 3.11 for an example. The hyperparameter λ is known in RC and equal to the period p_{per} , the additional parameters σ_f and l have to be tuned. Tuning can be done based on measurement data, i.e., increasing the smoothness l yields more correlation between data points resulting in a smoother estimate and σ_f is used to express prior on the amplitude of the disturbance. From an engineering point of view, choosing l large may be desired to take high-frequency modeling errors into account, see Remark 3.19.

Including periodicity and smoothness as prior helps to extrapolate beyond the currently known training data for fast learning. Moreover, note that the period λ may be a real number, in contrast to traditional time-domain memory loops where the periodicity is always an integer multiple of the sample time. This is an advantage of the GP-based approach, i.e., also allowing to suppress

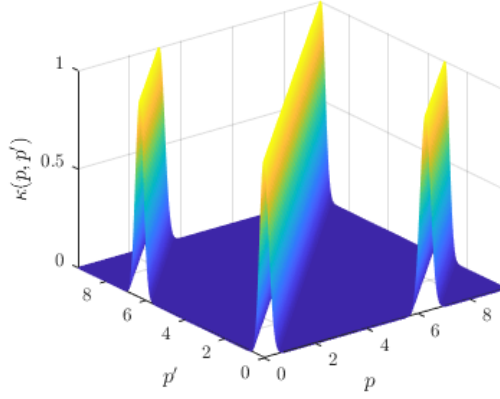


Figure 3.11: Example of a periodic kernel with hyperparameters $\lambda = 2\pi$, $l = 0.2$ and $\sigma_f = 1$.

disturbances with non-integer period times which is not possible with traditional RC (Hara et al., 1988), or requires interpolation in other spatial approaches (Yao et al., 2013).

Remark 3.19. *The high-frequency content in the GP-RC output f_{RC} is limited due to the smoothness, i.e., fulfilling a similar role as a robustness filter in traditional RC.*

3.4.6 GP-RC Procedure

Based on the above, a procedure to implement GP-RC is outlined below.

Procedure 3.20 (Position-domain RC using full GP).

(A) Initialization and prior

1. Design the learning filter, i.e., Procedure 3.17.
2. Select hyperparameters σ_f , λ , l and noise variance σ_n .
3. Set counters $k = i = 1$.

(B) At every sample k

- Obtain $\ell_d(k)$ by filtering $e(k)$ with L_c .

1. **if** $k > n_l$

- Add the i^{th} training observation to $\mathcal{D}(X, \mathbf{y}_d)$

$$X(i) = \bar{p} \quad (3.39)$$

$$\bar{y}_d(i) = \ell_d(k) + q^{-n_i} f_{\text{RC}}(k) \quad (3.40)$$

increase counter $i = i + 1$.

- Train the GP with data \mathcal{D} , i.e., compute α in (3.13).
- Prediction at test position p^* by computing (3.45b) with precomputed α .
- Set $f_{\text{RC}}(k) = \hat{d}_\mu(p)$.

2. *else*

- No training data available yet, i.e., $f_{\text{RC}}(k) = 0$.

3. *end* set $k = k + 1$.

An interesting observation is that the GP-RC output is non-zero from the n_l^{th} sample onwards, in contrast to classical RC, where a delay of one period is required before the disturbance can be compensated. This is caused by the extrapolation capabilities of the GP, which allows predicting the disturbance model n_l samples in the future based on the current observations, leading to much faster learning in the first period, in contrast to existing spatial approaches.

Remark 3.21. *In this procedure, all samples are included in the GP training data set \mathcal{D} , furthermore, at every sample the GP is trained with the available data. Due to high sample-rates in combination with smoothness and periodicity in the kernel, there may be a significant amount of redundant data available. Hence, the above procedure can significantly speed up by only including a subset of the samples to the training set, in addition, training of the GP can be done at a lower rate than the predictions.*

3.5 Computationally efficient GP-based RC

The computational complexity of GP regression hampers its practical implementation in RC, i.e., the computation time of a full GP scales cubically with the number of data points $\mathcal{O}(N^3)$. In this section, a sparse GP approximation based on inducing points is presented that, in combination with a periodic kernel, is highly suitable for the implementation in spatial GP-based RC. Also, optimization of hyperparameters and inducing points for sparse GP regression is presented (Snelson and Ghahramani, 2006).

3.5.1 Efficient GP for RC

The cubic complexity is caused by the inversion of the $N \times N$ matrix $(K_{XX} + \sigma_n^2 I_N)$, that appears in the posterior distribution (3.10). Several methods have been investigated to address the complexity requirement, including discarding data (Quiñonero-Candela and Rasmussen, 2005), full GP approximations (Liu et al., 2020), and prior approximations such as the fully independent training conditional (FITC) (Snelson and Ghahramani, 2006). In view of RC, the FITC approximation is particularly suitable, since it relies on a set of $M \ll N$ inducing points X_m . Due to the periodic kernel, the inducing point positions can be concentrated within one spatial period, i.e., a small number of inducing points is sufficient. This reducing the computation complexity to $\mathcal{O}(M^2 N)$.

3.5.2 Sparse GP regression for RC

To outline the sparse GP approximation, first define a sparse training set $\bar{\mathcal{D}}(X_m, \mathbf{f}_m)$ consisting of M pre-defined inducing points X_m with function values $\mathbf{f}_m = f(X_m)$. This set is used to support the full training data set \mathcal{D} .

Next, the joint prior (3.9) is written explicitly as function of the inducing points function values \mathbf{f}_m , i.e.,

$$p(\bar{y}_d, f_*) = \int p(\bar{y}_d, f_* | \mathbf{f}_m) p(\mathbf{f}_m) d\mathbf{f}_m, \quad (3.41)$$

where f_m is marginalized out, see, e.g., Snelson and Ghahramani (2007). Next, a key assumption in many GP approximations is that \mathbf{y}_d and f_* are conditionally independent and only connected through the inducing points \mathbf{f}_m , i.e., $\mathbf{f} \perp f_* | \mathbf{f}_m$. As a result, the joint probability distribution of two function values is equal to the product of the individual probabilities, i.e., the joint prior (3.41) now becomes

$$q(\mathbf{y}_d, f_*) = \int q(\mathbf{y}_d | \mathbf{f}_m) q(f_* | \mathbf{f}_m) p(\mathbf{f}_m) d\mathbf{f}_m, \quad (3.42)$$

where $q(\mathbf{y}_d | \mathbf{f}_m)$ and $q(f_* | \mathbf{f}_m)$ are referred to as the training and test conditionals respectively, and the prior on the inducing points remains exact, i.e., $p(\mathbf{f}_m) = \mathcal{N}(0, K_{UU})$, see, e.g., Liu et al. (2020); Quiñonero-Candela and Rasmussen (2005); Snelson and Ghahramani (2006) for a derivation. Here, the matrix $K_{UU} \in \mathbb{R}^{M \times M}$ is the covariance function κ evaluated at all combinations of inducing points X_m . The FITC algorithm approximates specifically the training conditional as follows

$$q(\bar{y}_d | f_m) = \mathcal{N}(K_{XU} K_{UU}^{-1} f_m, \Lambda + \sigma_n^2 I_N) \quad (3.43)$$

with $\Lambda = \text{diag}[K_{XX} - Q_{XX}]$ a diagonal approximation of the kernel matrix K_{XX} , and $Q_{AB} = K_{AU} K_{UU}^{-1} K_{UB}$, whereas the test conditional remains exact.

The posterior distribution for a new test point p_* is given by

$$p(f_*|p_*, X_m, \mathcal{D}) = \mathcal{N}(\mu_{FITC}, P_{\text{post}}^{\text{FITC}}), \quad (3.44)$$

where the mean and covariance are

$$\mu_{\text{FITC}} = K_{*U} \Sigma^{-1} K_{UX} (\Lambda + \sigma_n^2 I_N)^{-1} \mathbf{y}_d \quad (3.45a)$$

$$P_{\text{post}}^{\text{FITC}} = K_{**} - K_{U*}^\top (K_{UU}^{-1} - \Sigma^{-1}) K_{U*} \quad (3.45b)$$

$$\Sigma = K_{UU} + K_{UX} (\Lambda + \sigma_n^2 I_N)^{-1} K_{XU}. \quad (3.45c)$$

In line with the full GP, a single test point the FITC algorithm is also of the form (3.12) with

$$\alpha_{\text{FITC}} = \Sigma^{-1} K_{UX} (\Lambda + \sigma_n^2 I_N)^{-1} \mathbf{y}_d. \quad (3.46)$$

In the posterior distribution (3.45), the inversion of the matrix Σ is now $M \times M$ instead of $N \times N$. In addition, the inversion of $\Lambda \in \mathbb{R}^{N \times N}$ is inexpensive since it is a diagonal matrix. This reduces the computational effort to $\mathcal{O}(M^2 N)$.

Remark 3.22. *The sparse GP scales linearly with the number of data points N , instead of cubically for the full GP, significantly reducing the computational complexity. In a practical application N should be limited, e.g., by using a subset of the data containing the most recent N observations.*

The sparse FITC GP is implemented in Procedure 3.20 by additionally selecting the number of inducing points M , and their positions X_m in step (A). Then, in step (B) replace α with α_{FITC} to compute the sparse mean with (3.45b).

3.5.3 Hyper parameters and inducing points optimization

The location of the inducing points, as well as the hyperparameters, can be chosen manually. Alternatively, these can be optimized by maximizing the log marginal likelihood

$$\log p(\mathbf{y}_d | X_m) = -\frac{1}{2} \log |Q| - \frac{1}{2} \mathbf{y}_d^\top (Q)^{-1} \mathbf{y}_d \quad (3.47)$$

where $Q = K_{UU} + \Lambda$, with respect to the inducing points X_m and optionally hyperparameters. This can be performed off-line to initialize the inducing point locations and hyperparameters in one step (Snelson and Ghahramani, 2006).

3.6 Simulation case study

In this section, a simulation case study is carried out to show that spatial RC, with full and sparse GP, can effectively compensate for spatially periodic disturbances that are non-periodic in the time domain. To evaluate the obtainable performance, a comparison is made with traditional RC, which can obviously not deal with the velocity variations.

3.6.1 System and disturbance

The setting in Figure 3.3 is considered, where P replicates the dominant dynamics in the ϕ_z direction of the industrial substrate carrier. It is modeled as a second order mass-spring-damper system

$$P(s) = \frac{1}{Js^2 + ds + k}, \quad (3.48)$$

with inertia $J = 1 \text{ kg}\cdot\text{m}^2$, damping $d = 1 \text{ Nm/s}$, and stiffness $k = 10^4 \text{ N/m}$, and discretized by zero-order-hold with sampling frequency $f_s = 1000 \text{ Hz}$. A stabilizing feedback controller is designed, given by

$$C(z) = \frac{67539(z+1)(z-0.9196)}{(z-0.4524)(z+0.1137)}. \quad (3.49)$$

Furthermore, a spatially periodic disturbance $d(t)$ acts on the system, collocated with the control input torque T . The torque T in the substrate carrier corresponds to the sum of the lateral forces generated by the roller segments, multiplied by the length from the roller to the PoI on the belt, i.e., the effective torque at the PoI in r_z . The spatial disturbance is given by

$$\bar{d}(p) = 1.5 \sin(p) + 0.8 \sin(3p) + 0.6 \sin(9p) + 0.4 \sin(18p) + 0.2 \sin(27p),$$

where the velocity of $p(t)$ is $2\pi \text{ rad/s}$ at the start of the simulation, resulting in a time-domain disturbance period of exactly 1 second. In the second phase, the velocity varies over time, resulting in a non-periodic time-domain disturbance. Finally, the velocity is constant at 10.9 rad/s , such that the time-domain disturbance is periodic and the period is not an integer multiple of 1 second. The time-domain disturbance is shown in Figure 3.4. The reference signal is set to zero, i.e., a pure disturbance attenuation problem is considered.

3.6.2 Design of GP-RC, sparse GP-RC and temporal RC

Spatial GP RC: The learning filter is designed according to Procedure 3.17, resulting in $n_l = 1$ samples preview. The marginal likelihood (3.47) is optimized with respect to the hyperparameters to obtain $\sigma_f = 2.7$, $l = 0.13$, and $p_{\text{per}} = 2\pi$. The noise variance used throughout the simulation is $\sigma_n = 10^{-4}$, which is additive white noise on the disturbance. To reduce the computational load, every 10^{th} sample is added to \mathcal{D} and subsequently the GP is trained.

Sparse spatial GP RC: For the sparse GP RC, the number of inducing points M is set to 100, which are equidistantly distributed in the range $[0, p_{\text{per}})$. The same hyperparameters are used as in the full GP case, also every 10^{th} sample is used to train the sparse GP.

Temporal RC: For traditional RC, the learning filter is designed as the inverse of the complementary sensitivity function $T = 1 - S$, see, e.g., Longman (2010). The memory loop size is equal to the disturbance period at the start of the simulation, i.e., $N_{RC} = 1000$ samples. Traditional RC is used as a measure for the obtainable performance in the first constant velocity phase.

3.6.3 Results: GP-based disturbance model

First, the obtained GP-based disturbance model that is captured in the spatial memory loop is further investigated. In Figure 3.12, a snapshot of the GP-based memory loop is shown. The actual disturbance and GP posterior are given for two training cases, i) where $N = 1600$ observations ranging in the interval $[0, \frac{\pi}{2}]$ are used, and ii) where only the training value on $[0, 2\pi]$ are used to train the GP. A good estimate of the disturbance is obtained over the entire range. Due to the periodicity and smoothness in the kernel, i.e., a sample at the current position p is connected to its neighboring samples and samples at $p - n \cdot p_{per}$, the identified disturbance model clearly extends beyond the training points. Furthermore, where no data is present yet, the GP mean tends to zero and the confidence bound grows.

To investigate the model quality of the sparse GP, Figure 3.13 shows the root mean squared estimation error for the full GP and the sparse GP as function of the number of inducing points M . The inducing points are equidistantly distributed in on $(0, 2\pi]$. This shows that from $M = 80$ onwards the sparse GP obtains an equivalently good estimate of the disturbance compared to the full GP with 1600 training points. This indicates that a small number of inducing points, in combination with the periodicity of the kernel, is sufficient to support the full data set.

3.6.4 Results: performance comparison

The spatial RC and traditional RC error responses are shown in Figure 3.14, where the gray area indicates where the disturbance is non-periodic in time. In addition, the 2-norm of the error for each spatial period j normalized by its length N_j , is given in Figure 3.15. The following observations are made.

- The time response shows that spatial RC (—) significantly reduces the error from sample $n_t = 1$ onwards, i.e., suppression is obtained in the first period, after which the error reduces even more when more training data is gathered. Note that traditional RC does require one full period before compensation.
- The error norm shows that spatial RC (✕) and sparse spatial RC (✕) are not influenced by the change in velocity and maintain to have good performance. Off-course, traditional RC performance (✕) decreases due to the

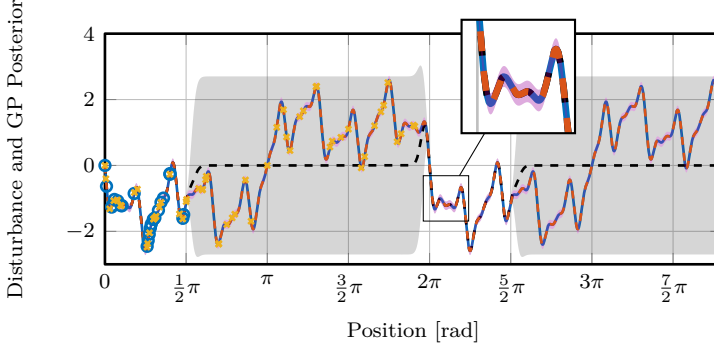


Figure 3.12: The true spatial disturbance (—), and the estimated GP posterior distribution evaluated on position $[0, 4\pi]$ depicted for two training sets. First, training points (○) in the interval $[0, \frac{\pi}{2}]$ resulting in the mean (--) and the standard deviation (■), which shows that the estimate extrapolates beyond the data due to periodicity, and the mean becomes zero where no information is available. Second, the full training set with $N = 1600$ points (×) on $[0, 2\pi]$ resulting in the mean (--) and variance (■), showing that a good estimate of the disturbance is obtained in the entire range $[0, 4\pi]$ based on data in the first spatial period.

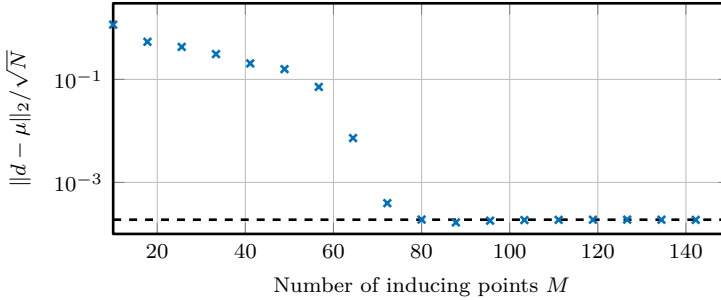


Figure 3.13: Root mean squared (RMS) estimation error of the full GP (--) with $N = 1600$ training points, and the sparse GP RMS estimation error (×) as function of the number of inducing points M , indicating that a small number ($M \approx 80 \ll N$) of equidistantly distributed inducing points in $[0, 2\pi]$ sufficiently supports the full data set.

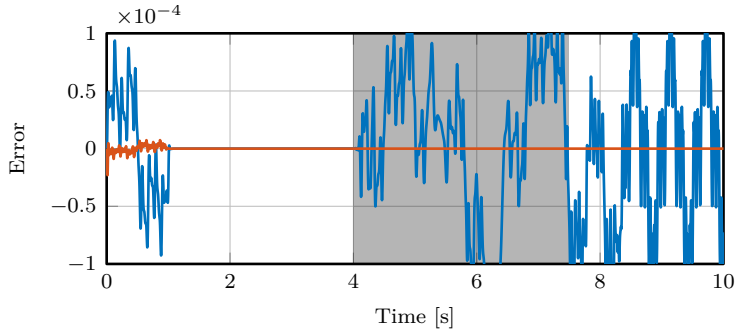


Figure 3.14: Simulated positioning error for traditional RC (—), and the spatial RC (—). Spatial RC is invariant under the velocity change, as indicated by (■), whereas traditional RC leads to performance degradation in case of velocity change.

inadequate buffer size. The temporal buffer size can be adapted for each velocity change, however, this introduces additional transients, it requires interpolation to use the learned compensation signal at the other velocity, and cannot cope with non-integer buffer sizes and continuous velocity changes.

- The performance of spatial RC, utilizing only 10 % of the data, is equal to the traditional RC performance in the first constant velocity part. Hence, it uses the data very efficiently due to the suitable prior knowledge.

Indeed, it follows that spatial RC is not influenced by the changing disturbance. The sparse GP approaches the full GP performance relatively well, while at the same time reducing the computational load.

3.6.5 Results: computation time

To show the benefit of the FITC GP in contrast to a full GP in terms of computation time, the computation time is measured as a function of the number of training data points, see Figure 3.16. The number of inducing points for the FITC GP is $M = 50$ points. It can be seen that indeed the full GP computational complexity grows with $\mathcal{O}(N^3)$, as indicated by the fit. The sparse GP with computational complexity $\mathcal{O}(M^2N)$ scales linearly with N , as shown by the linear fit. Furthermore, it can be observed that the computation gain starts to become significant for $N > 500$ data points, which is easily reached in mechatronic applications. Note that these results are obtained on a regular PC for illustration purposes, efficient employment on dedicated hardware can significantly reduce the computation time.

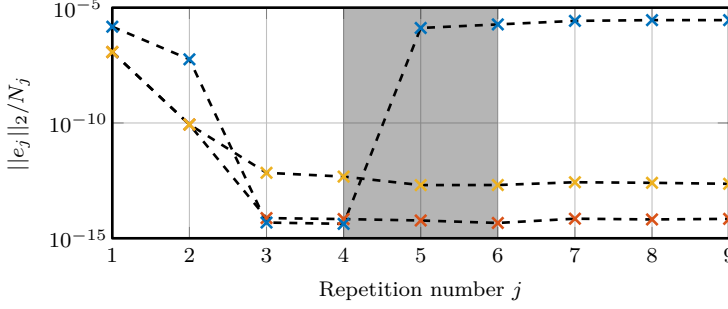


Figure 3.15: Simulated error 2-norm normalized with the period length N_j as function of the repetition number j , for traditional RC (\times), spatial RC with full GP (\times), and with sparse GP for $M = 100$ inducing points (\times). This indicates that spatial GP-RC obtains highly similar performance compared to traditional RC, while being able to maintain performance during velocity changes, indicated by (■) as in Figure 3.4, where traditional RC degrades significantly.

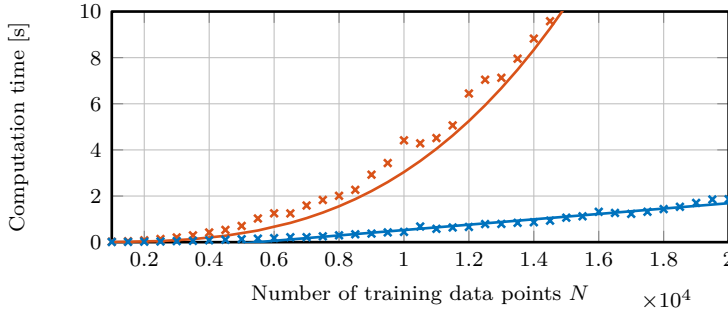


Figure 3.16: Computation time of the full GP with a cubic fit $\mathcal{O}(N^3)$ (—), and a sparse FITC GP for $M = 50$ inducing points with a linear fit $\mathcal{O}(M^2N)$ (—), as function of the training data N . The implementation for this example is not yet optimized, therefore, a large buffer size is chosen to avoid the overhead of the implementation yielding relatively large computation times.

3.7 Application to a substrate carrier system

In this section, spatial GP-RC is applied to the industrial substrate carrier in Figure 3.1 introduced in Section 3.2.1, that is subject to spatially periodic disturbances induced by the rollers. The aim is to validate that indeed spatial RC can reject the spatial disturbance invariant under variations in the roller velocity.

3.7.1 Substrate carrier setup

As mentioned in Section 3.2.1, the control goal of the substrate carrier is to accurately position a medium, e.g., paper or plastic, that is fixated to the steel belt by means of a vacuum. The steel belt is steered using two segmented rollers, that control the r_x , r_y and ϕ_z directions of the point-of-interest (PoI), see Figure 3.2. In this experiment, the aim is to keep ϕ_z equal to zero, hence this is a pure disturbance rejection problem.

3.7.2 Spatial disturbances in the substrate carrier

Due to imperfections in the rollers and segments, a spatially periodic disturbance appears in the error that repeats every roller rotation. To show this, a single experiment is performed that consists of 3 parts,

- constant roller velocity of 9 rad/s,
- deceleration of 5 [rad/s²] for 0.8 second which is approximately two roller rotations, and
- constant roller velocity of 5 rad/s.

The error is measured with a baseline PD controller, see (—) in Figure 3.17, where a periodic component is visible that changes frequency during the different parts. The power spectral density (PSD) and cumulative power spectrum (CPS) of the error without offset with velocity 9 rad/s (—) and velocity 5 rad/s (- -) is given in Figure 3.18 as function of the spatial frequency [1/rotation], i.e., the frequency [Hz] scaled by the rotational velocity of the rollers [rotation/s]. This shows that especially at 1 1/rotation and 3 1/rotation the error contains clear contributions that are repeating in the roller-position domain invariant under velocity variations. The first part corresponds to one roller revolution and the 3rd harmonic is most likely induced by having 3 segments in each roller that do not perfectly align.

During normal operation, the belt runs at several operating velocities, this leads to a situation where the disturbance becomes non-periodic in time. Traditional RC can only attenuate this disturbance for constant velocities, and may even amplify the disturbance when the velocity changes. In the remainder of this section, it is shown that spatial GP-based RC attenuates the spatial disturbance invariant under velocity variations.

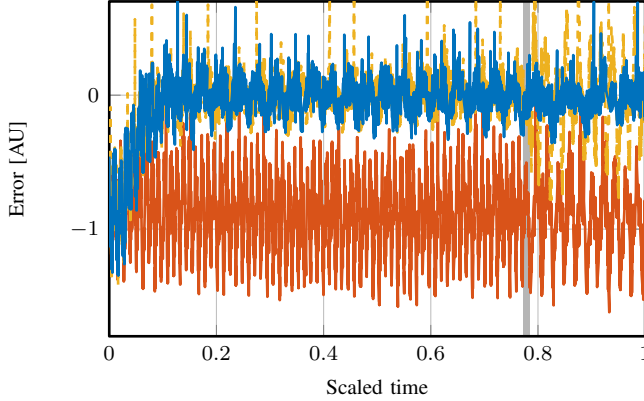


Figure 3.17: Error response as function of scaled time, for the PD controller (—), traditional RC (---) and spatial RC (—). The gray area (■) indicates where the roller velocity is changing. This shows that the periodic components and the DC offset is removed by the spatial RC controller independent of velocity changes. Here, AU is arbitrary unit.

3.7.3 Spatial GP-RC design

To implement spatial RC, a parametric model of the substrate carrier has been identified and the learning filter is constructed according to Procedure 3.17. The ZPETC algorithm is used to obtain a non-causal but stable learning filter, with $n_l = 2$ samples preview.

The periodic kernel hyperparameters are tuned such that the prior represents the actual spatial disturbance function. Tuning is performed with the measured error that is filtered by the learning filter, this results in an estimate of the spatial disturbance shown in Figure 3.19. Representing the filtered error as a function of position enables to tune the hyperparameters as follows;

- the spatial disturbance period is known and equal to one roller rotation, i.e., $\lambda = 2\pi$,
- the gain $\sigma_f = 27$ is an estimate of the deviation around the mean of the disturbance estimate,
- the GP estimate given the data is shown with $l = 0.05$ (—) and $l = 0.2$ (—) in Figure 3.19, this shows that a shorter length scale yields more high-frequency content in the estimate. To be more robust to noise and high-frequency modeling errors, a length scale $l = 0.2$ is preferred, and
- $\sigma_n = 1.7$ represents an estimate of the standard deviation of the noise yielding the confidence bound (■) in Figure 3.19.

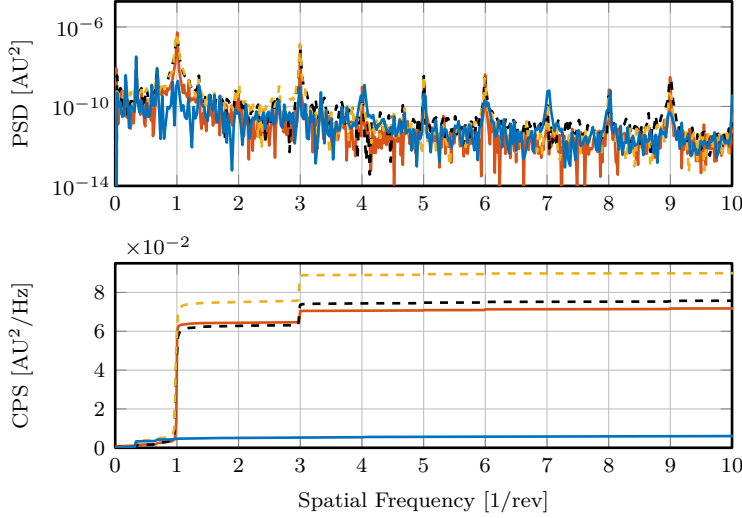


Figure 3.18: Power Spectral Density (PSD) (top plot) and Cumulative Power Spectrum (CPS) (bottom plot) as a function of the spatial frequency, for the error with PD controller at two different velocities (---) (9 [rad/s]) and (—) (5 [rad/s]). The disturbance is most prominent at the fundamental frequency [1/rev] and its second harmonic [3/rev] where the CPS shows significant increases in power independent of the velocity change. The error with spatial RC (—) and classical RC (---) after deceleration is also shown, indicating that spatial RC is not affected by the velocity change, whereas classical RC performance is severely degraded. Here, AU is arbitrary unit.

Note that a very short length scale l allows for more high-frequency components in the GP-RC output, i.e., the length-scale acts as a low-pass filter similar to a robustness filter in traditional RC. Because the smoothness l influences the GP input-output behavior, there is a direct connection between smoothness and stability through Assumption 3.13 and Theorem 3.15. Finally, to further reduce computational load, the data used to train a full GP is limited to 50 samples that are distributed equidistant over the last spatial period. The Sherman-Morrison update is used for GP regression to further improve the computational load for practical implementation, see Remark 3.7. The experimental setup runs at a sampling frequency of 4000 Hz.

3.7.4 Implementation aspects

In the general case for motion systems, the gain of $(PS)^{-1}$ is large for high frequencies, leading to amplification of high-frequency noise by the learning filter. Therefore, as a pragmatic solution, an additional zero-phase low-pass filter is

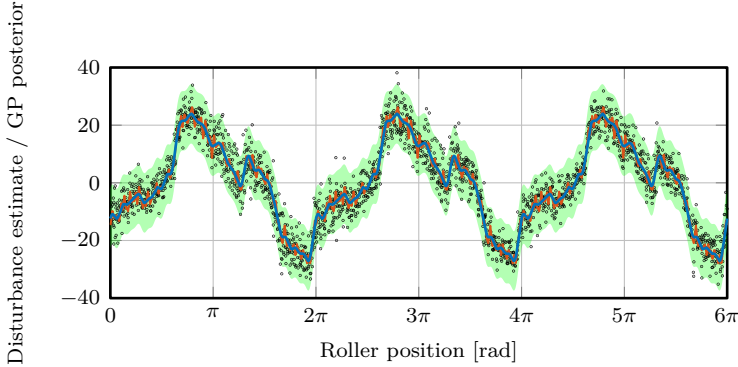


Figure 3.19: Experimental error data filtered with the learning filter L to obtain an estimate of the disturbance (\bullet), given as function of the roller position. The GP standard deviation 6σ (green) and mean (blue) with hyperparameters $\sigma_f = 27$, $\sigma_n = 1.7$, $\lambda = 2\pi$ and $l = 0.2$. To illustrate the effect of varying the length scale, the GP posterior mean is also computed for $l = 0.05$ (red) resulting in a much more erratic function.

placed in series with the learning filter that mitigates the effect of noise in ℓ . This does result in high-frequency modeling errors, but it improves the convergence of the disturbance model in the relevant frequency range, i.e., where a good disturbance model is most relevant. Note that the low-pass filter is designed such that the stability condition in Theorem 3.15 is still satisfied.

In addition, traditional RC is implemented as a comparison, with a buffer size that is equal to the disturbance period in the first constant velocity part of the experiment.

3.7.5 Results

The experiment outlined in Section 3.7.2 is carried out to analyze the performance of spatial RC in comparison to the baseline PD controller and traditional RC. The error responses are also shown in Figure 3.17. The converged error after deceleration is analyzed in the spatial frequency domain, i.e., Figure 3.18 shows in addition to the baseline PD error (red) also the spatial RC error (blue) and the traditional RC error (dashed blue), the corresponding cumulative power spectra (CPS) are also given. Note that a significant offset is present in the PD controlled error, this is removed from the data before computing the PSD and CPS such that the harmonics are better visible.

To analyze performance during velocity change, i.e., where the disturbance is non-periodic, the 2-norm of the error for each rotation j , normalized by the period length N_j , is given in Figure 3.20. Note that all experimental results

have been normalized for confidentiality, and the unit is denoted by arbitrary unit (AU). The following observations can be made.

- The dominant components in the error CPS, indicated by the increases in power at 1 and 3 [1/rotation] in the CPS in (—) in Figure 3.18, are completely suppressed by the spatial RC (—). In addition, Figure 3.17 shows that an offset is present in the PD controlled error (—) due to the lack of an integrator, which is learned and compensated for by the spatial RC (—). The overall performance improvement, including the offset, is a factor 12 on the RMS error, where a factor 3.5 is attributed to the suppression of the harmonic components.
- The PSD in Figure 3.18 shows that the error mainly at the fundamental frequency 1 [1/rotation] and the second harmonic 3 [1/rotation] are significantly reduced by the spatial RC (—) compared to the baseline error (—) that shows significant increases in the CPS. Note that only the 1th and 3rd contribute significantly to the CPS.
- The error-norm in Figure 3.20 shows that spatial RC (×) and traditional RC (×) have similar performance in the first constant velocity part, i.e., where the traditional RC buffer size ($N = 2787$) is compatible with the disturbance period, both resulting in an overall performance improvement of approximately a factor 12. This shows that spatial GP RC, with a low number of training points ($N = 50$), obtains equal performance for constant velocities.
- During deceleration, indicated by the gray area in Figure 3.20 and 3.21, the spatial RC error (×) is unaffected. This implies that the approach is robust with respect to velocity variations due to the spatial disturbance model. As a comparison, traditional RC (×) shows a performance degradation if the disturbance frequency changes. Due to inadequate buffer size, the traditional RC is not able to converge to a small error in the second constant velocity part.
- The PSD in Figure 3.18 shows that a periodic component at spatial frequency 0.3 [1/rev] is present, that most likely originates from a belt disturbance, which is amplified by the GP-RC (—). This can be explained since in general non-periodic components will be amplified by any RC as they do not align with the (spatial) buffer size, see, e.g., Chen and Tomizuka (2013).

From these observations it follows that indeed the spatial GP-RC approach suppresses the spatial disturbance and is not affected by changes in velocity, resulting in a major performance increase of a factor 12 of the RMS error.

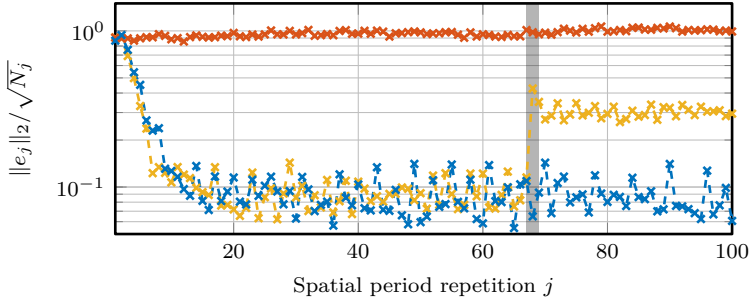


Figure 3.20: 2-norm of the error scaled with the \sqrt{N} for the PD controller (\times), Spatial RC (\times) and traditional RC (\times). The gray area () indicates where the velocity changes, showing the benefit of spatial RC where the performance is unaffected, whereas the traditional RC performance degrades due to inadequate buffer size after the velocity change.

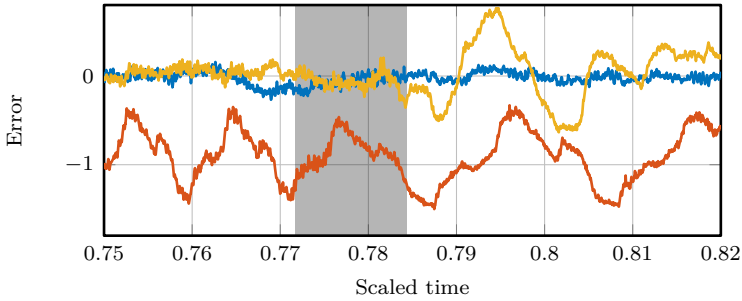


Figure 3.21: A zoom of the error as a function of time with PD only (—), traditional RC (—) and GP-based spatial RC (—) during the velocity change phase. This shows that indeed GP-based RC maintains good performance during and after the velocity change. Traditional RC obviously degrades performance due to the changing disturbance frequency that does not comply with the buffer size.

3.8 Conclusions

Position-domain disturbances that appear non-periodic in the time domain, but are repeating in the position domain, can be completely rejected by spatial repetitive control, where a key enabler is the introduction of a Gaussian Process (GP) based spatial memory loop, as presented in this chapter. The approach uses a memory loop in the position-domain together with a suitable learning filter to learn a spatial disturbance model. In contrast to existing repetitive control approaches, a spatial memory-loop is established by means of a Gaussian Processes with a suitable periodic kernel, thereby, efficiently dealing with non-equidistant observations. The disturbance is modeled as a stochastic process, i.e., a collection of random variables in space, that is estimated from data and suitable prior knowledge. The resulting distribution, in particular its mean, is a continuous function that is utilized in a spatial memory loop. The approach is validated in simulation, and on an industrial substrate carrier. Experimental results show a performance improvement of a factor 12 compared to currently implemented PD control, i.e., automatically learning and suppressing roller disturbances for arbitrary operating velocity in the industrial substrate carrier.

Ongoing work focuses on utilizing the full GP posterior distribution, i.e., also including the GP variance as a confidence measure on the disturbance model, see, e.g., Devasia (2017), that can act as a learning gain similar to traditional RC. In addition, extending the approach to cover multi-period spatial disturbances with different spatial periods (Blanken et al., 2020a; Chen and Tomizuka, 2011), e.g., to suppress roller and belt disturbances that have different (spatial) periods in the considered example, or to automatically learn commutation functions for brushless motors that minimize torque ripples.

CHAPTER

4

A Gaussian Process Approach to Multiple Internal Models in Repetitive Control¹

Abstract: Disturbances that come from multiple originating domains, e.g., time, position, or commutation-angle domain, are often encountered in practice due to the increasing complexity of mechatronic systems. The aim of this chapter is to present a generalized approach that enables asymptotic rejection of multi-dimensional disturbances which are periodic in the different originating domains, e.g., if speed changes, then spatially-periodic disturbances manifest themselves differently in the time domain. A multi-dimensional Gaussian process (GP) based internal model is employed in conjunction with a traditional repetitive control (RC) setting using non-equidistant observations, allowing to learn a multidimensional buffer for RC. A case study with a spatio-temporal disturbance confirms the benefit of this method.

¹The results in this chapter constitute Contribution 3 of this thesis. The chapter is based on Mooren, N. et al. (2022c). A Gaussian Process Approach to Multiple Internal Models in Repetitive Control. In *IEEE 17th International Conference on Advanced Motion Control*.

4.1 Introduction

Asymptotic rejection of unknown disturbances plays an important role to improve the performance of high-precision mechatronic systems. A key enabler for asymptotic disturbance rejection is the internal model principle (IMP), which states that a model of the unknown disturbance must be incorporated in a stable feedback loop (Francis and Wonham, 1976). Commonly used examples of IMP include a Proportional Integral (PI) controller, where a model of a constant disturbance, i.e., an integrator, is included in the feedback loop (Rivera et al., 1986), and an inverse notch filter to reject an oscillatory disturbance at a single known frequency. However, traditional feedback control performance is often not satisfactory in presence of varying disturbances. Several add-on type controllers for active disturbance rejection have been developed, including disturbance-observer-based control (DOBC) where a good model of the system is essential to estimate the exogenous unknown disturbance and consequently compensate for it, see, e.g., Li et al. (2014).

Engineered systems become increasingly complex, leading to disturbances that originate from different domains, e.g., time, position, or commutation angle (Hoelzle and Barton, 2016; Strijbosch et al., 2019), referred to as multi-dimensional disturbances. Examples include an industrial printer where a rotating belt generates a disturbance that is periodic in the belt-position domain (Mooren et al., 2020a), where at the same time the print head generates a disturbance that is periodic in time due to its repeating motion (Blanken et al., 2017b). Thermo-mechanical problems also appear, e.g., in wafer-stages with non-perfect commutations functions that induce spatial disturbances, while at the same time illumination of the wafer induces a thermal deformation (Evers et al., 2019; Veldman et al., 2019). In view of these multi-dimensional disturbances, traditional internal models are not applicable as they contain only a single domain, whereas systematic integration of multiple internal models in different domains is required.

Disturbances that are periodic in the time domain can be asymptotically rejected using repetitive control (RC), where an internal disturbance model is learned in a time-domain memory loop where the memory size is equal to the disturbance period time (Goodwin and Sin, 2014; Hara et al., 1988). RC is successfully applied to many industrial applications, including DC motors (Huo et al., 2019), printing systems (Blanken et al., 2017b) and hard-disk drives (Fujimoto, 2009; Steinbuch et al., 2007). To improve the flexibility of RC several extensions have been developed recently, e.g., multi-period disturbances in the time domain are compensated by using multiple RCs (Blanken et al., 2020a; Chang et al., 1998; Zhou et al., 2007), and robustness for slight variations in the period-time is improved using multiple buffers in higher-order RC (HORC) (Pipeleers et al., 2008; Steinbuch et al., 2007). To cope with disturbances that are periodic in the position domain, where observations become non-equidistant,

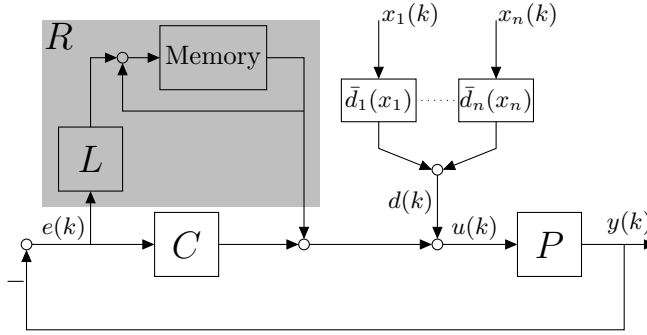


Figure 4.1: Disturbance rejection problem with the n -dimensional disturbance $d(k)$ and the conventional repetitive controller structure R .

spatial RC is developed by exploiting spatial internal models (Huo et al., 2019). Despite these recent improvements, existing RC approaches are not applicable when disturbances are periodic in multiple domains.

Although recent progress is made to enable asymptotic rejection of repeating disturbances, a unified framework that systematically integrates multiple domains is not yet available. Recent developments in RC utilize a Gaussian process (GP) based internal model for spatial models as in Chapter 3, and for temporal models in Chapter 2. These developments and the flexibility of GPs enables to systematically design multi-dimensional GP-based internal models for suppression of multi-dimensional disturbances in RC which is the aim of this chapter. The main contribution of this chapter is (C1) repetitive controller design for systematic integration of multi-dimensional periodic disturbances through a Gaussian-process-based buffer with a multi-dimensional additive periodic kernel, and (C2) a case study with a spatio-temporal disturbance.

In the remainder of this chapter, the multi-dimensional disturbance rejection problem is outlined in Section 4.2. In Section 4.3 the GP-based multi-dimensional internal model for RC is presented (C1), and in Section 4.4 a case study with a spatio-temporal disturbance is presented (C2). Conclusions are given in Section 4.5.

4.2 Problem formulation

4.2.1 Control setting

The control problem is depicted in Figure 4.1, where P is a linear time-invariant (LTI) single-input single-output (SISO) discrete-time (DT) system, C is a stabilizing feedback controller, and R is a repetitive controller (RC) that is designed in the forthcoming sections. The aim of this chapter is to design R such that

the disturbance induced error given by

$$e_d(k) = -(1 + P(q)C(q))^{-1}P(q)d(k), \quad (4.1)$$

where q is the unit time shift operator, i.e., $u(k-1) = q^{-1}u(k)$, is minimized in the presence of a multi-dimensional disturbance signal $d(k)$ that is in general non-periodic in time $k \in \mathbb{N}$. The multi-dimensional disturbance is further defined in the following section. Moreover, the sample time $T_s = 1$ throughout this chapter.

4.2.2 Multi-dimensional disturbances in RC

The n -dimensional disturbance signal $d(k)$ in Figure 4.1 is the summation of n signals $d_i(k)$ that are periodic in their respective domain x_i , e.g., position, time, or commutation angle, with $i \in \{1, 2, \dots, n\}$. This is formally presented in the following definition, where the notation $(\bar{\cdot})$ distinguishes functions from signals.

Definition 4.1. *The discrete disturbance signal $d(k) \in \mathbb{R}$ at sample k is obtained by evaluating the continuous n -dimensional function $\bar{d}(\underline{x}(k)) : \mathbb{R}^n \rightarrow \mathbb{R}$ for any $\underline{x}(k) = [x_1(k) \ x_2(k) \ \dots \ x_n(k)]^\top \in \mathbb{R}^n$, i.e.,*

$$d(k) = \bar{d}(\underline{x}(k)) = \sum_{i=1}^n \bar{d}_i(x_i(k)), \quad (4.2)$$

where the additive function \bar{d} is a sum of n functions $\bar{d}_i(x_i) : \mathbb{R} \rightarrow \mathbb{R}$ that are periodic in the domain x_i , i.e.,

$$\bar{d}_i(x_i) = \bar{d}_i(x_i + T_i) \quad \forall i \quad (4.3)$$

where $T_i \in \mathbb{R}$ is the period that is known in advance. Furthermore, the signals $x_i(k)$ for $i = 1, 2, \dots, n$ are known or can be measured noise-free, are possibly uncorrelated in the general case and can be non-equidistantly sampled.

The signal $d_i(k) = \bar{d}_i(x_i(k))$ is periodic in time if $x_i(k)$ increases linearly, but in the general case it is assumed that d_i is non-periodic in time. The overall disturbance $d(k)$ is periodic in time if all d_i are periodic with time-domain periods t_i and there exists a least common multiple $t = \text{lcm}(t_1, t_2, \dots, t_n)$, otherwise $d(k)$ is non-periodic in time. This is further illustrated in the following motivating example.

Example 4.2 (2-dimensional disturbance in a maglev-stage). *Consider the prototype magnetically levitated platform to carry a sample for atomic force microscopy in Figure 4.2. Here, a 2-dimensional disturbance $d(x_1(k), x_2(k))$ acts on the platform if the commutation is non-perfect, where the position signals $x_1(k)$ and $x_2(k)$ are not correlated, see Kramer et al. (2019) for details. To*

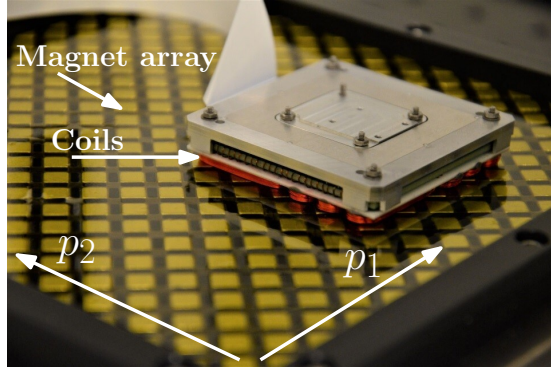


Figure 4.2: Picture of a prototype magnetically levitated platform (maglev) on an array of permanent magnets used to carry a sample for Atomic Force Microscopy (AFM). If commutation is non-perfect then the magnets induce a disturbance as function of the positions x_1 and x_2 .

replicate this setting, set $n = 2$ in (4.2) where x_1 and x_2 are the positions on the magnet array. The 2-dimensional disturbance function $\bar{d}(x_1, x_2)$ is replicated with $\bar{d}_1 = \frac{1}{2} \sin(2\pi \frac{x_1}{T_1})$ and $\bar{d}_2 = \frac{1}{2} \sin(2\pi \frac{x_2}{T_2})$ where $T_1 = 75$ and $T_2 = 50$ represent for example the magnet pitch, the resulting $\bar{d}(x_1, x_2)$ is depicted in Figure 4.3. Consider the following cases.

- i) If $x_1(k) = 2x_2(k)$ such that the platform moves diagonally and there is a static relation between x_1 and x_2 , i.e., the matrix $\bar{X} = [\bar{x}(1) \ \bar{x}(2) \ \dots \ \bar{x}(k)] \in \mathbb{R}^{n \times k}$ is it not full row rank, then $d(k)$ manifests periodic in time. The signal $d_1(x_1(k))$ has time-domain period $\frac{1}{2}T_1 = 37.5$ samples and $d_2(x_2(k))$ has $T_2 = 50$ samples, hence $\text{lcm}(50, 37.5) = 150$ samples, see Figure 4.3 in (—).
- ii) If x_1 and x_2 vary continuously as shown in the top right in Figure 4.3 in (—), then \bar{X} is full row rank and there is no static relation. This renders $d(k)$ non-periodic in time, see Figure 4.3 in (—).

Hence, if all signals x_i are uncorrelated the disturbance is non-periodic in time. Next, the limitations of traditional temporal buffers for the disturbances in Definition 4.1 are outlined.

4.2.3 Limitations of traditional RC

In traditional RC, a time-domain memory z^{-N} in a feedback loop is used to model the time-domain periodic disturbance. In the case of multi-dimensional disturbances, the main challenge that arises for RC is twofold, first, the memory

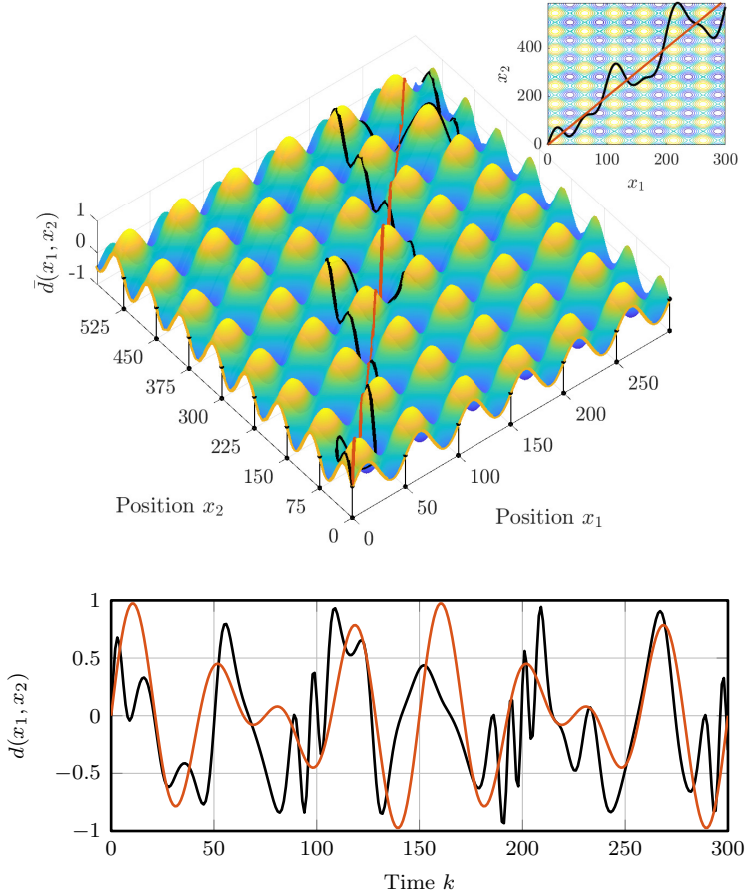


Figure 4.3: Example 4.2: top plot show the 2-dimensional disturbance mapping $\bar{d}(x_1, x_2)$ as function of x_1 and x_2 , with periods $T_1 = 50$ and $T_2 = 75$ and a top view. Bottom plot shows the resulting disturbance $d(k)$ if $x_1 = 2x_2$ when $d(k)$ is periodic in time (—) and for varying x_1 and x_2 rendering $d(k)$ is non-periodic in time (—).

loop must combine multiple domains, second, due to variations in $\underline{x}(k)$ the error data is non-equidistant in $\underline{x}(k)$. To compensate for the disturbance, the buffer must be a continuous function that is constructed from non-equidistant data and can be evaluated at any $\underline{x}(k) \in \mathbb{R}^n$. This implies that traditional memory loops, which rely on time-domain periodicity, cannot be used. The alternative RC approach in Chapter 3 uses non-equidistant data in the position-domain by employing a Gaussian process (GP) to learn a continuous periodic function in the position domain. However, combining multiple uncorrelated domains in RC is not yet available.

4.2.4 Problem definition

The aim of this chapter is to learn the multi-dimensional disturbance using an additive GP with suitable prior through a kernel function that is periodic in the underlying domains. This is done by using the structure and periodicity of \bar{d} , i.e., the functions \bar{d}_i are learned jointly to compose the continuous n -dimensional mapping $\bar{d}(\underline{x})$ from non-equidistant and non-periodic data.

4.3 Multi-dimensional repetitive control

4.3.1 Multi-dimensional buffer for RC

The multi-dimensional GP-based RC R is depicted in Figure 4.4, where \mathcal{GP}_B is the multi-dimensional GP-based buffer that replaces the time-domain memory z^{-N} from traditional RC, and L is a stable learning filter. GP regression is performed at every sample k to learn a model of the multi-dimensional disturbance function $\bar{d}(\underline{x})$, based on N data points that are non-equidistant in the domains \underline{x} . For compensation in RC, the GP-based model can be evaluated at any $\underline{x}(k) \in \mathbb{R}^n$.

The data for GP regression consists of the past N values of $y_d(k) \in \mathbb{R}$ and the corresponding $\underline{x}(k)$. These are collected in $Y(k) \in \mathbb{R}^N$ and $X(k) \in \mathbb{R}^{n \times N}$ respectively given by

$$Y(k) = [y_d(k) \quad y_d(k-1) \quad \dots \quad y_d(k-N+1)]^\top, \quad (4.4a)$$

$$X(k) = [\underline{x}(k)^\top \quad \underline{x}(k-1)^\top \quad \dots \quad \underline{x}(k-N+1)^\top]^\top. \quad (4.4b)$$

Define the n -dimensional training data set $\mathcal{D}_N(k) = (X, Y)$ containing N pairs $(y_d(k), \underline{x}(k))$.

The number of data points N may vary over time and is not correlated with the periodicity, which is the case in traditional memory loops. The periodicity in each of the underlying domains is embedded in the multi-dimensional prior as will be shown in the remainder of this section.

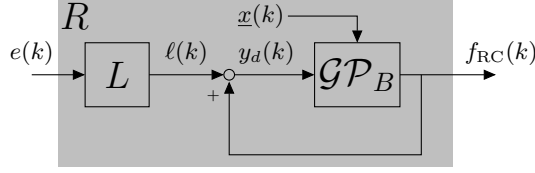


Figure 4.4: Multi-dimensional repetitive controller R with learning filter L Gaussian-process-based buffer \mathcal{GP}_B .

4.3.2 \mathcal{GP} regression for multi-dimensional disturbances

In this section, the multi-dimensional internal disturbance model of \bar{d} is learned from the data \mathcal{D}_N through Gaussian process regression. In GP regression is a distribution over functions is inferred from data and a suitable prior distribution (Murphy, 2012; Williams and Rasmussen, 2006). The n -dimensional and potentially non-equidistant data \mathcal{D}_N is used to learn the posterior distribution over functions, that can be evaluated at any $\bar{x} \in \mathbb{R}^n$ for compensation in RC. Next, the prior knowledge and posterior distribution are defined to give an expression for the GP-RC output $f_{RC}(k)$.

To define prior knowledge assuming that $\bar{d}(\underline{x})$ can be represented as a GP, i.e.,

$$\bar{d}(\underline{x}) \sim \mathcal{GP}(0, \kappa(\underline{x}, \underline{x}')), \quad (4.5)$$

where $\kappa(\underline{x}, \underline{x}')$ is an n -dimensional covariance function that describes the prior covariance between \underline{x} and \underline{x}' , and the prior mean function is assumed to be zero a priori. Moreover, \mathcal{D}_N contains noisy observations of the true disturbance, i.e.,

$$Y(X) = \bar{d}(X) + \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2 I_N) \quad (4.6)$$

where ϵ is zero-mean and follows an independent, identically distributed Gaussian distribution. To determine the posterior distribution at $\underline{x}_*(k) \in \mathbb{R}^n$ using \mathcal{D}_N and prior knowledge (4.5) for compensation, assume a joint normal distribution between the test points $\underline{x}_*(k) \in \mathbb{R}^n$ and the training points $\mathcal{D}_N(X, Y)$

$$\begin{bmatrix} Y \\ \bar{d}_* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K + \sigma_n^2 I_N & K_* \\ K_*^\top & K_{**} \end{bmatrix} \right), \quad (4.7)$$

where $K \in \mathbb{R}^{N \times N}$ is the covariance function $\kappa(X, X)$ evaluated at (X, X) , and similar for $K_* = \kappa(X, X_*) \in \mathbb{R}^{N \times 1}$ and $K_{**} = \kappa(X_*, X_*) \in \mathbb{R}$. Consequently, the posterior distribution at the test points \underline{x}^* is given by $p(\bar{d}_* | \mathcal{D}, X_*) = \mathcal{N}(\bar{d}, \Sigma)$ with

$$\bar{d}(X_*) = K_*^\top (K + \sigma_n^2 I_N)^{-1} Y, \quad (4.8a)$$

$$\Sigma(k) = K_{**} - K_*^\top (K + \sigma_n^2 I_N)^{-1} K_* \quad (4.8b)$$

the posterior mean and covariance respectively (Williams and Rasmussen, 2006).

For compensation in RC, the most likely sample from the posterior distribution (4.8), i.e., the posterior mean $\bar{d}(X_*)$, is used. The RC output is given by

$$f_{\text{RC}}(k) = K_*^\top (K + \sigma_n^2 I_N)^{-1} Y(k), \quad (4.9)$$

for a test point $\underline{x}^*(k)$.

The choice of prior knowledge is essential to obtain an accurate model and be able to extrapolate beyond the given data points. How to choose the prior covariance function in (4.5) is outlined next.

4.3.3 Multi-dimensional kernel selection

A suitable prior covariance function κ in (4.5) where periodicity in each of functions \bar{d}_i is incorporated together with the additive structure in Definition 4.1 allows combine multiple domains in a single GP-based buffer for RC.

Assume that the functions \bar{d}_i for $i = 1, 2, \dots, n$ can be modeled as a GP, i.e., $\bar{d}_i(x_i) \sim \mathcal{GP}(0, \kappa_i(x_i, x'_i))$, where $\kappa_i(x_i, x'_i)$ is the covariance function that specifies prior knowledge regarding \bar{d}_i . Since, each \bar{d}_i is periodic and smooth, a periodic covariance function of the form

$$\kappa_i(x_i, x'_i) = \sigma_{f,i}^2 \exp \left(\frac{-2 \sin^2 \left(\frac{\pi |x_i - x'_i|}{T_i} \right)}{l_i^2} \right), \quad (4.10)$$

is used where l_i is the smoothness, T_i the period and $\sigma_{f,i}$ a gain, see, e.g., (Williams and Rasmussen, 2006, Chapter 4). Consequently, the multi-dimensional covariance function κ for $\bar{d}(\underline{x})$ in (4.5) is

$$\kappa(\underline{x}, \underline{x}') = \sum_{i=1}^n \kappa_i(x_i, x'_i) \quad (4.11)$$

which is an additive n -dimensional covariance function (Durrande et al., 2011).

As a result, a sample taken from the prior distribution $\mathcal{N}(0, \kappa(X, X'))$ with covariance function (4.11) is smooth and can be decomposed in a sum of functions that are periodic and smooth in each of the underlying domains x_i , this is illustrated in the following example.

Example 4.3 (Spatio-temporal kernel). *Consider $n = 2$ in (4.11) with hyperparameters $T_1 = 100$, $T_2 = 150$, $l_1 = l_2 = 1$ and $\sigma_{f,1}^2 = \sigma_{f,2}^2 = 1$. The signal $x_1(k)$ is a position signal that varies over time and $x_2 = k$, i.e., a spatio-temporal setting. The kernel matrices $K_1(x_1, x'_1)$, $K_2(k, k')$ and the sum $K(\underline{x}, \underline{x}') = K_1(x_1, x'_1) + K_2(k, k')$ and samples drawn from the corresponding prior distributions $\mathcal{N}(0, K_i)$ are shown in Figure 4.5. The following observations are made.*

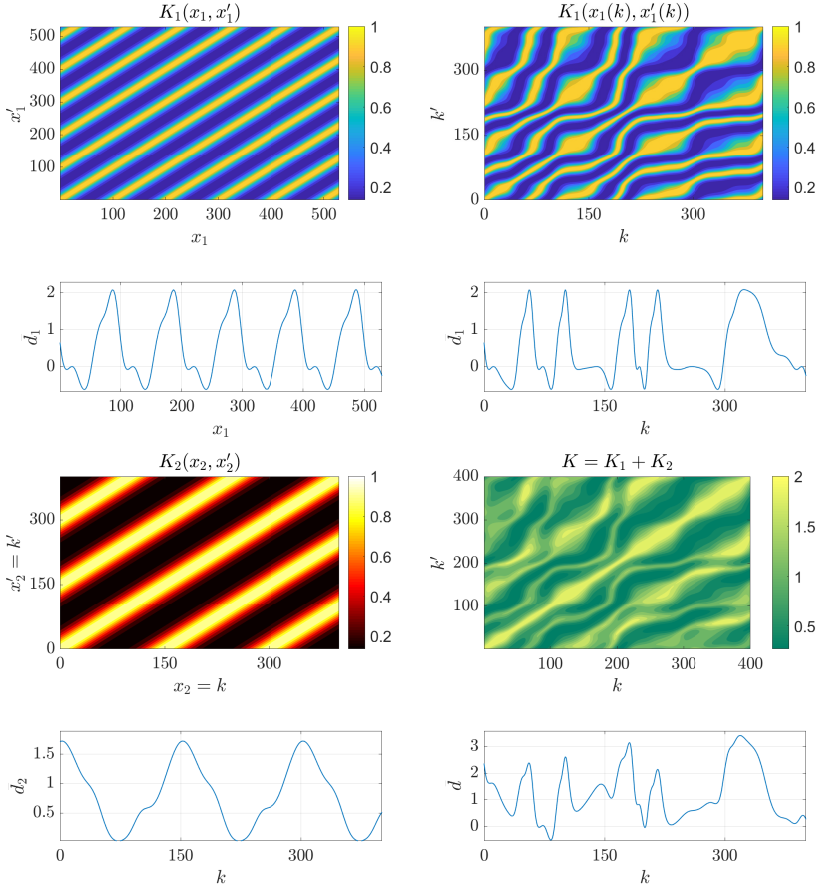


Figure 4.5: Example 4.3 with a 2-dimensional spatio-temporal kernel $K(x_1, x_2) = K_1(x_1) + K_2(x_2)$ (bottom right), that is the sum of two periodic kernels $K_1(x_1, x_1')$ (top left) and $K_2(x_2, x_2')$ (bottom left) with periods $T_1 = 100$ and $T_2 = 150$. Samples are taken from their corresponding prior distributions $\mathcal{N}(0, \kappa_i(X, X))$ that are periodic with T_1 and T_2 respectively. The kernel $K_1(x_1(k), x_1'(k))$ is also shown as function of k (top right) where both the kernel and a random sample is non-periodic. Finally, the sum of K_1 and K_2 yields $K(\bar{x}(k))$ (bottom right) that is non-periodic.

- The kernel matrix $K_1(x_1, x'_1)$ with period $T_1 = 100$ is large for any pair (x_1, x'_1) where $|x_1 - x'_1| \approx 100$ and is small elsewhere. This implies that x_1 and x'_1 are correlated if i) they are close together (smoothness) or they are T_i samples apart (periodicity). A random sample from $\mathcal{N}(0, \kappa(X, X'))$ is shown below, which is indeed periodic and smooth similar to K_2 with period 150.
- The kernel $K_1(x_1(k), x'_1(k))$ and its samples are non-periodic as a function of time k . The second plot in Figure 4.5 shows K_1 and its sample directly as a function of time k instead of $x_1(k)$, this yields a non-periodic kernel and non-periodic sample that depends on the specific variation of $x_1(k)$ over time.
- The 2-dimensional additive kernel $K(\underline{x}, \underline{x}')$ with $\underline{x} = [x_1 \ x_2]^\top$ is non-periodic as a function of k in the general case. Samples taken from the corresponding prior distribution $\mathcal{N}(0, \kappa(X, X'))$ are the sum of a sample from the individual prior distributions in the time domain due to the additive kernel structure.

Remark 4.4. A limitation of the presented approach, with a periodic kernel, is that individual differences between the coils are averaged out. These differences can be taken into account with a locally periodic kernel, see, e.g., Duvenaud (2014), where periodicities are local instead of over the entire range.

Example 4.3 shows that the n -dimensional kernel $K(\bar{x}(k))$ is non-periodic while the underlying kernels $K_1(x_1, x'_1)$ and $K_2(x_1, x'_1)$ are periodic in x_1 and x_2 respectively. This allows to integrate multiple domain with their periodicity, while perform GP regression with time-domain data.

4.3.4 Design procedure for GP-based RC

The following procedure outlines the design of GP-based RC for multi-dimensional disturbances.

Procedure 4.5 (Multi-dimensional RC design).

Given a model of the process sensitivity SP perform;

1. Design $L \approx (SP)^{-1}$ with $S = (1 + SP)^{-1}$ as a stable inverse as in Goodwin and Sin (2014); Hara et al. (1988), e.g., using ZPETC (Tomizuka, 1987).
2. Select $N \in \mathbb{N}$, determine n in Definition 4.1 by identifying the sources of the disturbance, and select the kernel hyperparameters in (4.11). As a guideline one can select N as the sum of the underlying periods, i.e., $N = \sum_{i=1}^n T_i$.
3. At each time step k :

- a) add the new sample $(\underline{x}_i(k), y_d(k))$ to \mathcal{D}_N and compute the RC output using (4.9).
- b) if \mathcal{D}_N exceeds N then remove $(\underline{x}(k - N), y_d(k - N))$ from \mathcal{D}_N .

There is a trade-off in the number of data points N used for GP regression, i.e., a better model can be obtained at the cost of increased computational complexity. Several approaches are available to reduce the computational complexity of GP regression, see, e.g., Snelson and Ghahramani (2006) and in Chapter 3.

Remark 4.6. *The inverse of SP can be non-causal with n_l samples preview, then implement the causal part L_c as learning filter and replace $\underline{x}(k)$ by $\underline{x}(k - n_l)$ in (4.4b) to include the preview in the buffer.*

Remark 4.7. *Conditions for closed-loop stability are equivalent to the results in Chapter 3 and omitted here.*

Remark 4.8. *If smoothness in the covariance function is large, then it is in practice often redundant to use all data points for GP regression, since two neighboring samples are highly correlated. If the singular values of the matrix $(K + \sigma_n^2 I_N)$ become small for a newly added data point, then the new data point can be considered redundant and discarded from \mathcal{D}_N .*

4.4 Rejecting 2D spatio-temporal disturbances

The rejection of spatio-temporal disturbances that are non-periodic in the time-domain through GPRC is shown in the following case study. These disturbances are encountered in, e.g., printing systems where the periodic motion of the print head induced a time-domain periodic disturbance while at the same time a roller that rotates with a varying angular velocity for paper transportation induces a disturbance that is periodic in the roller position domain and non-periodic in time (Blanken et al., 2020b; Mooren et al., 2020a). A similar setting is considered in the following case study.

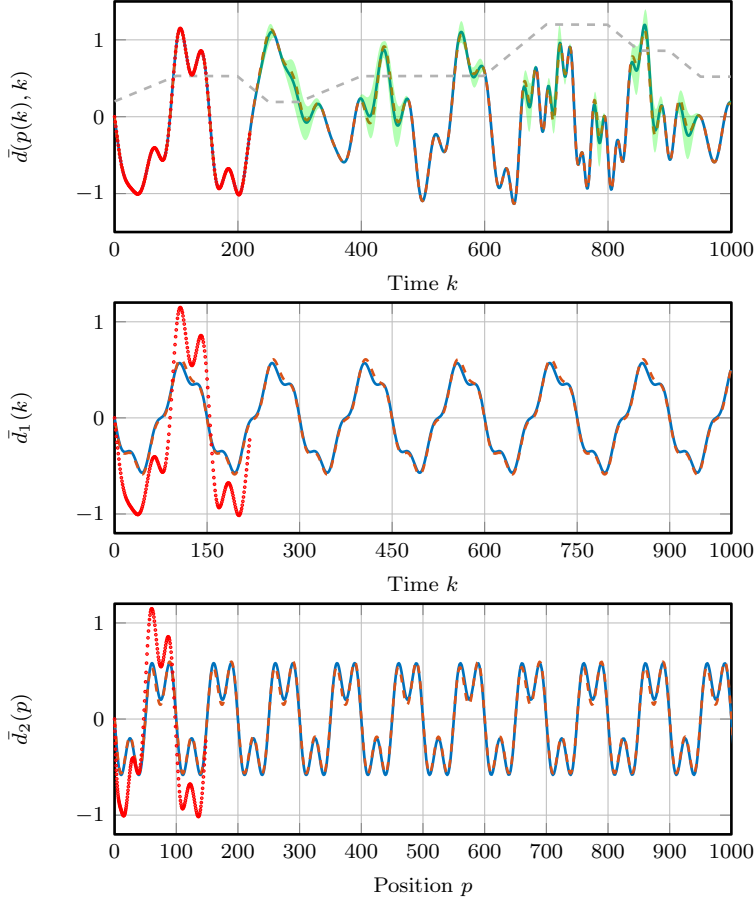


Figure 4.6: Top plot: spatio-temporal disturbance $\bar{d}(p, k) = \bar{d}_1(k) + \bar{d}_2(p)$ that is non-periodic in time (—) from which observations (●) are used for GP regression yielding the posterior mean (---) and variance (■). The position signal p is given in (---). Center plot: underlying temporally periodic disturbance component \bar{d}_k (—) with period $T_1 = 150$ and the GP estimate (---). Bottom plot: underlying spatially periodic disturbance component \bar{d}_p (—) with period $T_2 = 100$ and the GP estimate (---) as function of position. This shows that the estimate extrapolates well outside of the data points.

4.4.1 System and spatio-temporal disturbance

Consider the control setting in Figure 4.1 with

$$P(z) = \frac{0.05(z+1)}{z^2 - 1.98z + 1}, \quad (4.12)$$

and the stabilizing feedback controller is given by

$$C(z) = \frac{5.005(z+1)(z-0.81)}{(z+0.52)(z-0.03)}. \quad (4.13)$$

A spatio-temporal disturbance as in Definition 4.1 is present for $n = 2$, where the spatial and temporal functions are

$$\bar{d}_1(k) = \frac{1}{2} \sin\left(\frac{2\pi}{T_1}k\right) + \frac{1}{10} \sin\left(4\frac{2\pi}{T_1}k\right) \quad (4.14a)$$

$$\bar{d}_2(p) = \frac{1}{2} \sin\left(\frac{2\pi}{T_2}p\right) + \frac{3}{10} \sin\left(3\frac{2\pi}{T_2}p\right) \quad (4.14b)$$

respectively, with periods $T_1 = 150$ and $T_2 = 100$ as show in the second and third plot in Figure 4.6 in (—). The (roller) velocity of $p(k)$ varies over time, see (--) in Figure 4.6, rendering $d(k)$ non-periodic in time, see (—) in Figure 4.6 (top plot) as in Example 4.2. Moreover, the disturbance is subject to additive zero-mean normally distributed white noise with variance $\sigma_n^2 = 10^{-6}$.

4

4.4.2 Multi-dimensional GPRC design

The L filter is designed using ZPETC as in Procedure 4.5. All available data is used for GP regression such that N increases over time. Note that N is a design parameter, i.e., there is a trade-off between small N to reduce computation time and large N for estimation performance. The hyperparameters are chosen as $T_1 = 150$, $T_2 = 100$, $l_1 = l_2 = 1$, $\sigma_{f,1} = \sigma_{f,2} = 0.5$ and $\sigma_n^2 = 10^{-6}$. By increasing l a lower-frequency RC output is obtained which can be used to improve robustness against model errors, see Chapter 2.

4.4.3 Estimation performance

First, the estimation performance of the GP without RC is investigated with a small number of noisy observations $N = 220$ for training. Consequently, predictions are made outside of the training data and compared with the true disturbance. The result is shown in Figure 4.6, where (●) are the observations, (--) the posterior mean, and 99.7% the confidence bound (■).

An accurate estimate of the true disturbance, that is non-periodic in time, is obtained by estimating the underlying periodic functions (second and third plot) which extrapolates well using the periodic priors. Where the estimate deviates from the true disturbance the variance increases indicating that the estimate is less reliable. Next, the performance with GP RC is presented.

4.4.4 GP-based RC results

Control performance is analyzed with the error as shown in Figure 4.7 without GP-RC (—) and with GP-RC (—). As a measure for convergence the 2-norm

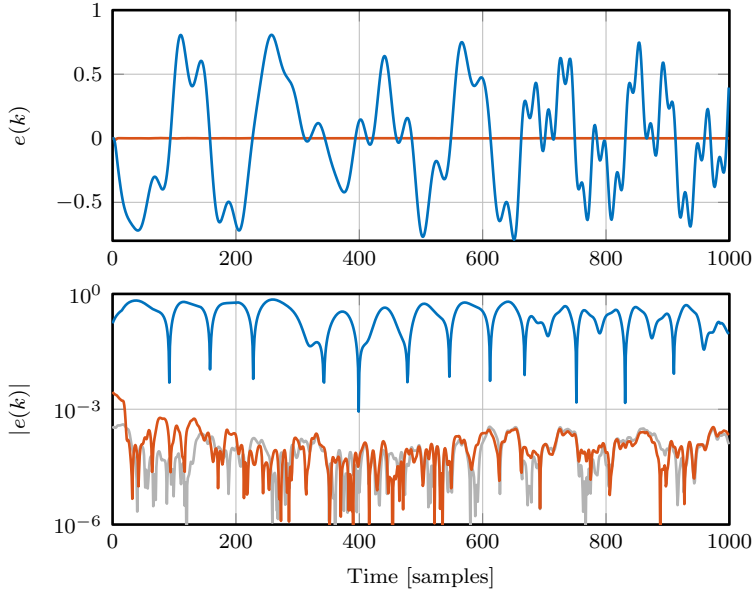


Figure 4.7: Error with feedback control (—) and with GP-based RC (—) as function of time (top plot) and the absolute value of the error on a logarithmic scale (bottom plot). As a comparison the contribution of the noise to the error is given in (—) (bottom plot) forming the best achievable performance.

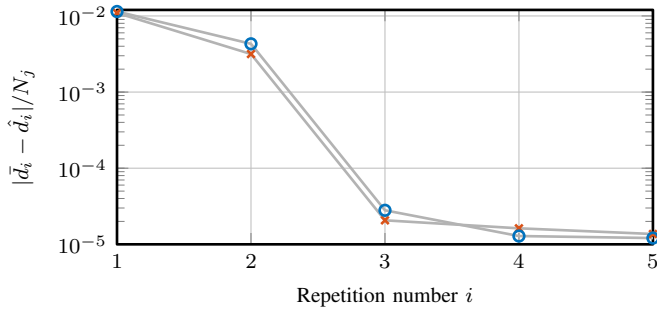


Figure 4.8: The 2-norm of the estimation errors for \bar{d}_1 (×) and \bar{d}_2 (○) computed over the periods T_1 and T_2 respectively, normalized by the period length N_i in samples. An accurate estimate is obtained after $i = 2$ periods.

of the estimation error for $d_1(p)$ and $d_2(k)$ is computed for $i = 1, 2$ during the periods T_1 and T_2 respectively, see Figure 4.8. The following observations are made:

- The non-periodic multi-dimensional disturbance is completely suppressed with GP-based RC in the first few samples, see Figure 4.7. The bottom plot in Figure 4.7 shows the moving mean of e on a log-scale confirming that GP-RC reduces the error up to the contribution of the noise (—).
- The non-periodic disturbance is learned through the periodic functions in the time and position domain. The 2-norm of the estimation error for \bar{d}_i is shown in Figure 4.8 indicating that after $i = 2$ periods the estimation errors are converges. This implies that also with fewer data sufficient control performance can be obtained.

To conclude, the considered GP-based multi-dimensional buffer for RC enables the rejection of multi-dimensional disturbances for arbitrary variations in $\underline{x}(k)$ over time.

4

4.5 Conclusion

The rejection of a multi-dimensional disturbance that is potentially non-periodic in time and periodic in multiple uncorrelated underlying domains, e.g., position, time, or commutation angle, is enabled through a multi-dimensional GP-based buffer in the traditional RC setting. Traditional disturbance models in RC cannot cope with the non-periodicity in the time-domain and integrating multiple domains requires new internal models for RC. In this approach, a new GP-based multi-dimensional internal disturbance model is generated by taking into account the underlying structure and periodicity in each of the domains through a multi-dimensional covariance function. Consequently, the GP-buffer is used for compensation. The approach is validated by means of a case study where a spatio-temporal disturbance is rejected for arbitrary position variations. Moreover, GP regression estimates confidence bounds that can be exploited in RC as a varying learning gain in ongoing work.

CHAPTER

5

Compensating Torque Ripples in a Coarse Pointing Mechanism for Free-Space Optical Communication: A Gaussian Process Repetitive Control Approach¹

Abstract: Actuators that require commutation algorithms, such as the coarse pointing assembly (CPA) for free-space optical communication used in this paper, often have torque-ripple disturbances that are periodic in the commutation-angle domain that deteriorate the positioning performance. The aim of this paper is to model the torque ripple as a Gaussian Process (GP) in the commutation-angle domain and consequently compensate for it at arbitrary velocity. The approach employs repetitive control (RC) at a constant velocity. A spatial GP with a periodic kernel is trained using data that is obtained from the RC step resulting in a static non-linear function for compensation at arbitrary velocity. Stability conditions are provided for both steps. The approach is successfully applied to a CPA prototype to improve the tracking performance for laser satellite communication, where the torque ripple is compensated at arbitrary velocity.

¹The results in this chapter constitute Contribution 4 of this thesis. The chapter is based on Mooren, N. et al. (2022b). Compensating Torque Ripples in Coarse Pointing Mechanism for Free-Space Optical Communication: A Gaussian Process Repetitive Control Approach. (*In preparation for journal submission*).

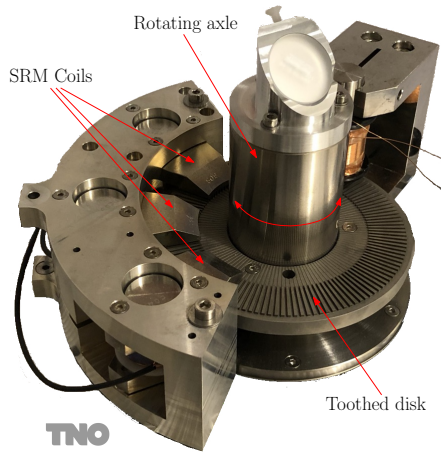


Figure 5.1: Picture of the coarse pointing assembly (CPA) prototype, developed and tested at the Netherlands Organisation of Applied Scientific Research (TNO).

5

5.1 Introduction

Many future technological advances, including scientific earth observations, 5G and 6G connectivity, and the Internet-of-Things, rely heavily on high-throughput and secure data communication with worldwide coverage. This calls for new advances in communication technology between satellites and from satellites to ground stations, well beyond the possibilities of the current radio-frequency (RF) standard. Recent developments in (inter-)satellite communication exploit free-space optical (FSO) communication, where a transmitter points a laser beam towards a receiver over a very long distance. Distances range from several hundreds of kilometers from lower-earth orbit (LEO) to ground, to thousands of kilometers between LEO satellites, up to tens-of-thousands of kilometers for links between ground stations and satellites in a geostationary orbit (GEO). FSO terminals can be much faster and lighter than their RF counterparts, partly because all the transmitted power can be condensed in a single beam with very small divergence, which on the other hand requires pointing accuracies in the order of micro-radians or even sub-micro-radians to cover such large distances. At the same time, FSO terminals need to point their lasers over very large angular ranges due to the relative motion between transmitter and receiver.

To enable a large-stroke rotation for tracking moving objects, a so-called coarse pointing assembly (CPA) is commonly used where size, weight, and cost are critical aspects for commercial (large series) space applications. Traditional CPAs are quite often relatively large, heavy, and expensive compared to the rest

of the FSO terminal, therefore, in (Kramer et al., 2020) a novel CPA actuation concept has been developed, see Figure 5.1, which offers a low-cost, low-power alternative for accurate long-stroke laser pointing for satellite-based FSO communication. This CPA concept, developed by the Netherlands Organisation of Applied Scientific Research (TNO), uses a switched reluctance motor concept (Miller, 2001) consisting of three motor phases and a commutation strategy to rotate a toothed disk and is relatively cost-effective to manufacture.

The positioning performance of the CPA depends on the interplay between the commutation functions and the mechanical tolerances; imperfect commutation easily leads to a torque ripple, e.g., due to misalignment of the three motor phases. Compensating this torque ripple disturbance can thus significantly improve the CPA performance. The commutation strategy is typically a trade-off between minimizing the torque ripple on one hand, and improving the power efficiency on the other hand (Miller, 2001; Vujičić, 2011), which are both key aspects for commercial space-based FSO communication. Moreover, the torque ripple that appears is repeating in the commutation-angle domain. In particular, if the CPA operates with a constant velocity, then this disturbance is periodic in the time domain, and it is non-periodic otherwise. The optimal torque-ripple compensation will also differ from CPA to CPA, due to manufacturing and assembly tolerances, which calls for automated routines to keep this CPA a cost-effective alternative.

Data-driven learning of a torque ripple compensation function can significantly improve control performance, see, e.g., Rohrig and Jochheim (2001), alternative model-based approaches require more modeling effort (Ai et al., 2021). Repetitive control (RC) enables asymptotic rejection of time-domain periodic disturbances with a known period, by learning an internal model from data (Francis and Wonham, 1976; Hara et al., 1988; Longman, 2010). In (Mooren et al., 2022d), a Gaussian process (GP)-based internal model is developed to incorporate prior knowledge for RC, which can be used to enable torque ripple compensation at constant velocity. Moreover, a spatial GP RC approach is presented in (Mooren et al., 2022a; Mooren et al., 2020b) for spatially-periodic disturbances that can be non-periodic in the time domain such as the torque ripple disturbance at varying velocity. However, this latter spatial approach updates continuously, requiring a substantial computational load, whereas an automated routine to compensate for the commutation-induced torque ripple does not necessarily require continuous updating, since it principally does not change over time. To avoid computational complexity in a low-power space-based environment, it is therefore preferred to learn a static compensation function that compensates for the torque ripple for arbitrary velocities.

Although recent developments in temporal RC can improve the positioning performance of the CPA for constant velocities, an approach that is computationally attractive and that allows compensating for the torque ripple at arbitrary velocity is not yet developed. The aim of this chapter is to compensate for the

torque ripple at arbitrary velocities by learning a static GP-based function in the position domain. The approach consists of two steps; first, time-domain GP-based RC (Mooren et al., 2022d) is applied for compensation at a constant velocity, and second, a spatial GP with a periodic kernel and data from step one is trained, resulting in a continuous compensation function that is periodic in the commutation-angle domain for compensation at arbitrary velocities. As such, the following contributions are identified:

- C1 analysis of the torque-ripple disturbance in the CPA to show that it is periodic in the commutation-angle domain, and modeling of the CPA as a linear system with a static non-linearity that represents the torque ripple (Section 5.2 and 5.3),
- C2 a two-step approach for learning of a static GP for torque ripple compensation at arbitrary velocity, including stability analysis for both steps (Section 5.4); and
- C3 experimental validation of the two-step approach for arbitrary velocity, including experimental validation of GP-based RC in (Mooren et al., 2022d) for constant velocity, on the CPA (Section 5.5).

The application of this method enables over 300% performance improvement in the CPA by suppressing a substantial part of the torque ripple at arbitrary velocity. Finally, conclusions are presented in Section 5.6.

5.2 Problem definition

The CPA depicted in Figure 5.1 is the main motivation for this work and is used for experimental validation. In this section, the setup is further introduced and the formal control problem is outlined.

5.2.1 Application motivation

The CPA in Figure 5.1 is part of the optical path for FSO communication, i.e., the control goal is to track a reference (laser beam) that depends on the relative position, orientation, and velocity between two moving objects. This implies that the rotational velocity of the rotating axle varies over time in the general case. The CPA uses a switched reluctance actuator containing three motor phases to drive a toothed disk with 131 teeth (Kramer et al., 2020). A cross-section of the coils and the toothed disk is schematically shown in Figure 5.2; by alternating the current between the three coils (through commutation functions) a continuous rotating motion can be generated. Important benefits of this actuator concept are the absence of physical contact between the coils and the rotating disk to reduce wear, and the absence of cogging disturbances. However, due to imperfect

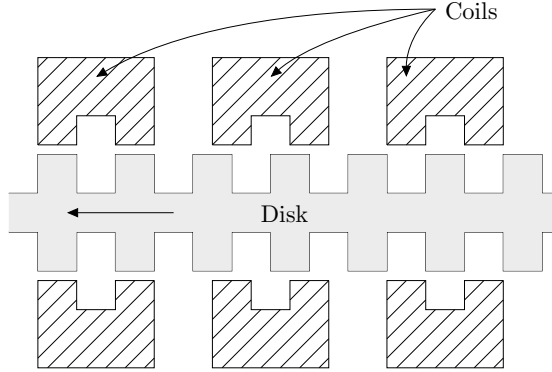


Figure 5.2: CPA actuation principle with three coils to drive the toothed disk.

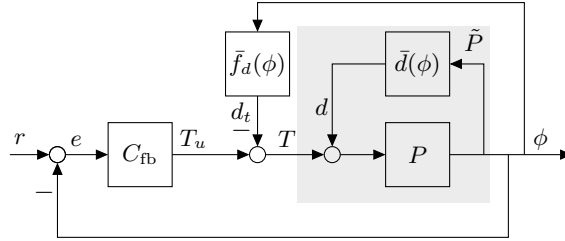


Figure 5.3: Torque ripple compensation strategy where the aim is to learn the compensation function $\bar{f}_d(\phi)$ from data. The CPA is modeled as \tilde{P} with non-linearity $\bar{d}(\phi)$ that is spatially periodic such that $d(k)$ represents a torque ripple or commutation-induced disturbance, and additional non-repeating disturbances \tilde{d} .

commutation between the different coil currents, there is a torque ripple present that could hamper performance.

5.2.2 Control setting

The relevant control setting for the CPA is depicted in Figure 5.3, where the goal is to design the static function $\bar{f}_d(\phi)$ to compensate for the torque ripple disturbance $\bar{d}(\phi)$, with ϕ the rotor position. The non-linear system \tilde{P} represents the CPA including commutation strategy and C_{fb} is a stabilizing feedback controller. It is assumed that the non-linear system \tilde{P} is composed of a linear time-invariant (LTI) part P in conjunction with the static non-linearity $\bar{d}(\phi) : \mathbb{R} \mapsto \mathbb{R}$ that is a function of the rotor position. The resulting signal that acts on the input to P

$$d(k) = \bar{d}(\phi(k)), \quad (5.1)$$

represents the torque ripple or commutation-induced disturbance to be compensated. Additional non-repeating disturbances, denoted by \tilde{d} , may appear in the CPA. Since these do not correlate with the position, they have very little influence on the presented approach and are therefore considered out of the scope of this chapter. The notation $\bar{\cdot}$ refers to spatial functions.

Ideally, the compensation function to be designed $\bar{f}_d(\phi)$ is an exact model of $\bar{d}(\phi)$, i.e., if

$$\bar{f}_d(\phi) = \bar{d}(\phi), \quad (5.2)$$

then the commutation-induced disturbance $d(k)$ is perfectly compensated. To show this, assume that the reference is suppressed, e.g., with an additive feed-forward $T_{ff} \approx P^{-1}r$ on T_u , then the commutation-induced error is given by

$$e = -PS(d - d_t), \quad (5.3)$$

where $S = (I + PC_{fb})^{-1}$ is the sensitivity function. If \bar{f}_d satisfies (5.2), then $d(k) = d_t(k)$ and the error (5.3) is zero for arbitrary velocity variations.

Consider the following properties of the non-linear function \bar{d} .

Definition 5.1. *The unknown static commutation-induced non-linearity $\bar{d}(\phi)$ is periodic, i.e.,*

$$\bar{d}(\phi) = \bar{d}(\phi + \beta), \quad (5.4)$$

with spatial period β that is equal to the tooth pitch in the CPA.

Assumption 5.2. *The unknown static non-linearity $\bar{d}(\phi)$ is sector bounded, i.e.,*

$$\gamma_1|\phi| \leq \bar{d}(\phi) - \xi \leq \gamma_2|\phi| \quad (5.5)$$

for some $\gamma_1, \gamma_2 \in \mathbb{R}$ with $\gamma_1 < 0 < \gamma_2$ and $\xi \in \mathbb{R}$.

Assumption 5.2 is required to provide stability conditions in the remainder of this chapter and is experimentally validated in Section 5.3. Definition 5.1 implies that $d(k)$ is periodic in the time-domain if the angular position $\phi(k)$ increases linearly, i.e., with constant rotational velocity, and $d(k)$ is non-periodic otherwise. This is similar to a torque ripple disturbance that also appears periodic in time with a constant rotational velocity, and non-periodic otherwise.

5.2.3 Two step control approach

A two-step design method is developed to learn the spatially periodic function \bar{f}_d from data. In the first step, data is collected to learn $\bar{f}_d(\phi)$ in the second step. To collect data for learning, recap that $d(k)$ is periodic in the time-domain with

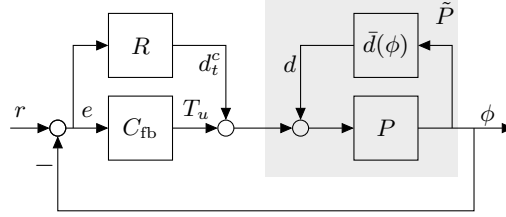


Figure 5.4: Step 1 of the developed approach: the repetitive controller R is employed to learn the time-domain model d_t^c of the torque ripple d with constant velocity. The data d_t^c is consequently used to learn $\bar{f}_d(\phi)$ for compensation at arbitrary velocity in Figure 5.3.

a known period if the rotational velocity is constant. However, due to imperfect commutation, the velocity might vary slightly in practice. In this situation, asymptotic rejection of $d(k)$ can be obtained through a robust repetitive control (RC) approach (Hara et al., 1988; Mooren et al., 2022d; Steinbuch et al., 2007) in the setting in Figure 5.4, i.e., R enables to asymptotically learn a model $d_t^c(k)$ of the periodic disturbance $d(k)$ in a time-domain buffer, even in the presence of a non-perfect model.

Compensation at an arbitrary velocity is done in the second step in the setting in Figure 5.3. It requires that the function $\bar{f}_d(\phi)$ is defined for all $\phi \in \mathbb{R}$, i.e., as a continuous function. Hence, the aim is to learn the continuous and periodic function $\bar{f}_d(\phi)$ in the commutation domain ϕ based on the time-domain data from step one, containing discrete points at various positions $\phi(k)$. Hence, interpolation and extrapolation of the data are required which is done by Gaussian process (GP) regression with a smooth and periodic prior as outlined in the remainder of this chapter.

5.3 CPA disturbance analysis

In this section, the commutation strategy that is designed for the CPA is briefly discussed. Moreover, it is shown that the CPA can indeed be modeled as in Figure 5.3, i.e., that a disturbance is present that is periodic in the commutation-angle domain.

5.3.1 CPA commutation strategy

The CPA setup with the commutation strategy for the three motor phases is schematically depicted in Figure 5.5. A brief explanation of the commutation strategy is shortly summarized below for background, details can be found in (Kramer et al., 2020). The system \tilde{P} in Figure 5.3 contains the commutation strategy and CPA dynamics, the physical CPA setup without commutation, i.e.,

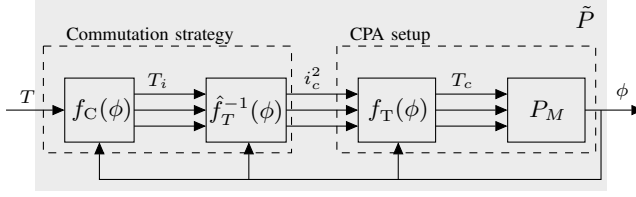


Figure 5.5: Commutation strategy and the CPA setup.

from the squared current of each of the coils i_c^2 to the rotor position ϕ is given by $P_M f_T(\phi)$ with $f_T(\phi) = \text{diag}(f_{T,1}, f_{T,2}, f_{T,3})$ an unknown position dependent function from coil current to torque output, and $P_M \in \mathcal{RH}_\infty$ the mechanical part that is proper and stable. Moreover, $f_{T,c}(\phi) : \mathbb{R} \mapsto \mathbb{R}$ for $c = 1, 2, 3$ is an unknown non-linear function of ϕ that maps i_c^2 to the torque generated by coil c

$$T_c = \frac{1}{2} \frac{\partial \ell}{\partial \phi} i_c^2 = f_{T,c}(\phi) i_c^2, \quad (5.6)$$

where $\ell \in \mathbb{R}$ is the inductance and while neglecting magnetic saturation, see, e.g., (Kramer et al., 2020; Schmidt et al., 2020, Chapter 5). The commutation strategy contains an approximate inverse of f_T denoted by \hat{f}_T^{-1} such that ideally $f_T(\phi) (\hat{f}_T^{-1}(\phi)) = I$ for all $\phi \in [0, \beta)$ with β the tooth pitch, and $f_C(\phi) : \mathbb{R} \mapsto \mathbb{R}^3$ are the commutation functions that distribute the required torque $T \in \mathbb{R}$ over the three individual motor phases. The functions f_C and \hat{f}_T^{-1} for this specific setup are developed in (Kramer et al., 2020) and used throughout the chapter.

Remark 5.3. The functions $f_C(\phi)$ and $\hat{f}_T^{-1}(\phi)$ are periodic with a single tooth and defined for $\phi \in [0, \beta)$ with $\beta = \frac{2\pi}{131}$ the tooth pitch, and extend periodically outside this interval, i.e., $f_i(\phi) = f_i(\phi + \beta)$.

Due to imperfections in the CPA setup and the commutation strategy, e.g., the requested torque per coil T_i is not equal to the actual torque T_c , there is a torque ripple disturbance present that depends on the position output ϕ as in Figure 5.3. Moreover, it is assumed that this torque ripple is repeating in the commutation-angle domain with a single tooth as in Definition 5.1.

5.3.2 Analysis of the disturbance

In this section, Definition 5.1 and Assumption 5.2 are experimentally validated to show that the torque ripple disturbance indeed repeats with a single tooth. Experiments are performed with a feedback controller at several constant rotational velocities ranging from 0.08 rad/s up to 0.14 rad/s with 0.02 rad/s increments. The resulting error is shown in Figure 5.6 as a function of time.

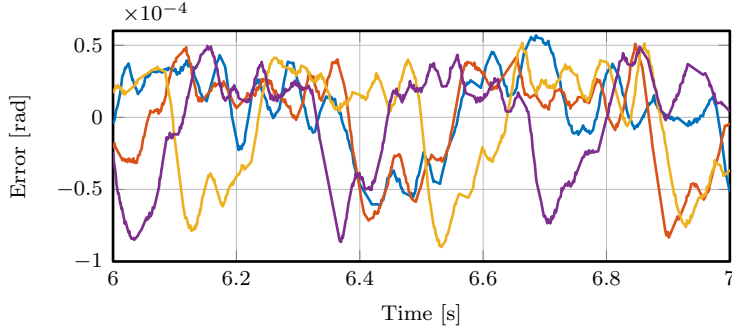


Figure 5.6: Measurement of the error as a function of time for different velocities 0.08 rad/s (—), 0.1 rad/s (—), 1.2 rad/s (—) and 1.4 rad/s (—) all increasing with 0.02 rad/s increments, which are all periodic but with different period.

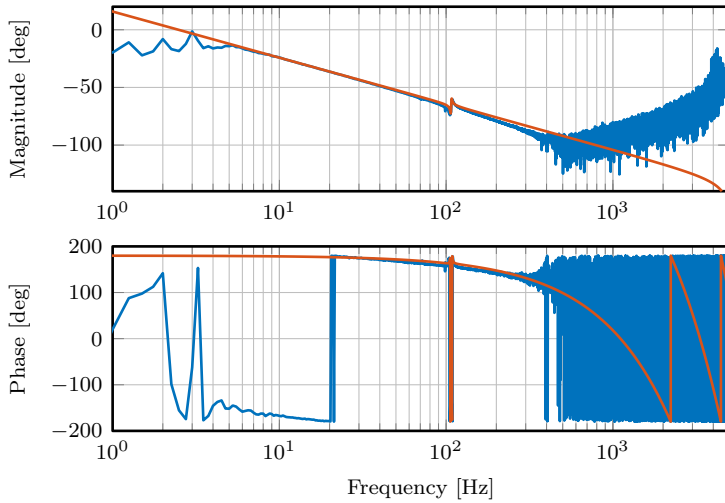


Figure 5.7: Measured frequency response function (FRF) (—) of the CPA setup with a model fit (—). At frequencies $f < 4$ Hz the friction in the CPA bearings leads to a measurement error.

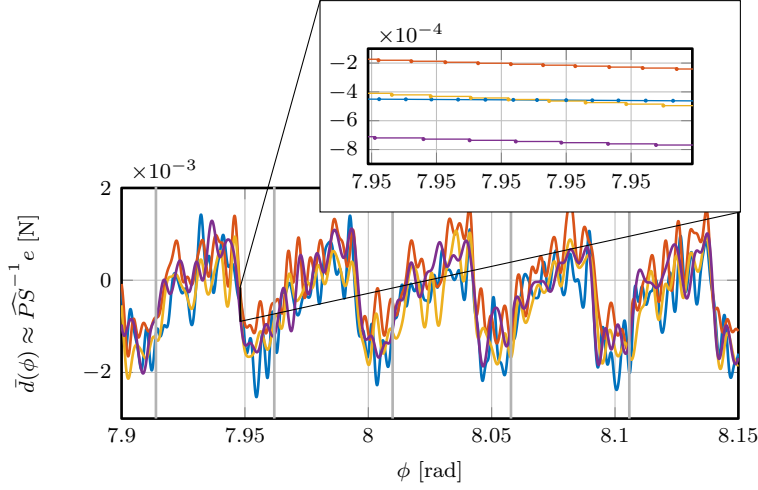


Figure 5.8: Measurement of the same error in Figure 5.6 now filtered with an inverse model of PS and plotted as a function of the position ϕ [rad], i.e., here all measurements have the same periodicity. Moreover, the zoom shows that the data sampled differently at each velocity.

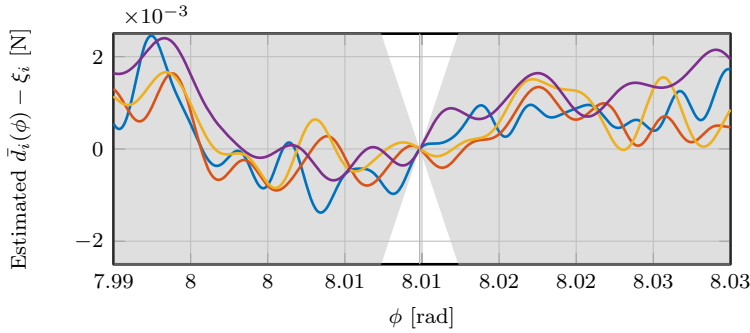


Figure 5.9: Estimates of $\bar{d}(\phi)$ obtained from measured error data at different velocities, 0.08 rad/s (—), 0.1 rad/s (—), 1.2 rad/s (—) and 1.4 rad/s (—), to validate the sector bound (\blacksquare) condition in Assumption 5.2 for $-\gamma_1 = \gamma_2 = 1$ rad/N.

Under the assumption that the CPA can be modeled as in Figure 5.3 with error (5.3), an estimate of the disturbance \hat{d} can be obtained from the measured error as

$$\hat{d} = \widehat{PS}^{-1}e, \quad (5.7)$$

where \widehat{PS}^{-1} is an approximate inverse model of the process sensitivity PS . The first transient part of the measurement data is omitted. A model \widehat{PS} of PS is obtained with the measured frequency response function of the CPA in Figure 5.7. Consequently, a stable inverse \widehat{PS}^{-1} is computed with zero-phase-error-tracking-control (ZPETC) (Tomizuka, 1987; van Zundert and Oomen, 2018). The resulting disturbance estimate is shown in Figure 5.8 as a function of ϕ . The following observations can be made.

- The dominant part of the error in Figure 5.6 is periodic in time for each velocity but the time-domain periods vary as the velocity changes.
- The disturbance estimates, i.e., filtered errors, as a function of the position in Figure 5.8 are all periodic with a single tooth $\frac{2\pi}{131} = 0.048$ rad independent of the velocity. Moreover, the disturbance estimates are similar for each velocity which motivates that the CPA can be modeled as in Figure 5.3 and validates Definition 5.1. There are however additional disturbances visible in Figure 5.8 that are not repeating; this chapter addresses the to compensation of the common commutation-induced or tooth periodic disturbance.
- Based on the measurement data in Figure 5.9, it can be concluded that the sector bound condition in Assumption 5.2 is valid for $\gamma_2 = -\gamma_1 = 1$.

Moreover, the zoom in Figure 5.8 shows that the data points are sampled differently for each velocity, i.e., non-equidistant in the rotor-position domain for a varying velocity. This yields that indeed the function $\bar{f}_d(\phi)$ for compensation must be defined for all ϕ to compensate for arbitrary velocity. The design of $\bar{f}_d(\phi)$ as a GP is outlined in the remainder of this chapter.

Remark 5.4. *The measured error can be used directly to approximate \bar{d} using a model \widehat{PS} , however, the equality of this estimate can be poor depending on the model quality. For this reason, in the developed two-step approach, RC (Mooren et al., 2022d) is used that is robust for modeling errors up to 100% and for slight velocity variations (Steinbuch et al., 2007).*

5.4 Gaussian process torque ripple compensation

In this section, the two design steps to learn the compensation function $\bar{f}_d(\phi)$ in Figure 5.3 are presented. It is shown first, how the continuous compensation function $\bar{f}_d(\phi)$ is learned given a data set with discrete observations of the disturbance through GP regression and suitable kernel design, which is the main contribution. This allows to evaluate $\bar{f}_d(\phi_i)$ at arbitrary $\phi_i \in \mathbb{R}$, i.e., also outside of the discrete training points. Thereafter, an RC approach is presented to generate the data set at constant velocity in the setting in Figure 5.4, i.e., with the non-linearity. Stability conditions are provided for both design steps.

5.4.1 Torque ripple compensation through GP regression

For compensation at arbitrary velocity, the compensation function $\bar{d}(\phi)$ must be a continuous function, i.e., known for all ϕ . This section outlines how a continuous function is generated based on a set of discrete observations of the disturbance through GP regression.

Suppose that a data set $\mathcal{D}_N(\phi_N, \hat{d}_N)$ is available where $\phi_N \in \mathbb{R}^N$ and $\hat{d}_N \in \mathbb{R}^N$ contain $N \in \mathbb{N}$ noisy data points $(\hat{d}(\phi(k)), \phi(k))$ that give information about the true spatial disturbance $\bar{d}(\phi)$, i.e.,

$$\hat{d}_N(\phi_N) = \bar{d}(\phi_N) + \epsilon_N, \quad (5.8)$$

and assume that $\epsilon_N \sim \mathcal{N}(0_N, \sigma_n^2 I_N)$ is additive independent identically distributed (i.i.d.) Gaussian noise. The data set \mathcal{D}_N is generated in a separate RC experiment at constant velocity as outlined in Section 5.4.4. The aim is to learn a continuous and periodic function based on \mathcal{D}_N that is also defined in-between the positions ϕ_N .

Many approaches exist to interpolate between discrete data points by assuming an underlying structure or basis functions, e.g., linear interpolation or least-squares regression. In this chapter, \bar{f}_d is obtained using GP regression, which does not require to specifically define basis functions. GP regression combines the data \mathcal{D}_N with prior knowledge, e.g., smoothness and periodicity, to determine a posterior distribution over functions (Murphy, 2012; Williams and Rasmussen, 2006). The mean of the posterior distribution, i.e., the most likely sample, is the continuous function $\bar{f}_d(\phi)$ that is used for compensation.

Suppose that the spatial disturbance function is a Gaussian process

$$\bar{d}(\phi) \sim \mathcal{GP}(\mu(\phi), \kappa(\phi, \phi')), \quad (5.9)$$

which is completely characterized by the mean function $\mu(\phi)$ that is assumed to be zero, and the covariance function $\kappa(\phi, \phi')$. The design of $\kappa(\phi, \phi')$ is outlined

in Section 5.4.2. By conditioning the GP, i.e., performing inference on the basis of \mathcal{D}_N and the prior (5.9), posterior distribution is obtained. The posterior distribution at any position $\phi_* \in \mathbb{R}$ can be computed by defining correlation between the ϕ_* and the data \mathcal{D}_N which is generated in a separate experiment, i.e.,

$$\begin{bmatrix} \hat{d}_N \\ \bar{d}(\phi_*) \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K + \sigma_n^2 I_N & K_* \\ K_*^\top & K_{**} \end{bmatrix} \right), \quad (5.10)$$

where $K \in \mathbb{R}^{N \times N}$ is the covariance function $\kappa(\phi_N, \phi_N)$ evaluated at all combinations of (ϕ_N, ϕ_N) , and similar for $K_* = \kappa(\phi_N, \phi_*) \in \mathbb{R}^{N \times 1}$ and $K_{**} = \kappa(\phi_*, \phi_*) \in \mathbb{R}$ (Williams and Rasmussen, 2006). As a result, the spatial disturbance model $\bar{f}_d(\phi_*)$ is given by the mean of the posterior distribution

$$\bar{f}_d(\phi_*) = \kappa(\phi_*, \phi_N) \alpha \quad (5.11)$$

$$\alpha = (K(\phi_N, \phi_N) + \sigma_n^2 I_N)^{-1} \hat{d}_N \in \mathbb{R}^N, \quad (5.12)$$

for the full posterior distribution, see, e.g., Mooren et al. (2022a); Williams and Rasmussen (2006). Note that the vector α is completely determined by the data $\mathcal{D}_N(\phi_N, \hat{d}_N)$, thus the implementation of (5.11) can be done efficiently in step 2 since the inversion in (5.12) can be computed a priori in step 1.

The compensation signal $d_t(k)$ is obtained by evaluated the GP-based spatial disturbance model (5.11) at the current position, i.e., $\phi_* = \phi(k)$,

$$d_t(k) = \bar{f}_d(\phi(k)) = \kappa(\phi(k), \phi_N) \alpha. \quad (5.13)$$

The following section presents the design of a suitable covariance function κ .

5.4.2 GP prior design for torque ripple disturbances

Since the spatial non-linearity is periodic as in Definition 5.1 the prior on $\bar{d}(\phi)$ must reflect this periodicity. Therefore, the covariance function in (5.9) is a periodic covariance function

$$\kappa(\phi, \phi') = \sigma_f^2 \exp \left(\frac{-2 \sin^2 \left(\frac{\pi(\phi - \phi')}{T} \right)}{l^2} \right), \quad (5.14)$$

with hyperparameters $\Theta_i = \{T, l, \sigma_f\}$. The hyperparameters can be interpreted as follows,

- $T \in \mathbb{R}$ is the spatial period of $\bar{d}(\phi)$ equal to the tooth pitch β ;
- $l \in \mathbb{R}$ is the smoothness; and
- $\sigma_f \in \mathbb{R}$ is a gain relative to the noise variance σ_n^2 .

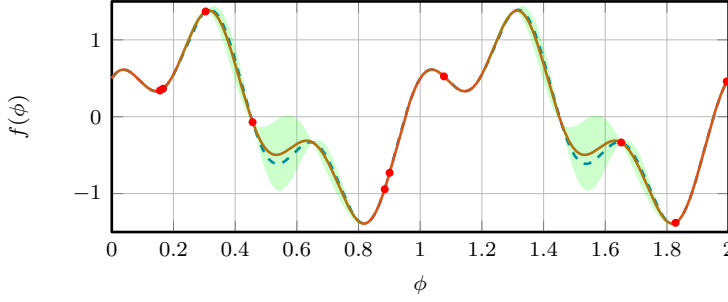


Figure 5.10: Example 5.5: Gaussian process regression is performed with 10 random samples (●) denoted by $(f(\phi), \phi)$ taken from the true periodic function $\bar{d}(\phi)$ (---). The resulting GP $f(\phi)$ has posterior mean (—) and posterior variance (■), which is a good estimate of $\bar{d}(\phi)$ also in-between training points by selecting appropriate smooth and periodic prior knowledge.

Now consider the following example.

Example 5.5. A periodic function $\bar{d}(\phi) = \sin(2\pi\phi) + \frac{1}{2}\sin(6\pi\phi + \frac{1}{2}\pi)$ with period 1 is used as true function, from which 10 random samples $(\bar{d}(\phi_i), \phi_i)$ for $i = 1, 2, \dots, 10$ are taken, see Figure 5.10. A periodic covariance function is used for GP regression, with the hyperparameters: $T = 1$ equal to the period, $\sigma_f = 1$ and $l = 1$ as in Figure 5.11. This shows that the covariance $\kappa(\phi, \phi')$ is large when $\phi - \phi'$ is either small (smoothness) or approximately equal to T (periodicity). GP regression is performed with the 10 data samples and periodic prior. The posterior mean and variance are shown in Figure 5.10 together with the true function. The posterior mean, i.e., the most likely sample from the posterior, is a continuous function that corresponds well with the true function also in-between the data points. Moreover, the variance increases where the estimate is less reliable.

Example 5.5 shows that with smooth and periodic prior in GP regression, an accurate estimate of the true periodic function can be obtained based on a limited number of discrete data points. This is used to interpolate and extrapolate the time-domain disturbance model for compensation at other velocities.

Remark 5.6. The periodic kernel (5.14) is also used in GP-based RC to learn a time-domain periodic function (Mooren et al., 2022d), or spatially periodic function (Mooren et al., 2020b).

5.4.3 Stability

To analyze stability of the interconnection in Figure 5.3 with compensation function \bar{f}_d , it is reformulated in the standard feedback interconnection of a linear

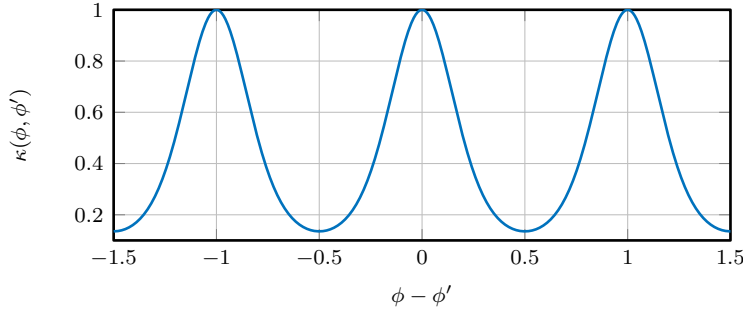


Figure 5.11: Example 5.5: periodic kernel function with period $T = 1$, smoothness $l = 1$ and $\sigma_f^2 = 1$.

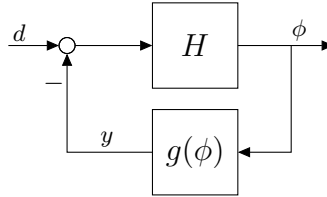


Figure 5.12: Standard feedback interconnection with the linear system H and a non-linear memoryless function $f(\phi)$.

system in feedback with a static non-linearity as in Figure 5.12. Consequently, a passivity-based stability analysis is used to provide stability conditions.

Lemma 5.7 (Circle criterion). *Consider the standard feedback interconnection in Figure 5.12 with the linear system $H \in \mathcal{RH}_\infty^{p \times p}$ and the memoryless function $f(\phi)$ as in (Khalil, 2002, Theorem 7.2) and (Desoer and Vidyasagar, 2009, p.140). If H is asymptotically stable and $f(\phi)$ satisfies the sector bound condition*

$$\gamma_1 |\phi| \leq f(|\phi|) \leq \gamma_2 |\phi| \quad (5.15)$$

for some $\gamma_1, \gamma_2 \in \mathbb{R}$ with $\gamma_1 < 0 < \gamma_2$, then the interconnection is uniformly asymptotically stable, i.e., the output converges asymptotically to a stable steady-state trajectory, for all $f(\phi)$ that satisfy (5.15) if the Nyquist plot of H lies in the interior of the circular disk $D(\gamma_1, \gamma_2)$, with $1/\gamma_1$ and $1/\gamma_2$ the intersections of $D(\gamma_1, \gamma_2)$ with the real axis.

The proof of Lemma 5.7 can be found in (Khalil, 2002, Theorem 7.2, Definition 6.2), and (Desoer and Vidyasagar, 2009, p.141).

Theorem 5.8. *Consider the setting in Figure 5.3 with linear system P , static unknown non-linearity $\bar{d}(\phi)$, non-linear function $\bar{f}_d(\phi)$ in (5.11), and suppose*

that Assumption 5.2 is satisfied. If

$$\|PS(e^{j\omega})\|_{\infty} < \frac{1}{\alpha}. \quad (5.16)$$

with $\alpha \in \mathbb{R}$, and $\bar{f}_d(\phi)$ is sector bounded,

$$-\rho|\phi| \leq \bar{f}_d(\phi) \leq \rho|\phi| \quad \forall \quad \phi \in [-\tfrac{1}{2}\beta, \tfrac{1}{2}\beta), \quad (5.17)$$

where ϕ can be shifted by $n\beta$ where $n \in \mathbb{Z}$, and for $\rho \geq 0$, such that

$$\rho + \gamma \leq \alpha \quad (5.18)$$

with $\gamma = -\gamma_1 = \gamma_2 \geq 0$ in Assumption 5.2, then the interconnection is asymptotically stable.

The proof of Theorem 5.8 is given in Section 5.7.

Remark 5.9. In Lemma 5.7 with $\lambda_1 = -\lambda_2$, the condition (5.17) in Theorem 5.8 is similar to a small gain condition (Khalil, 2002, Theorem 5.6). For $\lambda_1 \neq -\lambda_2$ more general results can be obtained by following the same reasoning as in Theorem 5.8.

Remark 5.10. The sector-bound condition (5.26) in Theorem 5.8 requires that $\bar{f}_d(0) = 0$. If $\bar{f}_d(0) \neq 0$ then it is of the form $\bar{f}_d(\phi) = \bar{f}_d^0(\phi) + c$ for some constant $c \in \mathbb{R}$ such that $\bar{f}_d^0(0) = 0$. In this case replace $\bar{f}_d(\phi)$ by $\bar{f}_d^0(\phi)$ in (5.26) and the results in Theorem 5.8 remain valid, as c can be interpreted as a constant disturbance.

Theorem 5.8 implies that the reciprocal of the \mathcal{H}_{∞} -norm of PS imposes the sector bound condition (5.26) and thereby an upper bound on the slope of $\bar{f}_d(\phi)$ around the origin. Moreover, the conditions in Theorem 5.8 can be checked on the basis of measurement data and the pre-designed function $\bar{f}_d(\phi)$ as shown in the following example for the CPA.

5.4.4 Repetitive control for constant velocity

The data set \mathcal{D}_N that is used to train the GP in Section 5.4.1 contains observations of the disturbance which are obtained through an a priori performed RC experiment at constant velocity. The RC approach in (Mooren et al., 2022d) with GP-based buffer is used, as it can be tuned to be robust for small period variations, moreover, even in the presence of modeling errors, RC can asymptotically learn the periodic part of the error.

The repetitive controller R in the setting in Figure 5.4 is shown in Figure 5.13 and given by

$$R = \frac{LM_{\mathcal{GP}}}{1 - M_{\mathcal{GP}}}, \quad (5.19)$$

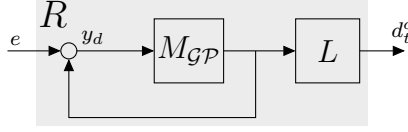


Figure 5.13: Repetitive controller R with Gaussian-process buffer M_{GP} and learning filter L .

where $M_{GP} = \mu^{GP}\Gamma$ is a GP-based buffer that is a FIR filter, Γ is a delay line of size N , and $\mu_{GP} \in \mathbb{R}^{1 \times N}$ are coefficients that are computed as follows

$$\mu_{GP} = \kappa(X_*, X)(\kappa(X, X') + \sigma_n^2 I_N)^{-1}, \quad (5.20)$$

with $X = T_s \cdot [N \ N - 1 \ \dots \ 1]^\top \in \mathbb{R}^N$, $X^* = N \cdot T_s \in \mathbb{R}$ and T_s is the sample time. The same periodic kernel function (5.14) is used, where T is the expected time-domain period.

To incorporate period-time robustness, e.g., if the velocity varies slightly, a larger buffer size N can be used in combination with a locally-periodic kernel, for details, see (Mooren et al., 2022d). Moreover, L is a stable learning filter to ensure stability.

Remark 5.11. Conceptually, alternative RC approaches such as (Steinbuch et al., 2007) that are robust to period-time variations can also be used for this step, however, the presented GP-based approach yields more design freedom. Moreover, the buffer size N in the presented approach does not have to be exactly equal to the number of samples in a single period, as the period is provided in the kernel function and can be any real number.

Theorem 5.12. The system in Figure 5.4 with repetitive controller R in Figure 5.13 and unknown non-linearity $\bar{d}(\phi)$ that is sector bounded as in Assumption 5.2 is asymptotically stable if the following conditions hold,

$$|\mu_{GP}\Gamma(e^{j\omega})(1 - S(e^{j\omega})P(e^{j\omega})L(e^{j\omega}))| < 1 \quad (5.21)$$

$$|P(e^{j\omega})S(e^{j\omega})S_R(e^{j\omega})| < \frac{1}{\gamma}, \quad (5.22)$$

for all $\omega \in [0, \pi]$, with $S_R = (I + PSR)^{-1}$ the modifying sensitivity.

The proof of Theorem 5.12 is given in Section 5.7.

The condition in Theorem 5.12 can also be checked a priori on the basis of a measured frequency response function of P . Moreover, by designing the learning filter L in (5.19) as

$$L \approx (PS)^{-1}, \quad (5.23)$$

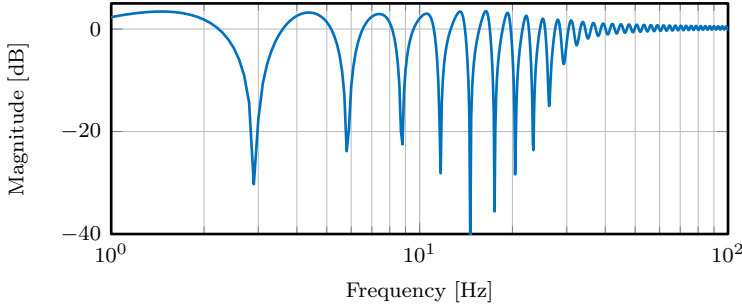


Figure 5.14: Modifying sensitivity function S_R with R designed for the CPA with rotational velocity 0.14 rad/s, such that the disturbance fundamental period equals $T = \frac{2\pi}{0.14 \cdot 131} = 0.343$ seconds. The periodic kernel has equal period, such that R suppresses disturbances at $\frac{1}{0.343} \approx 2.9$ Hz and its harmonics.

the stability condition in (5.21) is satisfied. Many approaches exist to invert the process sensitivity PS , e.g., using stable inversion or zero-phase-error-tracking-control (ZPETC).

Remark 5.13. If PS contains non-minimum phase zeros or delays, then the inverse can be non-causal, i.e., in the case of ZPETC $L = z^{-n_l} L_c$ with $n_l \in \mathbb{N}$ the number of samples preview and L_c the causal part of L . If L is non-causal then GP-based RC is also applicable, for implementation details see (Mooren et al., 2022d).

To show the attenuation properties of R in the setting in Figure 5.13, the disturbance induced error e_d can be written as

$$e_d = -PSS_R d, \quad (5.24)$$

where S_R represents the closed-loop suppression of R in addition to the feedback controller. Consider the following example where S_R is shown for the CPA.

Example 5.14. A repetitive controller R is designed for the CPA that rotates with a constant velocity. The periodic kernel (5.14) is used with the period T seconds that corresponds with the time-domain period of the error, e.g., for 0.14 [rad/s] the period equals $T = \frac{2\pi}{0.14 \cdot 131} = 0.343$.

The resulting modifying sensitivity is shown in Figure 5.14. This shows that R has high disturbance attenuation properties at $\frac{1}{T} \approx 2.9$ Hz and its harmonics corresponding with the period of the torque ripple at this specific velocity, in-between these frequencies the repetitive controller can slightly amplify disturbances.

Example 5.14 shows that indeed S_R is small at frequencies where the disturbance is expected and corresponds with the kernel period T . This is used to

asymptotically reject the disturbance d for constant rotational velocity, even if the model \widehat{PS} is non-perfect, resulting in the signal d_t^c that equals d , i.e., a time-domain model of the disturbance. Together with ϕ during the same experiment, this gives the data set \mathcal{D}_N for GP regression.

5.4.5 Procedure

The following procedure summarizes implementation of the two-step approach that is introduced in this chapter.

Procedure 5.15.

Step 1) Constant velocity learning:

- (a) Identify a model \widehat{PS} of PS and obtain L in (5.23).
- (b) Compute $\mu_{\mathcal{GP}}$ with (5.20), with the period in κ equal to $T = \frac{2\pi}{\nu \cdot 131}$ for the CPA, where ν is the velocity in rad/s.
- (c) Verify stability with Theorem 5.12.
- (d) Perform an RC experiment at constant velocity v and collect data \mathcal{D}_N .

Step 2) GP regression and compensation with \bar{f}_d :

- (a) Compute α with (5.12) and data $\mathcal{D}_N(\phi_N, \hat{d}_N)$ obtained with step 1d, and a set of hyperparameters where T is the spatial period, l a given smoothness and σ_f the variation around the mean.
 - (b) Verify stability with Theorem 5.8.
 - (c) Apply online, i.e., at each sample k , (5.13) for compensation given $\phi(k)$.
-

Remark 5.16. Multiple RC experiments at different velocities can be combined in \mathcal{D}_N to compute α in (5.12). This allows to average over non-repeating disturbances that are present in \mathcal{D}_N , i.e., similar to employing a learning gain in traditional RC (Tomizuka, 2008).

5.5 Application to the CPA

In this section, the experimental results on the CPA are presented. First, the results of applying GP-based RC to the CPA with constant velocity are shown. Second, the GP-based spatial compensation function \bar{f}_d that is computed on the basis of the time-domain RC model is shown. Third, the resulting performance for other constant and varying velocities are presented.

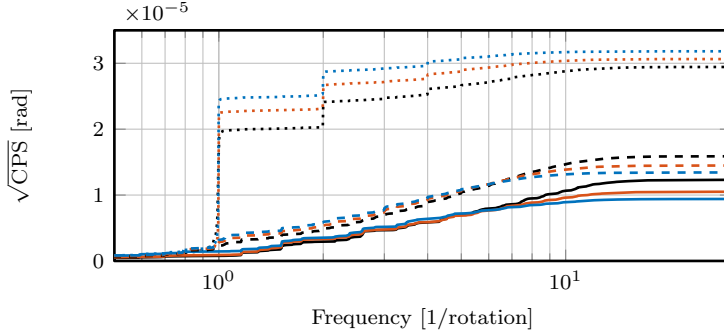


Figure 5.15: Cumulative amplitude spectrum (CAS) of the error with three velocities 0.08 rad/s (—), 0.1 rad/s (—), and 0.12 rad/s (—), for feedback (dotted), GP RC (solid) and the spatial GP compensation (dashed).

5.5.1 RC results at constant velocity

The GP-based repetitive controller R is designed for 3 velocities to, i) experimentally validate that it rejects the disturbance for constant velocities, and ii) generate training data for GP regression. Each RC is designed with the periodic kernel function (5.14) where the period T is modified correspondingly as in Step 2 of Procedure 5.15, and the buffer size N is such that one period fits in the buffer.

The resulting cumulative amplitude spectra (CAS) are shown in Figure 5.15 as a function of the spatial frequency in 1/tooth. The dashed lines show the current state-of-the-art performance with only a feedback controller, the solid lines show the obtained performance after convergence of the repetitive controllers. This shows that the repetitive controller R removes the repeating parts of the error that correspond with 1 1/tooth and its harmonics for constant velocity.

5.5.2 Learned GP compensation and stability

From the GP-based RC experiment at a velocity of 0.1 rad/s the data set \mathcal{D}_N is generated, where $\hat{d} = d_c^t$ contains 279 data points. A fit is made based on these data points with the hyperparameters; smoothness $l = 0.4$, the period equal to the tooth pitch $T = \frac{2\pi}{131}$, $\sigma_f^2 = 0.001$ the approximate amplitude around the mean, and the estimated noise variance $\sigma_v^2 = 6 \cdot 10^{-4}$ such that the confidence bound corresponds with the variance on the data points. The resulting GP-based compensation function that is periodic with a single tooth pitch β is shown in Figure 5.17 including the 99.7% confidence bound.

Stability is assessed with condition (5.18) in Theorem 5.8. First, α in (5.16) is determined in Figure 5.16 yielding that $1/\alpha = 0.07$ and hence $\alpha \approx 14.3$. Second, Figure 5.9 shows that $-\gamma_1 = \gamma_2 = \gamma = 1$ in Assumption 5.2 is a valid assumption

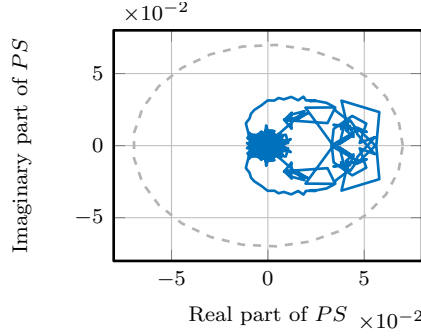


Figure 5.16: Nyquist plot of the process sensitivity PS that is enclosed in a circle $D(-\alpha, \alpha)$ around the origin with radius $1/\alpha = 0.07$.

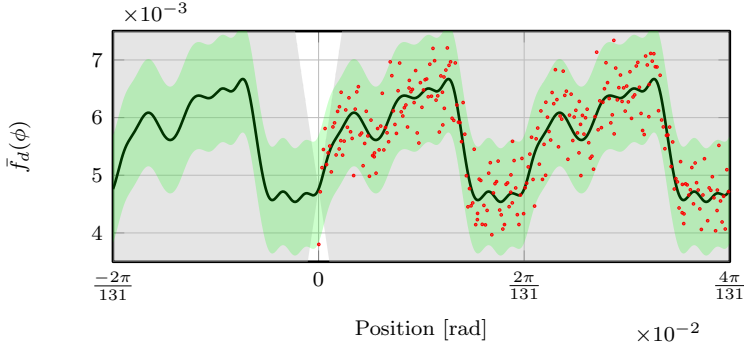


Figure 5.17: Learned GP-based compensation function \bar{f}_d (—) with 99.7 % confidence bound (■), which is trained with data that is obtained with the RC experiment at 0.1 rad/s. The sector bound condition (5.17) with $\rho = 0.5$ is shown in (■).

for the unknown non-linear function $\bar{d}(\phi)$. Third, the GP-based compensation function \bar{f}_d shown in Figure 5.17 satisfies the sector bound condition in (5.17) for $\rho = 0.5$. As a result, $\rho + \gamma = 1.5 \ll \alpha$ which implies that (5.18) in Theorem 5.8 is satisfied and the system is closed-loop stable.

Remark 5.17. The RC output d_t^c contains noise that is potentially amplified by the learning filter L at high frequencies. Therefore, the signal d_t^c is first filtered with a zero-phase low-pass filter with a cut-off frequency of 1000 Hz to remove the main contribution of noise before GP regression.

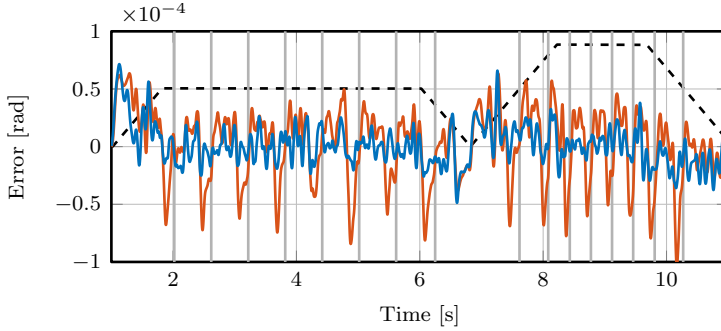


Figure 5.18: Experiment with a varying velocity in (—) (scaled), for the state-of-the-art feedback controller (—) and with the learned GP-based compensation function in (—).

5.5.3 Results

The learned spatial compensation function is employed at different velocities to compensate for the repeating part of the error. The dashed lines in Figure 5.15 show the performance with the GP-based compensation. This clearly shows that the periodic component is almost completely suppressed as desired, although the performance is slightly worse compared to RC for a specific rotational velocity. The RMS errors for each velocity and control strategy are summarized in Table 5.1, yielding performance improvement up to a factor 2.4 with \tilde{f}_d .

In addition, a measurement is carried out where the CPA rotates with a varying velocity for which RC is not applicable. The result is shown in Figure 5.18, where (—) show the error with feedback control and (—) is the error with the learned spatial compensation. This shows that independent of the velocity, the periodic component in the error that repeats with each tooth is significantly attenuated. The overall RMS value of the error is improved by 60 %. Moreover, a substantial part of the remaining error can be attributed to reference-induced errors during acceleration and deceleration which is not compensated for by the present approach.

5.6 Conclusion

Automatic learning of a static compensation function for torque ripple disturbances at arbitrary rotational velocity is enabled in this chapter. By employing a single repetitive control experiment at constant velocity a time-domain model of the disturbance is obtained. Consequently, a Gaussian process is used to learn, on the basis of discrete time-domain data, a spatially periodic and continuous function. The learned function can be used for compensation at arbitrary

Table 5.1: Root Mean Squared (RMS) error in micro-radians and the factor performance improvement with respect to feedback control for the three constant velocities.

Velocity [rad/s]	Feedback [μ rad]	GP RC [μ rad]	GP FB [μ rad]
0.08	29.387	12.412 (2.4)	15.864 (1.9)
0.10	30.628	10.568 (2.9)	14.629 (2.1)
0.12	31.852	9.455 (3.4)	13.547 (2.4)
varying	32.633	n/a	20.021 (1.6)

rotational velocities where the torque ripple is not periodic. Moreover, stability checks are provided using the circle criterion taking into account the non-linearity induced by the obtained commutation update function. The method is experimentally validated on the coarse point assembly (CPA) resulting in a user-friendly automated routine for an average performance improvement of a factor 2 on the CPA at varying velocity, which is a key enabler for FSO communication. This approach can thus successfully be employed during integration of future CPA production series, saving labor-intense testing and calibration steps, and thereby contributing to the cost-effectiveness of the CPA and its applicability in commercial space applications, such as FSO communication. Future research focuses on direct learning or updating of the individual commutation functions on the basis of data.

5.7 Proofs

Proof of Theorem 5.8

The interconnection in Figure 5.3 is reformulated in the standard feedback interconnection in Figure 5.12 with

$$f(\phi) = \bar{f}_d(\phi) - \bar{d}(\phi)$$

and $H = PS$. Subsequently, Lemma 5.7 is used to provide stability conditions. First, the process sensitivity PS is stable by design of C_{fb} . Second, condition (5.16) implies that the Nyquist plot of PS resides in a disk with radius $1/\alpha$ around the origin, denoted by $D(-\alpha, \alpha)$. It remains to show, using the sector bound condition (5.15) in Lemma 5.7, that the interconnection is stable if $\rho + \gamma \leq \alpha$. From the triangular inequality it follows that $|\bar{f}_d(\phi) - \bar{d}(\phi)| \leq |\bar{f}_d(\phi)| + |\bar{d}(\phi)|$. Using Lemma 5.7, the interconnection is globally asymptotically stable if

$$|\bar{f}_d(\phi)| + |\bar{d}(\phi)| \leq \alpha|\phi|. \quad (5.25)$$

Now use Assumption 5.2, i.e., $|\bar{d}(\phi)| < \gamma|\phi|$, to rewrite (5.25) into

$$|\bar{f}_d(\phi)| \leq (\alpha - \gamma)|\phi| = \rho|\phi|, \quad (5.26)$$

or equivalently $-\rho|\phi| \leq \bar{f}_d(\phi) \leq \rho|\phi|$ for $\rho \leq \alpha - \gamma$ which corresponds with (5.18). Finally, as $\bar{f}_d(\phi)$ is periodic with period β , see Definition 5.1, it is sufficient to check the sector bound condition (5.26) on the interval $\phi \in [-\frac{1}{2}\beta, \frac{1}{2}\beta)$ which completes the proof.

Proof of Theorem 5.12

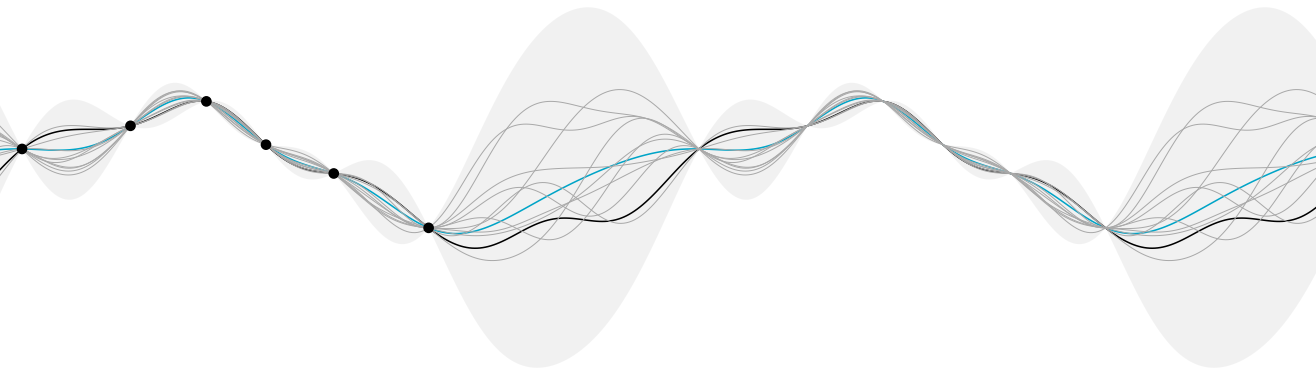
The proof consists of two part, first, (5.21) is derived in (Mooren et al., 2022d, Theorem 2) for stability of the linear part in Figure 5.4, i.e., for $\bar{d}(\phi) = 0$, second, it remains to show that (5.22) implies stability with $\bar{d}(\phi) \neq 0$ under Assumption 5.2 with Lemma 5.7. The setting in Figure 5.4 is reformulated in the standard feedback interconnection in Figure 5.12, yielding that

$$H = P(1 + P(C + R))^{-1} = PSS_R,$$

and $f(\phi) = \bar{d}(\phi)$. Using Lemma 5.7 and the sector bound condition in Assumption 5.2 the interconnection is stable if the Nyquist plot of H resides inside the disk $D(\gamma, -\gamma)$. This is equivalent to (5.22) which completes the proof.

Part III

Feedforward tuning for flexible motion tasks



Feedforward Motion Control: From Batch-to-Batch Learning to On-line Parameter Estimation¹

Abstract: Feedforward control can significantly improve tracking performance for high-precision motion systems. The aim of this paper is to develop a unified framework for both batch-wise and for on-line learning of feedforward parameters using least squares estimation. A statistical analysis is employed to analyze the effect of noise in a closed-loop setting. This provides new insights, both potential advantages as well as possible hazards of on-line estimation are considered, i.e., biased estimates can be obtained that hamper feedforward performance. Finally, a practical implementation is presented to mitigate the effect of noise, and theoretical results are validated through simulation and experiments on a benchmark motion system.

¹The results in this chapter constitute Contribution 5 of this thesis. The chapter is based on Mooren, N. et al. (2019a). Feedforward motion control: From batch-to-batch learning to online parameter estimation. In *2019 American Control Conference (ACC)*. IEEE, pages 947–952 and Mooren, N. et al. (2019b). From Batch-to-Batch to online learning control: experimental motion control case study. *IFAC-PapersOnLine*, 52(15):406–411.

6.1 Introduction

Feedforward control, ranging from manual tuning to learning control, is essential for enhancing positioning performance of motion systems. Manual feedforward tuning enables performance improvement by anticipating for known exogenous disturbances, where typically the to be applied reference is used (Lambrechts et al., 2005). For systems with repeating motion tasks, learning feedforward algorithms such as iterative learning control (ILC) are able to automatically learn from previous tasks, to compensate for the repeating contributions in the error (Bristow et al., 2006; van der Meulen et al., 2008). These methods benefit from preview, allowing non-causal feedforward controllers to compensate for disturbances before these affect the system, resulting in a major performance improvement compared to causal controllers (Van Zundert and Oomen, 2017).

Despite potential performance improvement of learning control, standard approaches may actually lead to a performance degradation in typical motion systems, where varying tasks are commonly performed. Standard learning algorithms generate a feedforward signal that exactly compensates for trial-invariant disturbances during a specific task (Gao and Mishra, 2014; van de Wijdeven and Bosgra, 2010). However, many motion systems perform non-repeating motion tasks, e.g., semiconductor wire-bonding (Boeren et al., 2016), lithography (Blanken et al., 2017a) and printing (Bolder et al., 2017). Hence, flexibility towards trial-varying references is essential, whereas current learning feedforward algorithms are generally highly sensitive to trial-varying exogenous signals (Hoelzle et al., 2011).

ILC algorithms that are flexible against varying tasks have been developed (Bolder et al., 2017; Hoelzle et al., 2011; van de Wijdeven and Bosgra, 2010). These methods combine model-based feedforward and ILC, resulting in flexible learning feedforward. The central idea in this method is to postulate a controller parameterization and learn the parameters using common principles from ILC (Bristow et al., 2006). From the perspective of system identification, flexible feedforward tuning techniques have been developed which essentially replaces the learning filter design in ILC by an estimation step. In Boeren et al. (2015), extended parameterizations are explored, encompassing joint input shaping and feedforward control. For both learning and identification methods batch-wise approaches are exploited, i.e., controller parameters are updated after each experiment, where a key aspect is to obtain unbiased estimates. In Blanken et al. (2017a), these learning and identification approaches are compared, directly connecting iterative feedforward tuning to inverse system identification.

Although important progress has been made in learning and identification for feedforward control for a class of tasks, these approaches typically consider a batch-wise operation. Consequently, performance improvement takes place after each task is completed. The aim of this chapter is to develop a framework for current-iteration feedforward tuning, i.e., updating the feedforward controller

during a motion task for immediate performance improvement, for general controller structures as in Boeren et al. (2014). The main contribution in the present chapter is to propose a unified framework for learning feedforward control, covering both batch-wise approaches as the development of current-iteration learning. The following sub-contributions are addressed in this work;

- C1 a detailed statistical analysis of the proposed framework is provided to show that biased estimates are obtained in earlier approaches,
- C2 a practical implementation for on-line feedforward tuning using RLS that reduces the effect of noise leading to biased estimates,
- C3 a simulation case study is performed to confirm the theoretical conclusions, including the immediate benefit of direct learning, and
- C4 experimental validation on a benchmark motion system.

Indeed, theoretical conclusions in this chapter may provide relevant new perspectives on earlier related approaches, including Butler (2012), which are further exploited.

This chapter is organized as follows. In Section 6.2, the general framework for feedforward control is covered and a parameter optimization problem is proposed. In Section 6.3, a theoretical analysis of the proposed parameter optimization in the presence of noise is provided. In Section 6.4, a practical solution is presented to improve the estimation in presence of noise. In Section 6.5, a simulation study is conducted in which the proposed framework is applied to a benchmark system, where the cases with and without noise are compared. In Section 6.6 experimental validation is provided. Finally, conclusions and ongoing research are provided in Section 6.7.

6.2 Towards parameter optimization for feedforward control

In this section, a generic framework is developed for both batch-to-batch and for on-line learning of feedforward parameters. First, a feedforward controller parameterization is presented for which the estimation problem has an analytic solution, i.e., through linear least squares optimization. Second, the connection is established between feedforward controller tuning and parameter estimation from a system identification point of view. Finally, solutions are provided both for a batch-wise setting as in an on-line setting to compute the feedforward parameters.

6.2.1 Problem definition in a unified framework

Consider the general control configuration in Figure 6.1 consisting of an input shaper C_r , feedforward controller C_{ff} , feedback controller C_{fb} , and true system P_0 that is linear time-invariant (LTI). The system, mapping the input u to the noise-free output y_0 , is given by

$$P_0(q^{-1}, \theta_0) = \frac{B(q^{-1}, \theta_b^0)}{A(q^{-1}, \theta_a^0)} \quad (6.1)$$

with $A(q^{-1})$ and $B(q^{-1})$ polynomials in the backwards shift operator q^{-1} , and θ_a^0 and θ_b^0 are the true system parameters. Furthermore, it is assumed that $A(q^{-1})$ and $B(q^{-1})$ are given by

$$\begin{aligned} A(q^{-1}, \theta_a) &= \sum_{i=1}^{n_a} \psi_i(q^{-1}) \theta_{a,i} \\ B(q^{-1}, \theta_b) &= \sum_{i=n_a+1}^{n_b+n_a} \psi_i(q^{-1}) \theta_{b,i} \end{aligned} \quad (6.2)$$

in which $\psi(q) \in \mathbb{R}[q^{-1}]$ are referred to as basis functions. Finally, the output $y_m = y_0 + v$ that is used for parameter estimation contains additive noise $v \sim \mathcal{N}(0, \sigma_v^2)$ that is zero-mean independent and identically distributed (i.i.d) with variance σ_v^2 .

The scheme in Figure 6.1 includes both feedback and feedforward control. The goal of the feedforward controller is to anticipate for known or repeating exogenous disturbances that act on the system, whereas the feedback controller attenuates unknown disturbances and model errors. In the remainder of this work, the focus is on feedforward control, i.e., optimization of $C_{ff}(\theta)$, since this is the main contribution to the tracking performance. The goal of the feedforward controller is defined as follows.

Definition 6.1 (Feedforward control goal). *Determine a feedforward controller such that the reference induced error given by*

$$e = S(C_r - PC_{ff})r, \quad (6.3)$$

with $S = (I + PC_{fb})^{-1}$ is minimized.

A well known result from classical feedforward control is that the plant inverse must be reflected in the feedforward controller (Lambrechts et al., 2005), which is also the case for the proposed combined input shaper and feedforward controller. By satisfying

$$C_{ff}(q, \theta_a) C_r^{-1}(q, \theta_b) = P^{-1}(q, \theta_0) \implies e(t) = 0 \forall t \quad (6.4)$$

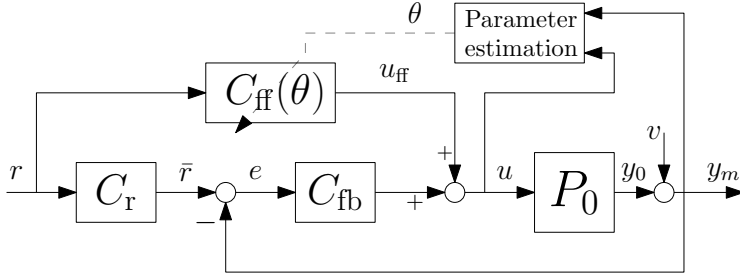


Figure 6.1: General control scheme with feedback controller C_{fb} , feedforward controller C_{ff} , input shaper C_r and plant P_0 .

it follows that the reference induced positioning error is eliminated. This can also be obtained by setting $C_r = I$ and directly parameterizing C_{ff} as P^{-1} , i.e., as a rational filter. However, identifying C_{ff} directly as rational filter yields a non-convex optimization problem, see, e.g., Blanken et al. (2017a). Moreover, direct inversion for system with non-minimum phase zeros might lead to internal stability issues and non-causal feedforward control, see, e.g., Van Zundert and Oomen (2017).

6.2.2 Feedforward controller parameterization

Consider the following feedforward controller parameterization, directly establishing the connection between the system parameters and the ideal feedforward parameters.

Definition 6.2 (Feedforward controller parameterization). *The input shaper and feedforward controller are parameterized as follows.*

$$\mathcal{C} = \left\{ (C_r, C_{ff}(\theta)) \mid \begin{array}{l} C_{ff}(\theta) = A(q^{-1}, \theta) \\ C_r = B(q^{-1}) \end{array}, \theta \in \mathbb{R}^{n_a} \right\} \quad (6.5)$$

Where $C_{ff}(\theta)$ is parameterized as function of the to be optimized parameters θ . Next,

$$\Psi = [\psi_1 \quad \psi_2 \quad \dots \quad \psi_{n_a}]^\top, \quad (6.6)$$

and to be estimated parameters

$$\theta = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_{n_a}]^\top \in \mathbb{R}^{n_a} \quad (6.7)$$

such that $C_{ff}(q^{-1}, \theta) = \Psi^\top \theta$.

Moreover, consider the following condition in the input shaper to avoid scaling of the reference, see, e.g., Boeren et al. (2014).

Assumption 6.3. *The input shaper satisfies that*

$$C_r(q^{-1})|_{q^{-1}=1} = 1. \quad (6.8)$$

Remark 6.4. *In this chapter, the focus is on optimization of $C_{\text{ff}}(\theta)$ and C_r is assumed to be fixed as in Figure 6.1, to avoid cumbersome notation. Conceptually, the following approach and analysis is similar for the general case, i.e., with $C_r(\theta)$. In the remainder of this work the notation θ refers to θ_a , similar for θ_0 referring to θ_a^0 and, $B(q^{-1}, \theta_b)$ is referred to as $B(q^{-1})$ assuming that θ_b is known.*

Using the parameterization (6.5) the reference induced error becomes

$$e = S(q^{-1}) \left(B(q^{-1}) - \frac{B(q^{-1})}{A(q^{-1}, \theta^0)} A(q^{-1}, \theta) \right) r, \quad (6.9)$$

hence, if $\theta \rightarrow \theta^0$, then the positioning error $e \rightarrow 0$, such that (6.4) is satisfied.

6.2.3 Parameter optimization for feedforward control

In the remainder of this section, an optimization problem is formulated to optimize $C_{\text{ff}}(\theta)$. The optimization is based on linear least squares, for which analytic solutions are provided for both the batch-wise case and a recursive solution for the on-line case.

Definition 6.5 (Parameter optimization problem). *Given measurement data sequences $\{u\}$ and $\{y_m\}$, determine θ^* such that the feedforward controller (6.5) minimizes the reference induced tracking error.*

Consider the following optimization problem for computation of θ^*

$$\hat{\theta}(k) = \min_{\theta} V(C_{\text{ff}}(\theta)), \quad (6.10)$$

where the objective function is of the form

$$V(\theta, k) = \frac{1}{2} \sum_{i=1}^k \epsilon^2(i) \quad (6.11)$$

$$\epsilon(k) := \bar{u}(k) - \phi(k)\theta \quad (6.12)$$

and the residual function $\epsilon(k) \in \mathbb{R}$ is linear in the parameter θ . Moreover, the regressor $\phi(k) \in \mathbb{R}^{1 \times n_a}$ is a filtered version of the measured output data with the basis functions,

$$\phi(k) = \Psi^\top(q)y_m(k) \quad (6.13)$$

and similar for $\bar{u}(k) \in \mathbb{R}$ that is the filtered input

$$\bar{u}(k) = B(q)u(k), \quad (6.14)$$

as depicted in Figure 6.2.

To show that θ converges to θ_0 if $v = 0$, note that the objective function (6.11) is quadratic and ϵ is linear in θ . Hence, a sufficient condition for the optimum is that the gradient of (6.10), given by

$$\nabla V(\theta, k)|_{\theta=\theta_0} = \sum_{i=1}^k \epsilon(i, \theta) \frac{\partial \epsilon(i, \theta)}{\partial \theta} \bigg|_{\theta=\theta_0} = 0 \quad (6.15)$$

is zero. To show that this is the case for $\theta \rightarrow \theta_0$, substitute $\phi(k)$ and $\bar{u}(k)$ in (6.12) to obtain

$$\epsilon(k) = B(q)u(k) - \Psi^\top \theta y_m(k) \quad (6.16)$$

$$= B(q)u(k) - A(q, \theta)y(k), \quad (6.17)$$

combining this with (6.1) results in,

$$\epsilon(k, \theta)|_{\theta=\theta_0} = 0. \quad (6.18)$$

This also render the gradient in (6.15) to zero for $v = 0$, which implies that θ_0 is the optimum of the cost function as desired. The case where $v \neq 0$ is analyzed in detail in the following section.

In the following two subsection, an analytic solution to the optimization problem (6.10) is derived for both a batch-wise setting as for the on-line setting.

B.1) Batch-to-batch parameter optimization

In a batch-wise optimization setting, an experiment or task is performed from which a batch of data is collected (Blanken et al., 2017a). The data collected during task nr j , denoted with $\{u^j\}$ and $\{y_m^j\}$ is used to optimize $C_{\text{ff}}(\theta)$ for the next task, i.e., determine θ^{j+1} and update the controller.

Next, it is shown that for batch-wise optimization, the optimization problem (6.10) is equivalent to a linear least squares problem. To this extend, the proposed residual function (6.12) is rewritten in matrix form using data from task j resulting in the following notation,

$$\mathcal{E} = \bar{U} - \Phi\theta \quad (6.19)$$

with $\mathcal{E} \in \mathbb{R}^k$, $\bar{U} \in \mathbb{R}^k$, $\Phi \in \mathbb{R}^{k \times n_a}$ given by

$$\mathcal{E} = \begin{bmatrix} \epsilon(1) \\ \epsilon(2) \\ \vdots \\ \epsilon(k) \end{bmatrix}, \Phi = \begin{bmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(k) \end{bmatrix}, \bar{U} = \begin{bmatrix} \bar{u}(1) \\ \bar{u}(2) \\ \vdots \\ \bar{u}(k) \end{bmatrix}$$

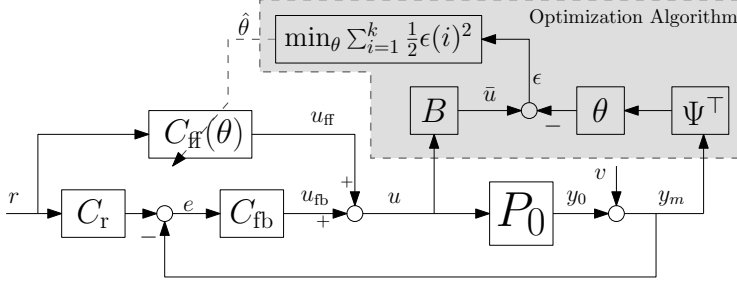


Figure 6.2: Combined framework for feedforward control and parameter optimization.

such that the objective function is alternatively written as $V(\theta) = \frac{1}{2} \mathcal{E}^\top \mathcal{E}$, then the optimization problem (6.10) is equivalent to the least squares problem

$$\bar{U} = \Phi \theta \quad (6.20)$$

with analytic solution

$$\hat{\theta} = (\Phi^\top \Phi)^{-1} \Phi^\top \bar{U}. \quad (6.21)$$

Assumption 6.6. Assume that either $u(k)$, in the open-loop setting, or $r(k)$ in the closed-loop setting, is persistently exciting such that the matrix $\Phi^\top \Phi$ is non-singular.

Consider the following procedure to optimize the feedforward parameters in a batch-to-batch setting.

Procedure 6.7 (Batch-to-batch parameter optimization).

1. Collect input-output data $\{u^j\}$ and $\{y_m^j\}$ from the current task.
2. Determine \bar{U}^j and Φ^j .
3. Compute $\hat{\theta}^{j+1} = ((\Phi^j)^\top \Phi^j)^{-1} (\Phi^j)^\top \bar{U}^j$.
4. Update the controller parameters, i.e., determine $C_{ff}^{j+1}(\hat{\theta}^{j+1})$, and continue with step 1 for task $j + 1$.

B.2) Current-iteration parameter optimization

In the previous section, a batch-wise solution for the optimization problem (6.10) is outlined. A disadvantage of this method is that the controller can only be updated in-between tasks, i.e., performance improvement after each task. In

this section, an on-line optimization is proposed based on recursive least squares (RLS), which allows to optimize the parameters efficiently during each iteration, enabling performance improvement during a task.

The recursive equivalent of the solution to (6.10) is given by

$$\theta(k) = \theta(k-1) + K(k) (\bar{u}(k) - \phi^\top \theta(k-1)) \quad (6.22)$$

in which the time dependent learning gain K is given by

$$K(k) = P(k)\phi(k) \quad (6.23)$$

and $P(k)$ is recursively computed as follows

$$P(k) = P(k-1) [I - \phi(k)\Sigma\phi(k)^\top P(k-1)] \quad (6.24)$$

where

$$\Sigma = (I + \phi(k)^\top P(k-1)\phi(k))^{-1}. \quad (6.25)$$

with initial conditions $P(t_0) = (\Phi(t_0)^\top \Phi(t_0))^{-1}$ and $\theta(t_0)$. A detailed derivation of the RLS algorithm can be found in (Åström and Wittenmark, 2013, Chapter 2), (Goodwin and Sin, 2014). Next, the following procedure is proposed for on-line optimization of the parameter using the RLS algorithm in (6.22) - (6.25)

Procedure 6.8 (On-line parameter optimization).

1. Define an initial parameter estimate $\theta(t_0)$ and initial condition $P(t_0)$.
2. At time k compute the learning gain $K(k)$ using (6.23) - (6.25).
3. Compute the parameters $\theta(k)$ using (6.22).
4. Update the controller $C_{\text{ff}}(\theta(k))$ using $\theta(k)$ and start at step 2 for the next iteration.

To conclude this section, an optimization problem is proposed for feedforward controller tuning and solutions for both batch-to-batch and on-line settings have been provided. The main advantage of on-line optimization is that performance can directly be improved instead of improvement after each iteration as in a batch-wise setting. However, both methods rely on obtaining an unbiased estimate of the optimal tuning parameters in order to minimize the positioning error (Bitmead and Anderson, 1980), as given in Definition 6.2. Therefore, in the next section a statistical analysis of the proposed optimization problem is provided in the presence of measurement noise.

6.3 Bias analysis in presence of noise

In this section, the estimator proposed in section 6.2 is further analyzed in the setting in Figure 6.1, i.e., when measurement noise is present and in a closed-loop setting.

Remark 6.9. *The considered problem related to the errors-in-variables setting (Söderström and Stoica, 1989), where noise is present in both the measured signals as in the regressor signal. In the considered setting only y_0 is subject to additive noise.*

The optimization problem (6.10) is equivalent to the least squares problem (6.20) and is further analyzed in the presence of noise. Therefore, the optimal estimate θ^* that minimizes the cost function, i.e.,

$$\left. \frac{\partial V(\theta)}{\partial \theta} \right|_{\theta=\theta^*} = 0 \quad (6.26)$$

is further analyzed to provide insight in which cases a biased estimate is obtained. Computing the gradient of $V(\theta)$ and equating this to zero

$$\frac{\partial V(\theta)}{\partial \theta} = -\Phi^\top (\bar{U} - \Phi\theta) = 0 \quad (6.27)$$

gives the optimal parameter estimate (6.21).

Before stating the main result in this work consider the following parameterization of the basis functions $\psi(q)$. The n^{th} basis function is of the form

$$\psi_n(q) = \sum_{j=0}^{n_\psi} a_n^j q^{-j} \quad (6.28)$$

where n_ψ is the highest order of all n_a basis functions and $a_n^j \in \mathbb{R}$ is the j^{th} coefficient corresponding to the n^{th} basis function.

Theorem 6.10 (Bias of the estimator). *The expected value of the optimal estimate $\hat{\theta}$ as function of y_0 and v is given by*

$$\mathbb{E} \hat{\theta} = [\Gamma_a^\top (\mathbf{R}_{yy} + 2\mathbf{R}_{yv} + \mathbf{R}_{vv}) \Gamma_a]^{-1} \cdot \Gamma_a^\top (\mathbf{R}_{yy} + \mathbf{R}_{yv}) \Gamma_a \theta_0 \quad (6.29)$$

where Γ_a is a matrix consisting of all coefficients corresponding to the n_a basis functions.

$$\Gamma_a = \begin{bmatrix} a_1^0 & \dots & a_{n_a}^0 \\ \vdots & & \vdots \\ a_1^{n_\psi} & \dots & a_{n_a}^{n_\psi} \end{bmatrix} \in \mathbb{R}^{n_\psi \times n_a} \quad (6.30)$$

Furthermore, the matrices $\mathbf{R}_{yy} \in \mathbb{R}^{n_\psi \times n_\psi}$ and $\mathbf{R}_{vy} \in \mathbb{R}^{n_\psi \times n_\psi}$ are the autocorrelation matrix and cross-correlation matrix on the basis of $k \rightarrow \infty$ data samples respectively. Also note that the autocorrelation matrix of the noise $\mathbf{R}_{vv} = \sigma_v^2 I_{n_\psi}$.

Proof. The proof is given in Section 6.8. \square

Theorem 6.10 allows to further analyze the setting where input-output data is collected in a closed-loop fashion, and on-line estimation is used to determine the feedforward parameter as in Figure 6.2, which is also exploited in Butler (2012). First, note that

$$y_0 = \text{SPC}_{\text{fb}}(r - v) \quad (6.31)$$

implying that correlation is present between y_0 and v due to feedback. Next using Theorem 6.10 it becomes evident that the obtained estimates will be biased due to measurement noise, since both $\mathbf{R}_{vy} \neq 0$ and $\mathbf{R}_{vv} \neq 0$ resulting in $\mathbb{E} \hat{\theta} \neq \theta_0$. This leads to severe performance degradation of not accounted for as will be shown in section 6.5.

In the following Corollaries, the ideal setting without noise and the case where input-output data is collected in open-loop with noise are considered.

Corollary 6.11 (Noise free optimization). *Consider the situation where measurement noise is not present, hence the noise related terms in (6.29) vanish resulting in*

$$\mathbb{E} \hat{\theta} = [\Gamma_a^\top \mathbf{R}_{yy} \Gamma_a]^{-1} \Gamma_a^\top \mathbf{R}_{yy} \Gamma_a \theta_0 = \theta_0. \quad (6.32)$$

From which it follows that the estimate $\hat{\theta}$ is an unbiased estimate of θ_0 if noise is not present.

Corollary 6.12 (Noisy optimization in open-loop). *Consider the situation where noise is present and the input output data is collected in open-loop, i.e., without a feedback controller being present. This implies that v and y_0 are uncorrelated using (6.31), i.e., $\mathbf{R}_{vy} = 0$, therefore the estimate (6.29) simplifies to*

$$\mathbb{E} \hat{\theta} = [\Gamma_a^\top (\mathbf{R}_{yy} + \sigma_v^2 I_{n_\psi}) \Gamma_a]^{-1} \Gamma_a^\top \mathbf{R}_{yy} \Gamma_a \theta_0. \quad (6.33)$$

This shows that the estimate becomes biased $\mathbb{E} \hat{\theta} \neq \theta_0$ in open-loop due to the effect of measurement noise.

From this analysis it follows that noise does influence the estimate in both the open-loop case as in a closed-loop case. In the following section, a practical approach is presented to mitigate the effect of noise and reduce the bias. Finally, a case-study and experimental validation is provided to illustrate the necessity of appropriately dealing with measurement noise in estimation problems to be able to successfully implement feedforward control.

6.4 Towards unbiased estimates for feedforward

In this section, a practical approach is presented for the implementation of Procedure 6.8 in an on-line setting and in presence of noise to reduce the bias.

6.4.1 Reducing estimation bias

Because noise is mainly dominating in the high-frequency range and performance is usually required up till a certain bandwidth, the effect of noise can be reduced by filtering both the input u and the output y_m with a low-pass filter before using it for the least squares estimation. To this extend, replace $\bar{u}(k)$ and $\phi(k)$ in (6.12) by

$$\tilde{u}(k) = F(z)\bar{u}(k) \quad (6.34)$$

$$\tilde{\phi} = F(z)\phi(k) \quad (6.35)$$

with $F(z) \in \mathcal{RH}_\infty$ a stable and causal filter of the form

$$F(z) = \left(\frac{1 - e^{2\pi f_c T_s}}{z - e^{2\pi f_c T_s}} \right)^n, \quad (6.36)$$

with n the largest order of the basis functions Ψ in C_{ff} and f_c the cut-off filter. Using this addition filter F yields that

$$\bar{\epsilon}(k, \theta) = F\epsilon(k, \theta) - F\Psi^\top \theta v(k) \quad (6.37)$$

with ϵ in (6.12). Hence, the low-pass filter F is ideally small in the high-frequency range where noise is present. However, note that F modifies the cost function that is solved, i.e., it reduces the effect of noise and thereby reduces the noise-bias, at the cost of introducing a bias through F which is in general assumed to be less than the noise bias. It is shown next, through simulations and experiments, that this approach significantly improves the estimation performance.

6.5 Illustrative case study

Typical motion systems including; wafer stages, printers and pick-and-place robot, are described by rigid-body behavior in addition to flexible modes (Gawronski, 2004)

$$P(s) = M^{-1} \frac{1}{s^2} + \sum_{i=1}^{\infty} \frac{(c_{mqi} + sc_{mvi})b_{mi}}{s^2 + 2\zeta_i\omega_i s + \omega_i^2}. \quad (6.38)$$

An example of a typical frequency response function of such a motion system is depicted in Figure 6.3, where clear rigid body dynamics are present until the

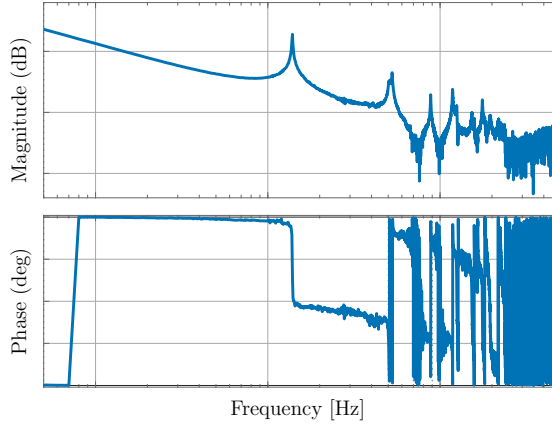


Figure 6.3: Frequency response function of a typical motion system.

first flexible mode. For controller design, the bandwidth is usually a factor 3 below the first flexible mode, i.e., where the rigid-body behavior is dominant. Hence, rigid-body dynamics are leading for performance, whereas flexible modes are taken into account for stability reasons.

6.5.1 Plant and feedback controller

In the remainder of this section, simulations are performed to confirm theoretical conclusions, i.e., illustrate the effect of measurement noise on the tracking performance. For the sake of illustration a single mass system $P(s) = 1/ms^2$ is used. First, consider the following discretized transfer function of P

$$P(z^{-1}) = \frac{T_s^2}{2} \frac{(z^{-1} + z^{-2})}{m_0(1 - 2z^{-1} + z^{-2})}, \quad (6.39)$$

where $T_s = 0.001$ is the sample time and $m_0 = 2$ the mass. Second, a feedback controller is designed

$$C_{fb}(s) = 1.9 \cdot 10^8 \times \frac{s + 15.71}{s^2 + 1.3 \cdot 10^4 s + 3.2 \cdot 10^6} \quad (6.40)$$

realizing a bandwidth of 10 Hz. Where the bandwidth is defined as the cross-over frequency of the open-loop transfer function, denoted with $L = PC_{fb}$. Next, consider the feedforward parameterization for this example to obtain the setting in Figure 6.2.

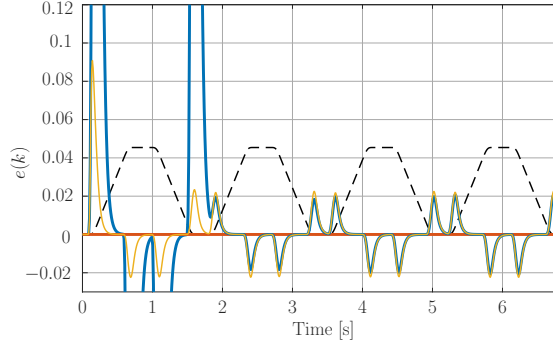


Figure 6.4: Positioning error for the case; on-line learning with noise (—), online learning without noise (—) and batch-wise learning with noise (—). Furthermore, a scaled version of the reference (—) is given.

6.5.2 Feedforward controller

According to Definition 6.2, the input shaper is given by the plant nominator polynomial, i.e.,

$$C_r(q) = \frac{1}{2}(q^{-1} + q^{-2}) \quad (6.41)$$

and $C_{ff}(q, \theta) = \psi(q)\theta$ with basis function

$$\psi(q) = \frac{1 - 2q^{-1} + q^{-2}}{T_s^2}, \quad (6.42)$$

that depends on the plant denominator polynomial. The goal is to optimize the feedforward controller parameter θ using the optimization problem (6.10), in which $\phi(k) = \Psi^T y(k)$ where Ψ consists of the single basis function (6.42) in this example, i.e., $\phi(k) = \psi(q)y_m(k)$.

6.5.3 Results and conclusions

The feedforward controller is optimized on-line using Procedure 6.7, and simulations are conducted with and without noise to shown the influence. Furthermore, batch-wise optimization is included to show the benefit of direct learning. The reference is taken as a sequence of fourth-order point-to-point tasks, see Figure 6.4, and white noise is used with variance $\sigma_v = 10^{-13}$. The resulting parameter estimates are depicted in Figure 6.5, and the obtained positioning error in Figure 6.4.

From these results the following observations can be made.

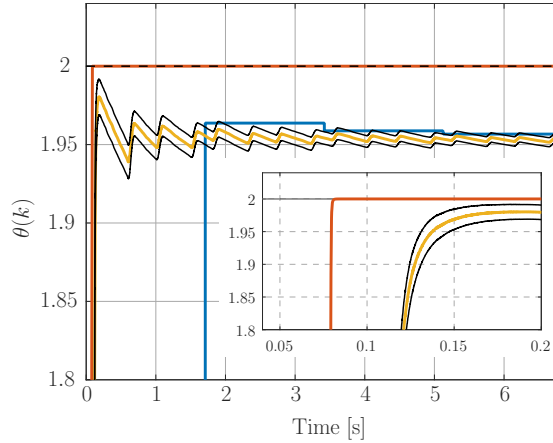


Figure 6.5: On-line parameter estimates, i.e., θ^{noise} with noise (—) with 99.7 % confidence interval (—), without noise $\theta^{\text{noise-free}}$ (—) and the true parameter θ_0 (- -). Batch-wise estimate is also provided (—). A zoom for $t \in [0.04, 0.2]$ shows the convergence of $\theta^{\text{noise-free}}$.

- In the noise-free case, the feedforward parameter converges to the true mass of the system, i.e., $\theta^{\text{noise-free}} \rightarrow \theta_0$. This confirms that an unbiased estimate is obtained if measurement noise is not present, as shown in Corollary 6.11. Note that after a small time interval, (6.4) is satisfied since the positioning error converges to zero.
- In case that measurement noise is present, although the variance decreases, the parameter estimate does not converge to the true system parameter $\theta^{\text{noise}} \not\rightarrow \theta_0$, i.e., the obtained estimate is biased confirming the theoretical conclusions in Section 6.3. It can be seen that due to the biased estimate the positioning error remains non-zero, i.e., (6.4) is not satisfied.
- The immediate benefit of direct learning is observed. For the batch-wise implementation a large error is present during the first task, whereas, for the on-line approach the estimate converges within a fraction of a task, see zoom in Figure 6.5. Hence, the on-line approach is more robust towards variations in system parameters.

Finally, it can be concluded that the simulations confirm the theoretical conclusions in this chapter. Next, an experimental case study is performed for validation and to show that with the practical solution in Section 6.4 the performance is significantly improved.

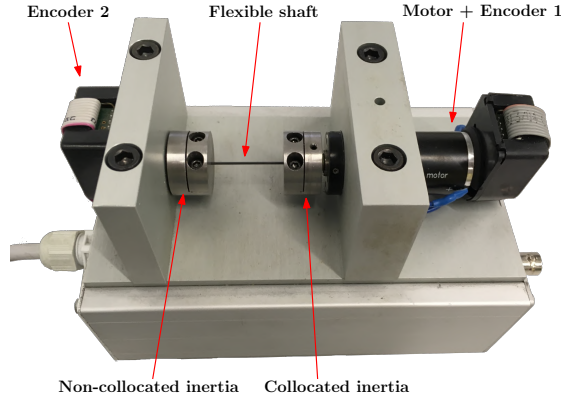


Figure 6.6: Experimental setup.

6.6 Experimental validation

In this section, the feedforward optimization method is applied to a benchmark motion system. In the previous section, the method is validated in a controlled situation, however, during the experiments many undesired effects are present including; sensor quantization, measurement noise, delays and model mismatches $P_0 \notin \mathcal{C}$. The aim of this experimental case study is to show the potential of this method in a practical situation.

6.6.1 Experimental setup

The experimental setup is depicted in Figure 6.6, it consists of two rotating inertias J_1 and J_2 connected with a flexible shaft. The combined inertia is approximately given by $3.68 \cdot 10^{-4} \text{ kg m}^2$. The rotation of both inertias is measured using incremental encoders with a resolution of $2\pi/2000$ radians. The control goal is to minimize the reference induced positioning error (6.3) of the non-collocated inertia J_2 . The following feedback controller is used

$$C_{fb} = \frac{0.83(z+1)(z-0.9898)}{z-0.8575)(z-0.8314)}, \quad (6.43)$$

resulting in an open-loop bandwidth of approximately 15 Hz.

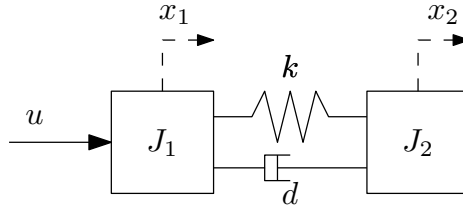


Figure 6.7: Two mass-spring-damper system.

6.6.2 Feedforward controller optimization

A simplified representation of the setup is given in Figure 6.7, where the transfer from u to the position x_2 is given by

$$P_{\text{ncol}}(s) = \frac{1}{s^2} \frac{ds + k}{J_1 J_2 s^2 + (J_1 + J_2)ds + (J_1 + J_2)k}. \quad (6.44)$$

By assuming that damping is small this becomes

$$P_{\text{ncol}}(s) \approx \frac{1}{\frac{J_1 J_2}{k} s^4 + (J_1 + J_2)s^2}. \quad (6.45)$$

Hence, to minimize the positioning error, the feedforward controller C_{ff} should approximate the plant denominator. Therefore, the following two basis functions are considered dominant and used in C_{ff} ,

$$\psi_1(q^{-1}) = \left(\frac{1 - q^{-1}}{T_s} \right)^2, \quad \psi_2(q^{-1}) = \left(\frac{1 - q^{-1}}{T_s} \right)^4 \quad (6.46)$$

which can be seen as an acceleration and snap feedforward term. Furthermore, the input shaper is given by

$$C_r(z) = \frac{1}{4} \frac{z^2 + 2z + 1}{z^3} \quad (6.47)$$

which is a scaled version of a fit of the system numerator, that satisfies (6.8), and contains 3 samples delay. The cut-off frequency for the low-pass filter $F(z)$ is set to $f_c = 25$ Hz, which is slightly above the bandwidth of 15 Hz. Next, procedure 6.8 is executed with initial parameter estimates $\theta(t_0) = [0 \ 0]^T$ and initial matrix $P(t_0) = 10^{-5} \cdot I_{2 \times 2}$. The reference is a fourth-order point-to-point motion of which the velocity is depicted in Figure 6.9.

6.6.3 Experimental results

The experimental results are shown in frequency-domain with a focus on the estimation, and in time-domain to show the performance improvement.

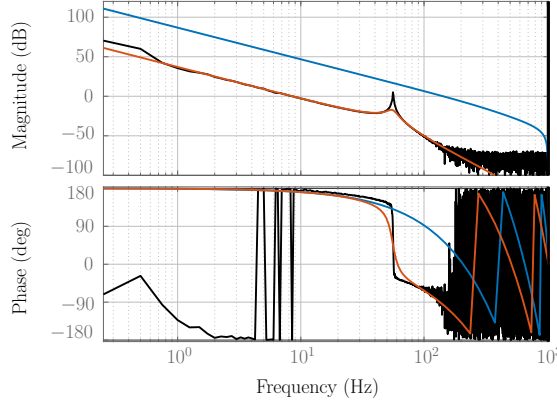


Figure 6.8: FRF of the experimental setup (—), and experimentally identified model $C_r C_{ff}^{-1}(\theta)$ with $F = 1$ (—) and with $f_c = 25$ Hz (—).

Frequency-domain results: Two experiments are performed, during the first experiment $F = 1$ (—), during the second experiment $f_c = 25$ (—) as proposed earlier. The obtained estimates are compared to the frequency response function (FRF) of the setup, see Figure 6.8. This shows that without the use of a low-pass filter a poor estimate is obtained, whereas with the low-pass filter a good estimate of the setup is achieved. It is interesting to note that, although the low-pass filter has a cut-off frequency of $f_c = 25$ Hz the resonance frequency of the plant around 58 Hz is still estimated closely. This is explained using the parametric model in (6.44) of which the resonance frequency is given by

$$\omega_{\text{res}} = \sqrt{\frac{(J_1 + J_2)k}{J_1 J_2}} \quad (6.48)$$

being a combination of the inertia and stiffness. Hence, by estimating the compliant contributions of the acceleration and snap terms as in (6.45), the resonance frequency is estimated closely. Furthermore, it is observed that the damping is not properly estimated because it is assumed to be zero and thus not adapted.

Time-domain results: To show the benefit of on-line learning in the time-domain, also two experiments are performed. During the first experiment only feedback control is used (—) and in the second experiment the proposed feedforward controller optimization is included (—), see Figure 6.9. The results show a significant performance improvement. Within a fraction of the first task, about 0.1 seconds, the algorithm has computed the feedforward parameters and the error is reduced with a factor 10, indicating the benefit of direct learning.

To conclude, a good and fast estimate of the feedforward parameters is obtained through on-line learning in a practical situation, furthermore the overall

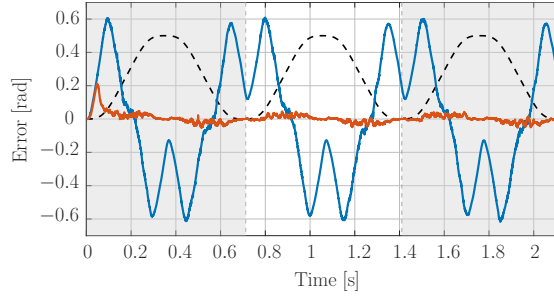


Figure 6.9: Positioning error obtained with feedback (—) compared to feedback and learning feedforward (—), and the scaled reference (---).

performance is improved with a factor 10.

6.7 Conclusions

In this chapter, a unified framework is provided for automatic feedforward controller tuning. The framework exploits both batch-wise tuning as well as on-line tuning, leading to new opportunities and insights for feedforward control. A linear least squares optimization problem with an analytic solution is developed to optimize the feedforward controller parameters. Furthermore, a detailed statistical analysis is provided in the case where measurement noise is present, indicating that biased estimates are obtained in earlier approaches (Butler, 2012). Furthermore, a simple and practical approach is presented to reduce the estimation bias in a practical situation. An example confirms the theoretical conclusions, i.e., indicating the possible hazard of using biased estimates for feedforward control resulting in a deterioration of the positioning performance. Moreover, the approach is validation through experiments on a benchmark motion system.

6.8 Proof

Proof of Theorem 6.10

The regressor matrix in the least squares solutions (6.21) can be rewritten as follows,

$$\Phi^\top = (Y_m \Psi^\top)^\top \quad (6.49)$$

with $Y_m^\top = [y_m(1) \quad y_m(2) \quad \dots \quad y_m(k)]$. Substitute (6.49) in (6.21) to obtain

$$\hat{\theta}(k) = ((Y_m \Psi^\top)^\top Y_m \Psi^\top)^{-1} (Y_m \Psi^\top)^\top \bar{U} \quad (6.50)$$

$$= (\Psi^\top Y_m^\top Y_m \Psi)^{-1} \Psi^\top Y_m \bar{U}. \quad (6.51)$$

Substitute, $Y_m = Y_0 + V$ and $\bar{U} = B_0 P_0 Y_0 = \Psi^\top Y_0 \theta_0$, such that (6.51) can be reformulated as a function of Y_0 and V

$$\hat{\theta} = [\Psi(Y_0^\top Y_0 + 2V^\top Y_0 + V^\top V)\Psi]^{-1} \Psi^\top (Y_0 + V) \Psi^\top Y_0 \theta_0. \quad (6.52)$$

Consequently, use the matrix of basis function coefficients Γ in (6.30) and define $Q(q^{-1}) = [1 \quad q^{-1} \quad \dots \quad q^{-n_\psi}]^\top$ to rewrite the basis functions as $\Psi_A = \Gamma_a Q(q)$. Substitute this result in (6.52) to obtain

$$\hat{\theta} = [\Gamma_a^\top Q(Y_0^\top Y_0 + 2V^\top Y_0 + V^\top V)Q^\top \Gamma_a]^{-1} \Gamma_a^\top Q(Y_0 + V)Q^\top \Gamma_a \theta_0, \quad (6.53)$$

which is equivalent to

$$\hat{\theta} = [\Gamma_a^\top (\hat{\mathbf{R}}_{yy} + 2\hat{\mathbf{R}}_{vy} + \hat{\mathbf{R}}_{vv})\Gamma_a]^{-1} \Gamma_a^\top (\hat{\mathbf{R}}_{yy} + \hat{\mathbf{R}}_{vy})\Gamma_a \theta_0, \quad (6.54)$$

with \hat{R}_{vy} and \hat{R}_{yy} given by

$$\hat{\mathbf{R}}_{vy}(k) = QV^\top Y_0 Q^\top, \quad \hat{\mathbf{R}}_{yy}(k) = QY_0^\top Y_0 Q^\top. \quad (6.55)$$

Now use that under mild conditions in (Söderström and Stoica, 1989, Appendix B) it follows that

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{t=1}^k \hat{\theta}(k) \rightarrow \mathbb{E} \hat{\theta}(k) \quad (6.56)$$

with probability one. Consequently,

$$\mathbf{R}_{yy} = \lim_{k \rightarrow \infty} \frac{1}{k} \hat{\mathbf{R}}_{yy}(k) \quad \mathbf{R}_{vy} = \lim_{k \rightarrow \infty} \frac{1}{k} \hat{\mathbf{R}}_{vy}(k), \quad (6.57)$$

which are the auto-correlation and cross-correlation matrices respectively,

$$\mathbf{R}_{ab}(k) = \begin{bmatrix} \mathbb{E} a(k)b(k) & \dots & \mathbb{E} a(k)b(k - n_\psi) \\ \vdots & \ddots & \vdots \\ \mathbb{E} a(k - n_\psi)b(k) & \dots & \mathbb{E} a(k - n_\psi)b(k - n_\psi) \end{bmatrix}, \quad (6.58)$$

and $\lim_{k \rightarrow \infty} \hat{\mathbb{R}}_{vv}(k) \rightarrow \sigma_n^2 I$. Finally, the expected value of (6.54) is given by (6.29) in Theorem 6.10 which completes the proof.

CHAPTER

7

On-line Instrumental Variable-Based Feedforward Tuning for Non-Resetting Motion Tasks¹

Abstract: Tracking of non-resetting point-to-point references with high task flexibility that cannot be separated into individual batches, e.g., with a varying step size, requires continuous updating of learning feedforward controllers rather than task-by-task updating. The aim of this chapter is to develop an adaptive feedforward controller for non-resetting point-to-point motion tasks by continuously updating the feedforward parameters based on data. A consistent and approximate optimal instrumental variable (IV) estimator is used in a closed-loop setting to update the feedforward parameters with a real-time bootstrapping procedure. A case study on a wafer-stage and experimental validation on a benchmark motion system show the immediate performance benefit.

¹The results in this chapter constitute Contribution 6 of this thesis. The chapter is based on Mooren, N. et al. (2022e). On-line Instrumental Variable-Based Feedforward Tuning for Non-Resetting Motion Tasks. (*Submitted for journal publication*).

7.1 Introduction

Learning feedforward control can yield tremendous performance improvement for high-precision motion systems that require highly flexible and non-resetting motion tasks as in Figure 7.1. Consider for example the wafer-stage for semiconductor manufacturing in Figure 7.2, which has a short-stroke stage (blue) for accurate and fast positioning of the semiconductor components in front of a gripper, and a long-stroke stage (yellow) to enable a large range. Sequentially picking up semiconductor components from the wafer constitutes such a varying and non-resetting point-to-point reference in Figure 7.1, where Δr_i varies depending on the relative position between the current and the next semiconductor component, and the dwell time Δt_i depends on other modules in the system to be ready for the next task, see, e.g., Boeren et al. (2016); Boeren et al. (2014); van der Veen et al. (2020). Moreover, the system dynamics are position-dependent due to the large range. Hence, a data-driven feedforward approach is required that can cope with continuous motion tasks with a varying step size and task length, i.e., the system does not resets to the same initial condition at the start of a next task which is required for batch-wise feedforward approaches (Bristow et al., 2006). As a result, batch-wise learning of feedforward parameters is not possible and does not exploit data during the current task.

Manual tuning of feedforward parameters and model-based feedforward can compensate for non-resetting and varying references, i.e., with an inverse model of the system any reference can be completely compensated. However, the performance is directly related to the inverse-model quality, and manual tuning of feedforward parameters (Oomen, 2019), e.g., mass, snap and jerk parameters, or non-linear phenomena such as hysteresis (Strijbosch et al., 2021), is often time-consuming and need to be performed on an inefficient machine-specific basis. Moreover, in the case of position-dependent behavior, a different feedforward tuning is required over the entire range. In view of these challenges, an automatic data-driven tuning approach is preferred to learn feedforward controllers.

Batch-wise learning approaches, such as iterative learning control (ILC), learn from previous tasks to improve tracking performance for systems with resetting references. Iterative learning control (ILC) learning a feedforward signal that exactly compensates the same reference, even in the case of a non-perfect model (Altin et al., 2017; Bristow et al., 2006; Veronesi and Visioli, 2014). If varying and resetting tasks appear, then ILC can be combined with basis function and learn the corresponding parameters (Bolder and Oomen, 2015; van de Wijdeven and Bosgra, 2010). However, as ILC requires the same initial condition at the start of each task r_i this batch-wise approach is not applicable (Gorinevsky et al., 1997) if the reference is non-resetting as in Figure 7.1.

On-line learning of feedforward parameters enables performance improvements within a task without imposing restrictions on the resetting behavior of the reference. In Butler (2012) recursive least squares (RLS) is successfully used

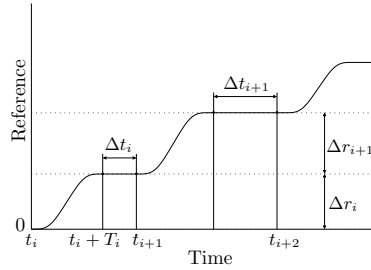


Figure 7.1: Example of a typical point-to-point non-resetting reference with varying step size $\Delta r_i \in \mathbb{R}$ and dwell time $\Delta t_i \in \mathbb{N}$.

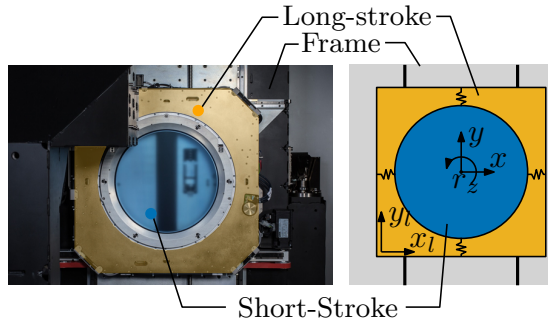


Figure 7.2: Motivating application: a long-stroke short-stroke wafer-stage for semiconductor manufacturing that performs non-resetting and varying point-to-point motion tasks.

for on-line estimation of the acceleration feedforward parameter on a wafer-stage resulting in an immediate performance improvement. However, in Chapter 6 it is shown that this approach inherently suffers from a closed-loop estimation problem resulting in biased estimates resulting in performance degradation. For accurate reference tracking, it is essential that unbiased parameter estimates in an on-line setting.

Instrumental variable (IV) estimators can obtain unbiased estimates in a closed-loop setting by appropriate design of so-called instrumental variables, see, e.g., Gilson et al. (2011); Gilson and Van Den Hof (2005); Janot et al. (2013); Ljung (1999); Söderström and Stoica (1989). In Boeren et al. (2018) an IV estimator is applied for optimal updating of feedforward parameters. This result shows that for rational systems with resetting point-to-point motion tasks, a combination of an adaptive input-shaper and feedforward-controller facilitates a convex optimization problem with an analytic solution. However, IV estimation is performed in a batch-wise setting it is not applicable to non-resetting varying references.

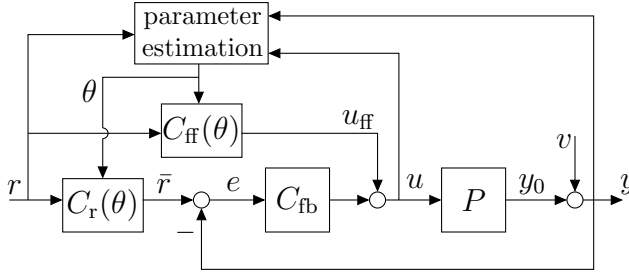


Figure 7.3: Control problem with adaptive input shaper C_r and feedforward controller C_{ff} with parameter $\theta(k)$ this is estimated in real-time.

Although recent progress has been made for on-line feedforward controller tuning for non-resetting tasks, an approach that yields an accurate estimate, i.e., unbiased with small variance, in a closed-loop setting is not yet available. The main contribution of this chapter is an optimal IV-based adaptive feedforward controller for non-resetting point-to-point references with an on-line bootstrapping approach (Freedman, 2009; Söderström and Stoica, 1989; Tjarnstrom and Ljung, 2002), yielding a convex optimization problem for efficient on-line implementation. The following contributions are identified.

- C1 IV estimation of feedforward parameters for arbitrary non-resetting point-to-point references,
- C2 recursive approximate optimal IV estimation through on-line bootstrapping for accurate parameter estimation,
- C3 a case study on the wafer-stage in Figure 7.2 with non-resetting varying point-to-point references, and
- C4 experimental validation on benchmark motion system.

The outline of this chapter is as follows. In Section 7.2, the control problem is formulated as an estimation problem that is formally presented. In Section 7.3, the recursive IV-based estimator for on-line feedforward parameter estimation is presented with an on-line bootstrapping approach. In Section 7.4, additional design aspects are outlined. In Section 7.5, a simulation case-study is performed with a wafer-stage setup, and in Section 7.6 the approach is validated on a benchmark motion system. Finally, in Section 7.7, the conclusions are presented.

7.2 Problem definition

7.2.1 Control setting and problem definition

The control setting is depicted in Figure 7.3, where P is a linear time-invariant (LTI) single-input single-output (SISO) system that represent the dynamics of the short-stroke wafer-stage in the x -direction, and C_{fb} is a stabilizing LTI feedback controller. The system P is rational given by

$$P = \frac{B_0(q^{-1}, \theta_0)}{A_0(q^{-1}, \theta_0)}, \quad (7.1)$$

where A_0 and B_0 are polynomials in the backwards time-shift operator q^{-1} , i.e., $u(k-1) = q^{-1}u(k)$ with $k \in \mathbb{Z}$ discrete time, and $\theta_0 \in \mathbb{R}^{n_\theta}$ are the true system parameters. The aim is to track a non-resetting point-to-point reference r depicted in Figure 7.1, by designing the adaptive feedforward controller $C_{ff}(\theta)$ and input shaper $C_r(\theta)$ that depend on $\theta(k)$. The signal $\theta(k) \in \mathbb{R}^{n_\theta}$ contains $n_\theta \in \mathbb{N}$ parameters that are updated in an on-line fashion by a parameter estimator on the basis of $r(k)$, the system input $u(k)$ and the noisy output $y(k) = y_0(k) + v(k)$, where $v(k) \sim \mathcal{N}(0, \sigma_v^2)$ is zero-mean normally distributed with variance σ_v^2 . In the remainder, the focus is on improving performance in the dwell time phase $\Delta t_i \in \mathbb{N}$, e.g., where the semi-conductor component in the earlier example is picked-up from the wafer.

Definition 7.1. *The control goal is to minimize the reference tracking error $\|e_r\|^2$ with $e_r = r - y$, during the dwell time Δt_i by designing the adaptive feedforward controller $C_{ff}(\theta)$ and input shaper $C_r(\theta)$.*

The tracking error e_r to be minimized is given by

$$e_r = (1 - SP(C_{ff}(\theta_k) + C_{fb}C_r(\theta_k)))r, \quad (7.2)$$

where $S = (1 + PC_{fb})^{-1}$ is the sensitivity function and $\theta_k = \theta(k)$. It is important to note that the error (7.2) is not necessarily equal to the feedback error $e = \bar{r} - y$, with $\bar{r} = C_r(\theta)r$, which is minimized by C_{fb} .

7.2.2 Feedforward controller parameterization

The input-shaper C_r and feedforward controller C_{ff} are polynomials and linear in the parameters θ , which facilitate a convex feedforward parameter estimation problem that is beneficial for the on-line implementation. Consider the following parameterization, see also Boeren et al. (2014).

Definition 7.2. *The input shaper $C_r(\theta)$ and feedforward controller $C_{ff}(\theta)$ are of the form*

$$C = \left\{ (C_r, C_{ff}) \left| \begin{array}{l} C_{ff} = A(q^{-1}, \theta_k) \\ C_r = B(q^{-1}, \theta_k) \end{array} \right. \right\}, \quad (7.3)$$

with $A(q^{-1}, \theta_k)$ and $B(q^{-1}, \theta_k)$ polynomials in q^{-1} given by,

$$A(q^{-1}, \theta_k) = \sum_{i=1}^{n_a} \psi^i(q^{-1})^i \theta_k^i = \Psi_A^\top \theta_A \quad (7.4)$$

$$B(q^{-1}, \theta_k) = 1 + \sum_{i=n_a+1}^{n_a+n_b} \psi^i(q^{-1})^i \theta_k^i = 1 + \Psi_B^\top \theta_B, \quad (7.5)$$

that are linear in θ_k . Where $\psi(q^{-1}) \in \mathbb{R}[q^{-1}]$ are the basis functions where $n_b, n_a \in \mathbb{N}$, also

$$\Psi_A = [\psi^1 \quad \dots \quad \psi^{n_a}]^\top \in \mathbb{R}^{n_a}[q^{-1}] \quad (7.6)$$

$$\Psi_B = [\psi^{n_a+1} \quad \dots \quad \psi^{n_\theta}]^\top \in \mathbb{R}^{n_b}[q^{-1}] \quad (7.7)$$

are the basis functions of C_{ff} and C_r respectively with corresponding parameters $\theta_A = [\theta^1 \quad \dots \quad \theta^{n_a}]^\top \in \mathbb{R}^{n_a}$, $\theta_B = [\theta^{n_b} \quad \dots \quad \theta^{n_a+n_b}]^\top \in \mathbb{R}^{n_b}$, and $\theta(k) = [\theta_A(k) \quad \theta_B(k)]^\top \in \mathbb{R}^{n_\theta}$ with $n_\theta = n_a + n_b$.

The following assumption is imposed on the input shaper.

Assumption 7.3. The input shaper C_r satisfies that

$$C_r(\theta, q^{-1})|_{q^{-1}=1} = 1, \quad (7.8)$$

such that it has unit steady-state gain.

Assumption 7.3 avoids scaling of the reference. Note that the reference $\bar{r}(k) = C_r r(k)$ reaches steady state n_k samples after $r(k)$ is constant which is the case in the dwell time Δt , with n_k the order of C_r . Consequently, $\bar{r}(k) = r(k)$ for $k \in \Delta T$ with $\Delta T = [t_i + T_i + n_k, t_{i+1}] \forall i \in \mathbb{N}$, see, e.g., Boeren et al. (2014). As a result, the tracking error (7.2) is identical to the feedback error

$$e = S(C_r(\theta) - PC_{\text{ff}}(\theta))r, \quad (7.9)$$

for all $k \in \Delta T$, minimization of (7.9) is obtained if, ideally,

$$C_r(\theta) = PC_{\text{ff}}(\theta) \quad (7.10)$$

for some θ such that $C_{\text{ff}}C_r^{-1} = P^{-1}$ contains a model of the inverse system.

7.2.3 Estimation for feedforward control

In this section, the control goal in Definition 7.1 is reformulated as an equivalent estimation problem under Assumption 7.3 and the feedforward controller parameterization \mathcal{C} . The ideal setting in (7.10) implies that $e = 0$ if $C_r(\theta) = A_0$ and $C_{\text{ff}}(\theta) = B_0$ for arbitrary references. Hence, it is necessary that the true system P must be in the model set that is defined by the basis functions Ψ_A and Ψ_B .

Assumption 7.4. *The true system is in the model set, i.e., the basis functions Ψ_A and Ψ_B satisfy that $\Psi_A\theta_A = A_0$, and $\Psi_B\theta_B = B_0$ for some θ_A and θ_B .*

The following definition outlines properties of the parameter estimator such that the tracking error is minimized during the dwell time Δt .

Definition 7.5. *In the setting in Figure 7.3 under Assumptions 7.3 and 7.4, and with the parameterization \mathcal{C} , the tracking error $e_r(k)$ for $k \in \Delta$ is minimized by obtaining an accurate estimate $\theta(k)$, i.e., such that the estimation error $\theta(k) - \theta_0$ asymptotically has normal distribution*

$$\sqrt{k}(\theta - \theta_0) \rightarrow \mathcal{N}(0, P) \quad \text{for } k \rightarrow \infty, \quad (7.11)$$

with the following properties:

R1 $(\theta(k) - \theta_0) \rightarrow 0$ as $k \rightarrow \infty$; and

R2 P is minimal.

Definition 7.5 yields that by design of an appropriate estimator, i.e., that recovers the true plant parameters in the presence of noise v and in a closed-loop setting, the tracking error is minimized during the dwell time Δt_i .

7.2.4 Control problem

Estimation of the feedforward controller parameters is in practice often done using a least-squares estimator, see, e.g., Butler (2012); Fujimoto and Yao (2005). However, it is shown that biased estimates are often obtained if the estimation is performed in a closed-loop setting with measurement noise, see, e.g., Chapter 6. The aim of this chapter is to use a suitable estimator, in view of Definition 7.5, to estimate the feedforward parameters in the closed-loop setting in Figure 7.3 with $v \neq 0$.

7.3 On-line IV-based feedforward controller tuning

In this section, an approximate optimal IV estimator with on-line bootstrapping is presented that yields a recursive update law for the feedforward parameters $\theta(k)$ in the setting in Figure 7.3, which consequently minimizes the tracking error during the dwell time, see Section 7.3.1 and 7.3.3. This includes conditions on the optimal instrumental variables to obtain the theoretical lower bound, which appears to depend on the true system parameters. Therefore, an on-line bootstrapping procedure is presented that approximates the optimal instrumental variables in practice in Section 7.3.2. Finally, an implementation procedure is provided in Section 7.3.4.

7.3.1 Optimal IV estimation for feedforward

A suitable estimator must be developed for the adaptive feedforward parameters $\theta(k)$ as shown in Definition 7.5 to minimize the tracking error. To do so, an IV estimator is used that enables recursive tuning of $\theta(k)$ on the basis $u(k)$, $y(k)$ and $r(k)$. Moreover, it is shown that the presented IV estimation problem yields a recursive update law for the feedforward parameters $\theta(k)$ that is suitable for on-line implementation.

Theorem 7.6. *Consider the setup of Figure 7.3 with feedforward parameterization \mathcal{C} in (7.3), and under Assumptions 7.3 and 7.4. The feedforward parameters that satisfy R1 and R2, and consequently minimize the $e_r(k)$ for $k \in \Delta T$, are obtained by*

$$\theta(k) = \arg \min_{\theta} V_{\text{IV}}(k, \theta), \quad (7.12)$$

with

$$V_{\text{IV}}(k, \theta) = \left\| \left[\frac{1}{k} \sum_{i=1}^k z(i) (F(q^{-1})\epsilon(i, \theta)) \right] \right\|^2, \quad (7.13)$$

$$\epsilon(k, \theta) = u(k) - \phi^\top(k)\theta(k), \quad (7.14)$$

where $\epsilon(k, \theta) \in \mathbb{R}$ is linear in θ , $z(k) \in \mathbb{R}^{n_z}$, $F(q^{-1})$ is a stable filter, and

$$\phi(k) = \begin{bmatrix} \Psi_A(q^{-1})y(k) \\ -\Psi_B(q^{-1})u(k) \end{bmatrix} \in \mathbb{R}^{n_\theta}. \quad (7.15)$$

The estimate (7.12) is consistent, i.e., $(\theta_0 - \theta(k)) \rightarrow 0$ for $k \rightarrow \infty$, if

$$C1 \quad \mathbb{E}[z(k)F(q^{-1})A_0(\theta_0)v(k)] = 0,$$

$$C2 \quad \mathbb{E}[z(k)F(q^{-1})\phi(k)] \text{ is non-singular, and}$$

the estimator error has optimal asymptotic distribution $(\theta(k) - \theta_0) \rightarrow \mathcal{N}(0, P_{IV})$ with minimal variance if

$$z(k) = A_0^{-1}(q^{-1}, \theta_0)\phi_r(k) := z^{\text{opt}}(k) \quad (7.16)$$

$$F(q^{-1}) = A_0^{-1}(q^{-1}, \theta_0) := F^{\text{opt}}(q^{-1}) \quad (7.17)$$

where

$$\phi_r(k) = \begin{bmatrix} \Psi_A(q^{-1})y_r(k) \\ -\Psi_B(q^{-1})u_r(k) \end{bmatrix}, \quad (7.18)$$

the noise-free part of $\phi(k)$ and $u_r(k)$ and $y_r(k)$ are the noise-free input and output respectively.

The proof of Theorem 7.6 is included in Section 7.8.

In this chapter, a recursive IV-based procedure is presented that exploits on-line bootstrapping to closely approximate optimal IV. Under optimality conditions, the result in Theorem 7.6 yields that the plant parameters are consistently recovered with minimal variance. Moreover, in the non-optimal case the parameter estimates remain unbiased but the variance can be larger than the theoretical lower bound. As a result, the reference-induced tracking error (7.9) is minimized. The presented approach updates each sample in contrast to the existing approaches, e.g., (Boeren et al., 2017) which exploits resetting reference tasks. In the case of repeating reference tasks the batch-wise approach in (Boeren et al., 2017) can be used, where bootstrapping is avoided as shown later.

Next, consider the following persistence of excitation condition.

Definition 7.7. *The input u is persistently exciting of order n if for all k there exists an integer m such that*

$$\gamma_1 I > \sum_{i=k}^{k+m} \xi(i) \xi(i)^\top > \gamma_2 I, \quad (7.19)$$

$$(7.20)$$

where $\gamma_1, \gamma_2 > 0$, and $\xi(k) = [u(k-1) \ \dots \ u(k-n)]$ see (Åström and Wittenmark, 2013; Ioannou and Sun, 1996, Chapter 2.4) (Ioannou and Sun, 1996).

Assumption 7.8. *The reference is persistently exciting according to Definition 7.7.*

Theorem 7.6 reveals that with the IV-estimator (7.12) the estimate $\theta(k) \rightarrow \theta_0$ for $k \rightarrow \infty$ if the instrumental variables $z(t)$ are uncorrelated with the noise $v(t)$ (C1) and correlated with the regressor (C2), satisfying R1. Moreover, with the optimal instrumental variables z^{opt} in (7.16) and the optimal pre-filter F^{opt} in (7.17), the theoretical lower-bound of the covariance matrix P_{IV} is obtained, satisfying R2. However, since both z^{opt} and F^{opt} depend on the unknown system parameters θ_0 , approximations of the optimal instrumental variables and pre-filter are provided next with on-line bootstrapping. Finally, under Assumption 7.4 and 7.8 and with the estimator in Theorem 7.6 the estimate is said to be identifiable (Söderström and Stoica, 1989).

Note that C1 directly imposes the limitation that $z(t)$ cannot be constructed from measured signals that contain noise. Therefore, an external signal is required; in this work the reference is used to construct the instrumental variables, i.e., $z(t) = f(r(t), \dot{r}(t), \ddot{r}(t), \dots)$, as in, e.g., (Boeren et al., 2018).

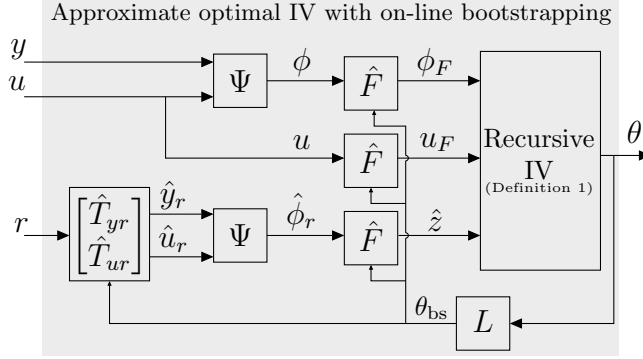


Figure 7.4: Schematic representation for approximate optimal instrumental variable (IV) estimation for adaptive feedforward with on-line bootstrapping, where $\Psi = [\Psi_A \quad -\Psi_B]^\top$.

7.3.2 Towards optimal feedforward using on-line bootstrapping

In this section, an approximation of the optimal instruments z^{opt} and pre-filter F^{opt} is provided using on-line bootstrapping to approximate the optimal variance, see, e.g., Freedman (2009); Söderström and Stoica (1989); Tjarnstrom and Ljung (2002) for more details on bootstrapping. This is done by replacing θ_0 by an estimate denoted $\theta_{\text{bs}}(k)$ at each sample. The aim is to approximate the theoretical lower-bound of the estimator variance. The approach is presented in Figure 7.4, where L is an optional pre-filter, and \hat{z} and \hat{F} are approximation of their optimal versions. In this section, set $L(q^{-1}) = q^{-1}$, such that $\theta_{\text{bs}}(k) = \theta(k-1)$ is the previous estimate.

First, the approximation of the optimal pre-filter in (7.17) is straight forward by replacing θ_0 with $\theta_{\text{bs}}(k)$ yielding

$$\hat{F}(q^{-1}, \theta_{\text{bs}}, k) = (\Psi_A(q^{-1})\theta_{\text{bs}}^A(k))^{-1}. \quad (7.21)$$

Second, the approximation of z^{opt} in (7.16) is more involved, because it requires the noise-free regressor $\phi_r(k)$ in (7.18) that required y_r and u_r . To approximate $\phi_r(k)$, note that u_r and output y_r can be written as function of $r(k)$, i.e.,

$$y_r(k) = T_{yr}(\theta_0)r(k) \quad (7.22)$$

$$u_r(k) = T_{ur}(\theta_0)r(k), \quad (7.23)$$

with

$$T_{yr}(\theta_0) = S(\theta_0)C(\theta) \quad (7.24)$$

$$T_{ur}(\theta_0) = P(\theta_0)S(\theta_0)C(\theta) \quad (7.25)$$

$$C(\theta) = C_{\text{ff}}(\theta) + C_{\text{fb}}C_r(\theta), \quad (7.26)$$

here, $T_{ur}(q^{-1}, \theta_0)$ and $T_{yr}(q^{-1}, \theta_0)$ are the mappings from r to the noise-free input and output respectively. Since T_{ur} and T_{yr} also depend on θ_0 , a similar bootstrapping approach is utilized, i.e.,

$$\hat{u}_r(k) = \hat{T}_{ur}(\theta_{\text{bs}}(k))r(k) \quad (7.27)$$

$$\hat{y}_r(k) = \hat{T}_{yr}(\theta_{\text{bs}}(k))r(k) \quad (7.28)$$

are used as the approximations of the noise-free input and output with

$$\hat{T}_{yr}(q^{-1}, \theta_{\text{bs}}) = S(\theta_{\text{bs}})C(\theta) \quad (7.29a)$$

$$\hat{T}_{ur}(q^{-1}, \theta_{\text{bs}}) = P(\theta_{\text{bs}})S(\theta_{\text{bs}})C(\theta). \quad (7.29b)$$

Note that the filters (7.29) can become unstable for some θ_{bs} , in this case a stable approximation is presented in Section 7.4. Consequently, the noise-free regressor is given by

$$\hat{z}(k) = [\Psi_A \hat{T}_{yr}(\theta_{\text{bs}}) \quad \Psi_B \hat{T}_{ur}(\theta_{\text{bs}})] r(k). \quad (7.30)$$

In summary, the optimal instruments and the pre-filter are approximated with (7.30) and (7.21) respectively, where the parameters $\theta_{\text{bs}}(k)$ are (filtered versions) of $\theta(k-1)$ which is referred to as on-line bootstrapping. In the following section, the resulting update law for the feedforward parameters θ is outlined.

Remark 7.9. *If the reference is resetting, i.e., can be split into individual batches, then the parameterized filters (7.29) can be replaced by a stable approximation of C^{-1} , see, (Boeren et al., 2017). However, that approach is limited to SISO systems which is not the case for the presented approach with bootstrapping.*

7.3.3 Recursive IV for on-line feedforward control

A recursive solution to the parameter estimation problem in Theorem 7.6 that is suitable for the on-line tuning of $C_r(\theta)$ and $C_{\text{ff}}(\theta)$ is presented in the following lemma.

Definition 7.10. *The parameters $\theta(k)$ in $C_{\text{ff}}(\theta)$ and $C_r(\theta)$ at sample k are given by the recursive solution to (7.12) on the basis of $u(k)$, $\phi(k)$, $z(k)$, and the (initial) parameter estimate $\theta(k-1)$, with the update law*

$$\theta(k) = \theta(k-1) + K(k) (\nu(k) - \Phi^\top(k)\theta(k-1)), \quad (7.31)$$

with

$$\begin{aligned}
K(k) &= P(k-1)\Phi(k)[\Lambda(k) + \Phi^\top(k)P(k-1)\Phi(k)]^{-1} \\
\Phi(k) &= \begin{bmatrix} R^\top(k-1)z(k) & \phi_F(k) \end{bmatrix} \in \mathbb{R}^{n_\theta \times 2} \\
\Lambda(k) &= \begin{bmatrix} -z^\top(k)z(k) & 1 \\ 1 & 0 \end{bmatrix} \in \mathbb{R}^{2 \times 2} \\
\nu(k) &= \begin{bmatrix} z^\top(k)\eta(k-1) \\ u_F(k) \end{bmatrix} \in \mathbb{R}^2 \\
R(k) &= R(k-1) + z(k)\phi_F^\top(k) \\
\eta(k) &= \eta(k-1) + z(k)u_F(k) \\
P(k) &= P(k-1) - K(k)\Phi^\top(k)P(k-1)
\end{aligned}$$

where $K(k) \in \mathbb{R}^{n_\theta \times 2}$, $P(k) \in \mathbb{R}^{n_\theta \times n_\theta}$, $\eta(k) \in \mathbb{R}^{n_z}$, and $R(k) \in \mathbb{R}^{n_z \times n_\theta}$ are recursively computed, and $u_F(k) = F(q^{-1})u(k)$ and $\phi_F(k) = F(q^{-1})\phi(k)$, constituting the recursive solution to the IV estimate in Theorem 7.6.

The update law for the feedforward parameters (7.31) is used to update the feedforward controllers at each sample. Hence, the control input u_{ff} and the reference \bar{r} at sample k are given by

$$u_{\text{ff}}(k) = \Psi_A(q^{-1})\theta_A(k)r(k) \quad (7.32)$$

$$\bar{r}(k) = \Psi_B(q^{-1})\theta_B(k)r(k) \quad (7.33)$$

where $\theta(k)$ is computed using Definition (7.10).

To compute $\theta(k)$, only the previous estimate $\theta(k-1)$, the current value of the input $u(k)$, the output $y(k)$ and the instruments $z(k)$ are required, i.e., the recursive algorithm does not require memory of all the past samples to compute the parameter update for the current time step. Moreover, the matrix inversion to compute $K(k)$ is of size 2×2 for all k which is computationally inexpensive, this allows to implement the IV-based feedforward controller in an on-line setting.

Remark 7.11. If the reference can be separated in individual tasks that start from the same initial conditions, then the batch-wise solution to the identification problem in Theorem 7.6 is given by

$$\theta_i = (R_N^\top R_N)^{-1} R_N^\top U_N,$$

for task i with N the number of data points and $R_N = \sum_{i=1}^N z(i)F(q^{-1})\phi^\top(i)$ and $U_N = \sum_{i=1}^N z(i)F(q^{-1})u(i)$, see, e.g., Boeren et al. (2015).

7.3.4 Procedure

The following procedure outlines the implementation for approximate optimal IV-based feedforward control.

Procedure 7.12 (On-line approximate bootstrapping IV).

Step 1: Initialization

- (a) Set $\theta(0) = \theta_{\text{init}}$, $P(0) = \alpha I_{n_\theta}$ with α a large number, $\eta(0) = \underline{0}$ and $R(0) = \underline{0}$ with $\underline{0}$ a matrix of appropriate size, and $k = 1$. Moreover, define $\Psi_A(q^{-1})$ and $\Psi_B(q^{-1})$.

Step 2: On-line IV-based feedforward

- (a) Compute $\hat{F}^{\text{opt}}(\theta_{\text{bs}})$, $\hat{T}_{yr}(\theta_{\text{bs}})$ and $\hat{T}_{ur}(\theta_{\text{bs}})$ in (7.29) with $\theta_{\text{bs}}(k) = L(q^{-1})\theta(k-1)$.
- (b) Obtain $u(k)$, $y(k)$ and $r(k)$, compute $\phi(k)$ in (7.15), and $z(k)$.
- (c) Compute $\theta(k)$, and consequently $\bar{r}(k)$ and $u_{\text{ff}}(k)$ using Definition 7.10.
- (d) set $k \rightarrow k+1$ and repeat step 2a) - 2c).
-

This procedure completes the basic implementation of on-line bootstrapping IV for adaptive feedforward. In Section 7.4 additional implementation aspects are provided, e.g., how to ensure stability of the filter $T_{ur}(\theta_{\text{bs}})$ and $T_{yr}(\theta_{\text{bs}})$.

Remark 7.13. *The bootstrapping step can also be performed in a batch-wise approach instead of at each sample. By starting to bootstrap with $k > k_s$ and $k_s > 1$, the estimate $\theta(k)$ has already converged to avoid divergence at the start of the algorithm.*

7.4 Design aspects

In this section, implementation aspects are provided, including a stable approximation of the adaptive filters T_{ur} , T_{yr} , and \hat{F}^{opt} , the inclusion of an exponential forgetting factor in the Definition 7.10, and how to choose the optimal filter L in Figure 7.4.

7.4.1 Stable approximations for on-line bootstrapping

The filters $\hat{T}_{yr}(\theta_{\text{bs}})$, $T_{ur}(\theta_{\text{bs}})$ and $\hat{F}^{\text{opt}}(\theta_{\text{bs}})$ can be unstable for some $\theta_{\text{bs}}(k)$ such that \hat{u}_r and \hat{y}_r can become unbounded which is undesirable. The filters defined in (7.21) and (7.29) are expected to be stable by design. Therefore, the following procedure presents a stable approximation approach where the possibly unstable poles in $\hat{T}_{yr}(\theta_{\text{bs}})$, $T_{ur}(\theta_{\text{bs}})$ and $\hat{F}^{\text{opt}}(\theta_{\text{bs}})$ are approximated with stable poles.

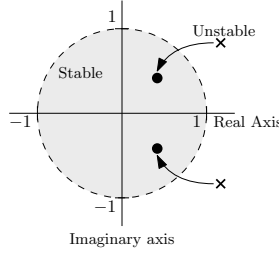


Figure 7.5: Graphical illustration of the stable approximation in Procedure 7.14 where unstable poles (x) are approximated with stable poles (●) with zero magnitude error.

Procedure 7.14 (Zero magnitude stable approximation).

1. Separate the denominator of G into a stable part A_s and an unstable part A_u , i.e.,

$$G(q^{-1}) = \frac{B(q^{-1})}{A_u(q^{-1})A_s(q^{-1})}, \quad (7.34)$$

where $A_u(q^{-1}) = a_0 + a_1q^{-1} + \dots + a_nq^{-n}$.

2. The stable approximation of G such that $|G| = |\hat{G}_s|$ is given by

$$\hat{G}^s(q^{-1}) = \frac{B(q^{-1})}{A_u^f(q^{-1})A_s(q^{-1})}, \quad (7.35)$$

where $A_u^f(q^{-1}) = a_n + a_{n-1}q^{-1} + \dots + a_0q^{-n}$.

Procedure 7.14 creates a stable approximation that has zero magnitude error, similar to zero-magnitude-error-tracking-control (ZMETC) to obtain a stable approximation in Butterworth et al. (2008), see Figure 7.5. Alternative approaches are conceptually possible, e.g., ZPETC (Tomizuka, 1987) to obtain a zero-phase error estimate if desired, by replacing unstable poles by non-minimum phase zeros. The motivation for Procedure 7.14 is that it replaces unstable poles by stable poles which is expected since the filters $\hat{F}(\theta_{bs})$, $\hat{T}_{ur}(\theta_{bs})$, and $\hat{T}_{yr}(\theta_{bs})$ are expected to be stable as well.

Remark 7.15. If the P contains a rigid-body mode, i.e., poles at $1 + 0i$, then $F^{\text{opt}}(\theta_0)$ is unstable. To avoid this for $\hat{F}(\theta_{bs})$ the basis functions can be altered such that the poles are shifted slightly into the unit disk.

7.4.2 Filtering of $\theta(k)$ for bootstrapping

The parameter estimate θ obtained with Procedure 7.12 can be erratic when few data points are available, potentially leading to divergence of the bootstrapping approach. Directly using these erratic parameters for bootstrapping and feedforward is undesirable. Therefore, the filter L in Figure 7.4 is introduced to filter the output of the recursive IV algorithm before bootstrapping, e.g., with a low-pass filter characteristic. In this chapter, a moving average filter is used such that

$$\theta_{bs}(k) = \underbrace{\frac{(1-\gamma)q^{-1}}{1-\gamma q^{-1}}}_{L(q^{-1})} \theta(k) \quad (7.36)$$

with $\gamma = e^{-2\pi f_c T_s}$ where $f_c \geq 0$ is the cut-off frequency and T_s the sample time. Note that f_c is a tuning parameter, decreasing f_c yields that θ_{bs} is smoother, and for $\lim_{f_c \rightarrow \infty} L(q^{-1}, f_c) = q^{-1}$, i.e., no filtering. Note that with $f_c > 0$ the filter L introduces a modeling error in \hat{F}^{opt} , \hat{T}_{ur} and \hat{T}_{yr} . Hence, there is a trade-off between performance and robustness in the tuning of f_c that is preferably large.

7.4.3 Incorporating system knowledge

Modern control systems are implemented in discrete-time, i.e., a part of the plant that is introduced through zero-order-hold discretization and system delays are often known in advance. To use this knowledge instead of estimating it, the system can be parameterized as

$$P(q^{-1}, \theta_0) = \tilde{P}(q^{-1}) \frac{B_0(q^{-1}, \theta_0)}{A_0(q^{-1}, \theta_0)}, \quad (7.37)$$

where $\tilde{P}(q^{-1})$ contains the known part. Consequently, by using $\tilde{u}(k) = \tilde{P}(q^{-1})u(k)$ instead of $u(k)$ in Definition 7.10, the adaptation law is adapted accordingly. Moreover, the implementation of the adaptive feedforward becomes $\tilde{C}_r(q^{-1}) = C_r(q^{-1})\tilde{P}(q^{-1})$ accordingly.

Remark 7.16. *Alternatively, the inverse of \tilde{P} can be implemented in the feedforward controller, i.e., $\tilde{C}_{ff} = \tilde{P}^{-1}C_r$, if \tilde{P}^{-1} is stable. Moreover, a combination where the numerator and denominator of \tilde{P} are implemented in C_r and C_{ff} respectively is also possible.*

7.5 Wafer-stage case study

In this section, a case study is performed that mimics the model and the type of references that are encountered in the wafer-stage motivating example in

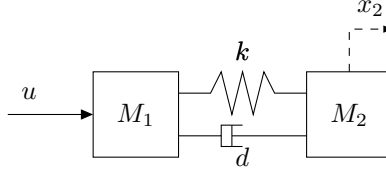


Figure 7.6: Case study: simplified model of the short-stroke wafer stage dynamics in the x -direction.

Figure 7.2. A simplified model of the short-stroke stage dynamics in the x -direction is depicted in Figure 7.6, i.e., it is modeled as two masses with a flexible element in-between.

7.5.1 Wafer-stage model

The model is given by

$$\begin{aligned} P(q^{-1}) &= \tilde{P}(q^{-1}) \frac{\Psi_B(q^{-1})\theta_0^B}{\Psi_A(q^{-1})\theta_0^A} \\ &= \tilde{P}(q^{-1}) \frac{9.26 \cdot 10^{-6}(1 + q^{-1})}{(1 - q^{-1})^2(1 - 1.967q^{-1} + 0.9977q^{-2})}, \end{aligned}$$

where \tilde{P} contains a discrete-time approximation of 2.2 samples delay, and the basis functions are

$$\Psi_B(q^{-1}) = \left(\frac{1-q^{-1}}{T_s} \right) \quad (7.38)$$

$$\Psi_A(q^{-1}) = \left[\left(\frac{1-q^{-1}}{T_s} \right)^2 \quad \left(\frac{1-q^{-1}}{T_s} \right)^3 \quad \left(\frac{1-q^{-1}}{T_s} \right)^4 \right]^\top. \quad (7.39)$$

with true system parameters $\theta_0^A = [39.4 \quad -0.018 \quad 0.00031] \cdot 10^{-5}$ and $\theta_0^B = -24.4 \cdot 10^{-5}$. A Bode plot of the system is shown in Figure 7.7 in (—). A stabilizing feedback controller is given by

$$C_{fb}(q^{-1}) = \frac{0.69 + 0.006q^{-1} - 0.68q^{-2}}{1 - 1.672q^{-1} + 0.696q^{-2}} \quad (7.40)$$

resulting in a bandwidth of approximately 10 Hz. Moreover, the system output is subject to additive normally distributed zero-mean noise with variance $\sigma_v = 10^{-4}$.

The aim is to track a point-to-point reference with a random step size Δr_i and random dwell times Δt_i , similar to the actual references encountered in the wafer stage setup. Several point-to-point reference tasks are depicted in Figure 7.8 which are shifted to the origin for comparison.

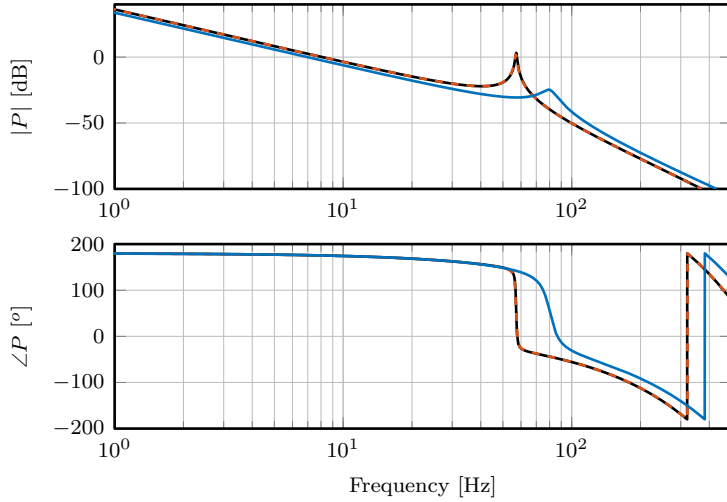


Figure 7.7: Case study: bode plot of the true system (—) and the model that is obtained with recursive approximate optimal IV identification with on-line bootstrapping (--) with the initial estimate (—).

7.5.2 Tuning

In view of Assumption 7.4, the basis functions for feedforward are selected as in (7.39). The initial parameter estimates are chosen as

$$\begin{aligned}\theta^A(0) &= [0.75 \quad 1.25 \quad 1.5] \theta_0^A, \\ \theta^B(0) &= 0,\end{aligned}$$

such that there is a significant initial error. Moreover,

$$P(0) = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix} \cdot 10^4,$$

and the cut-off frequency in (7.36) $f_c = 5$ Hz to filter out the high-frequency variations in θ before on-line bootstrapping.

7.5.3 Estimation performance through Monte Carlo simulation

To study the convergence and the variance of the recursive IV estimator with on-line bootstrapping, a set of 1000 Monte-Carlo simulations is performed with

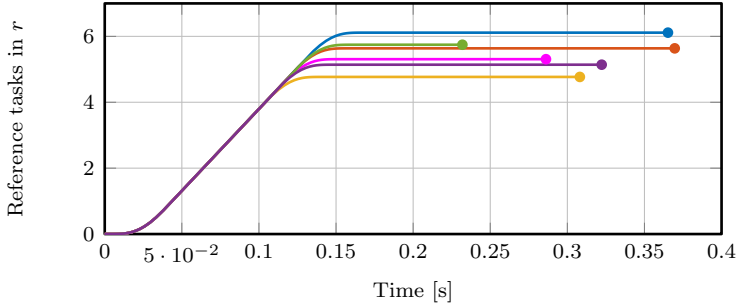


Figure 7.8: Case study: individual reference tasks taken from the point-to-point reference in the simulation case study. Note that each tasks has a different random step size and dwell time, the end of a task is indicated with the markers (●).

random noise realizations where each simulation consists of 60 point-to-point tasks as in Figure 7.1. Also, the optimal recursive IV estimator with true system parameters is used in the simulations to compare the theoretical lower bound for the variance with the obtained approximate optimal IV variance. Moreover, a simulation is performed without bootstrapping such that C1 and C2 in Theorem 7.6 are satisfied and the estimator is asymptotically unbiased, but the variance is not optimized. To do so, the converged estimation error $\theta - \theta_0$ of all simulations is shown in the histogram plot in Figure 7.9 for the first parameter with Procedure 7.12 (left) and with optimal IV (right).

From the results, it can be concluded that indeed the estimation error is normally distributed as shown in Theorem 7.6. Moreover, the variance of the on-line bootstrapping IV estimator (■) closely resembles the optimal IV estimator variance (■), in contrast to the IV estimator without bootstrapping (■) where the variance is much larger. The mean and variance of the optimal estimator and the approximate bootstrapping IV estimator are summarized in Table 7.1, confirming these observations.

Finally, the obtained plant estimate $C_r C_{ff}^{-1} \approx P$ in line with (7.10) is shown in Figure 7.7 together with the true plant P_0 and the initial estimate. The result shows that indeed a good estimate of the true system is obtained.

7.5.4 Feedforward performance results

The results of directly using the estimate $\theta(k)$ obtained with approximate optimal on-line bootstrapping IV for feedforward control as in Figure 7.3 is shown in Figure 7.10, where the tracking error $e = r - y$ is shown in addition to the feedback error $e_r = \bar{r} - y$. Moreover, the tracking error e_r is also shown without feedforward control for comparison.

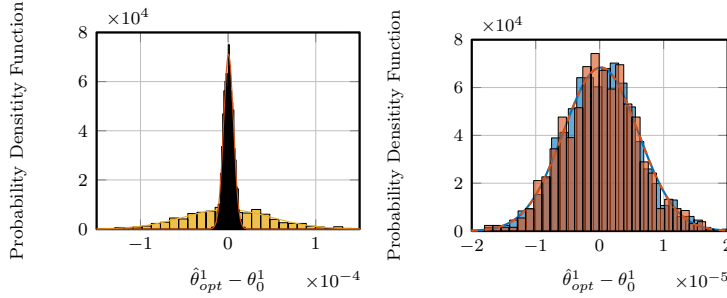


Figure 7.9: Case study: histogram plot of the estimation error for θ_2 over 1000 Monte Carlo simulations for optimal IV (■) and with a fixed set of instrumental variables (■), i.e., without bootstrapping (left). By employing the presented bootstrapping method in Procedure 7.12 the estimator variance (■) (right) is significantly improved and closely approximates the optimal variance (■).

Table 7.1: Mean and standard deviation of the converged error with 1000 Monte-Carlo simulations for optimal IV, approximate optimal IV with bootstrapping and IV without bootstrapping.

	Opt. IV	IV + bootstrapping	IV
Mean	$1.61 \cdot 10^{-7}$	$1.59 \cdot 10^{-7}$	$-1.17 \cdot 10^{-6}$
Standard deviation	$5.28 \cdot 10^{-6}$	$5.53 \cdot 10^{-6}$	$5.06 \cdot 10^{-5}$

The results show that with feedforward a significant performance improvement is obtained within the first task. Moreover, the focus is on the part where the velocity of the reference is zero, indicated by the gray areas, for the pick-up of a semiconductor component. In this phase, both error e_r and e are identical because of the underlying Assumption 7.3, indicating that good performance is obtained.

7.6 Experiments on a benchmark motion system

A benchmark motion system is used for experimental validation in a practical situation, i.e., where the system is not fully captured by the basis functions. The introduced approximate optimal IV approach with on-line bootstrapping is used for feedforward, moreover, a comparison is made with a recursive least squares (RLS) approach using the same basis functions.

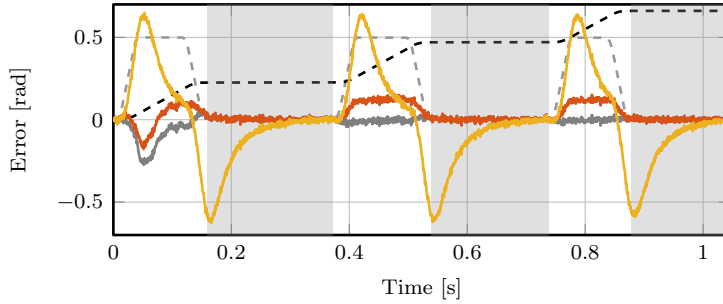


Figure 7.10: Case study single simulation: convergence of the tracking error e_r in (7.2) (—) with on-line IV-based feedforward in Procedure 7.12 in comparison to the tracking error e_r without feedforward (—). A scaled version of the reference (--) and its velocity (--) are also shown, the areas (■) indicate ΔT where performance is required, i.e., where e (—) in (7.9) is equal to e_r .

7.6.1 Experimental setup and tuning

The experimental setup is shown in Figure 7.11 which consists of two masses that are connected by a flexible axle as schematically depicted in Figure 7.6. The position of the collocated mass is used for feedback which is measured by an encoder with a resolution of $\frac{2\pi}{2000}$ rad, which induces additional quantization noise. A frequency response function of the setup is measured and depicted in Figure 7.12 in (—), which contains similar dynamics as the model used in Section 7.5. Therefore, the same set of basis functions is used. The setup also contains 2.2 samples of delay and a sampling zero that is included in \tilde{P} as in (7.37), i.e., this is not estimated. All other settings and initial parameters are identical so the simulation case study and provide in Section 7.5.2.

7.6.2 Experimental results

The resulting system estimates $C_r C_{ff}^{-1}$ and errors obtained in the experiments are shown in Figure 7.12 and Figure 7.13 respectively, from which the following conclusions can be made.

- The obtained plant estimate for feedforward, with the converged parameters, corresponds well with the measured FRF, indicating that a good model is obtained. In contrast to the recursive LS estimator, which yields a very poor estimate of the system due to an estimation bias, see Chapter 6.
- The tracking performance with IV-based on-line feedforward is significantly improved compared with only feedback control, i.e., within 0.2 seconds the error converges to a small value and remains small.

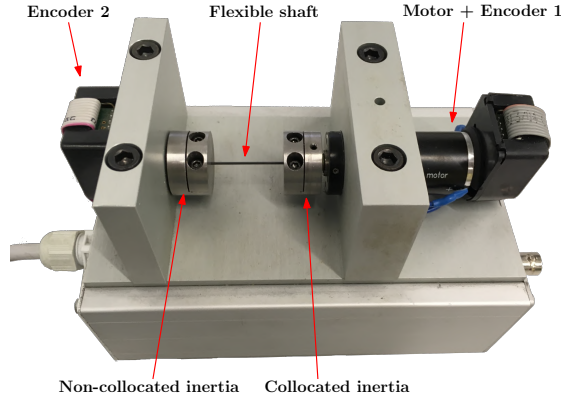


Figure 7.11: Benchmark motion system used for experimental validation.

- The tracking performance with Recursive LS-based on-line feedforward is even worse than without feedforward due to the bad system estimate.

These results confirm that in the presence of noise the recursive IV-based estimator is able to obtain a good system estimate and consequently good performance in the dwell time as required for the actual wafer-stage application.

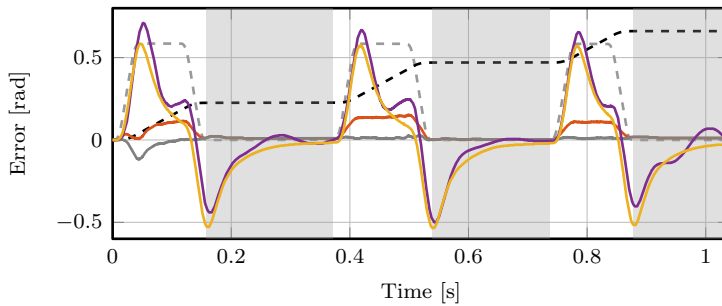


Figure 7.13: Experimental validation: tracking error e_r (—) in (7.2) obtained with Procedure 7.12 yielding good performance in comparison with the case without feedforward (—). Also a recursive least-squares based feedforward (—) is used that performs poor due to estimation bias. A scaled version of the reference (—) and its velocity (—) are also shown, and the areas (■) indicate where performance is required and where e (7.9) is equal to e_r .

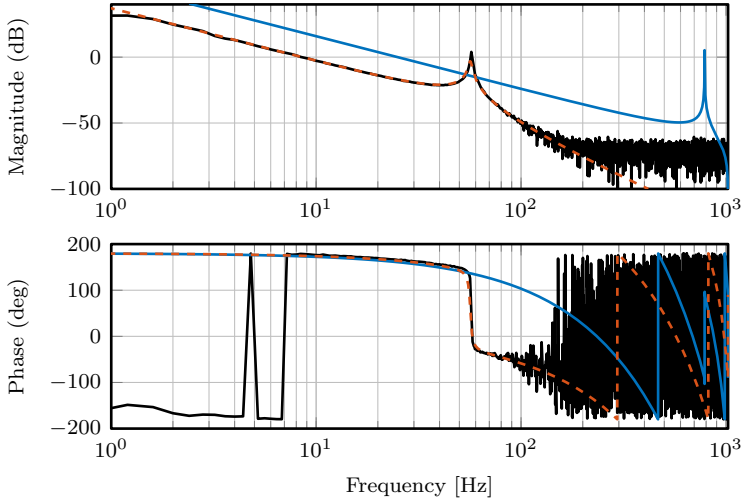


Figure 7.12: Experimental validation: bode plot of the measured frequency response function (FRF) of the experimental setup (—), the plant model $C_r C_{ff}^{-1}$ obtained with approximate optimal on-line IV feedforward (—) and with recursive least squares identification (—).

7.7 Conclusions

On-line learning of adaptive feedforward parameters for the tracking of varying point-to-point references is enabled through an approximate optimal instrumental variable (IV) estimator with an on-line bootstrapping approach. Traditional batch-wise learning approaches, such as ILC, are not applicable due to the non-resetting behavior of such references. In this chapter, feedforward parameters are estimated in an on-line setting on the basis of the reference and the system input and output, by using an IV estimator an unbiased estimate is approximate optimal variance is obtained in a closed-loop setting with measurement noise. A simulation case study of a wafer-stage setup and experiments on a benchmark motion system shows the immediate performance improvement that is obtained within a task. Ongoing work focuses on the implementation for systems that have slowly varying parameters by incorporating a forgetting factor in the on-line IV algorithm, such that the feedforward parameters are updated accordingly.

7.8 Proofs

Proof of Theorem 7.6

The proof of Theorem 7.6 consists of three parts, i) it is shown that the minimizer of (7.12) has an analytic expression $\theta(k)$, ii) it is shown that for $k \rightarrow \infty$ this solution converges to θ_0 under the conditions in Theorem 7.6, and iii) it is shown that the obtained estimator variance, with $z^{\text{opt}}(k)$ and F^{opt} , is equal to the theoretical lower bound.

Part i) The cost function V_{IV} is quadratic $\|x\|^2 = x^\top x$ and linear in θ , hence, a sufficient condition for the minimizer is that $\frac{\partial V(\theta)}{\partial \theta} = 0$. By substituting (7.14) in (7.13) and equating the derivative to zero, it follows that the minimizer is $\theta(k) = (R_k^\top R_k)^{-1} R_k^\top U_k$ with

$$R_k = \sum_{i=1}^k z(i)F(q^{-1})\phi^\top(i), \quad U_k = \sum_{i=1}^k z(i)F(q^{-1})u(i),$$

and $(R_k^\top R_k)$ is non-singular.

Part ii) To prove consistency, i.e., $(\theta - \theta_0) \rightarrow 0$ for $k \rightarrow \infty$ with probability one, under the conditions in Theorem 7.6, use the relation $u(k) = \phi^\top \theta_0 + \bar{v}(k)$ with $\bar{v}(k) = -A_0 v(k)$, and define $V_k = \sum_{i=1}^k z(i)F(q^{-1})\bar{v}(i)$ to write $U_k = R_k \theta_0 + V_k$. Consequently, the estimation error is

$$(\theta(k) - \theta_0) = (R_k^\top R_k)^{-1} R_k^\top V_k,$$

using (Söderström and Stoica, 1989, Lemma B.2), where it is shown that under mild conditions

$$\lim_{k \rightarrow \infty} R_k = \mathbb{E} [z(k)F(q^{-1})\phi_r(k)] =: R \quad (7.41)$$

$$\lim_{k \rightarrow \infty} V_k = \mathbb{E} [z(k)F(q^{-1})\bar{v}(k)] =: V, \quad (7.42)$$

with $\phi_r(k)$ the noise-free part of $\phi(k)$. Consequently, the asymptotic estimation error, i.e., for $k \rightarrow \infty$, $(\theta_0 - \theta) = (R^\top R)^{-1} R^\top V$ is zero if R is non-singular, and $\mathbb{E} z(i)F(q^{-1})\bar{v}(i) = 0$, which is the case since $z^{\text{opt}}(k)$ in (7.16) is uncorrelated with v .

Part iii) The final part of the proof is based on (Söderström and Stoica, 1989, Chapter 8.2), where it is shown that $(\theta(k) - \theta_0) \rightarrow \mathcal{N}(0, P_{\text{IV}})$ for $k \rightarrow \infty$ with

$$P_{\text{IV}} = (R^\top R)^{-1} R^\top \Gamma R (R^\top R)^{-1}, \quad (7.43)$$

where $R = \lim_{k \rightarrow \infty} \sum_{i=1}^k z(i)F(q^{-1})\phi^\top(i)$ under mild assumptions in Söderström and Stoica (2002), and

$$\Gamma = \sigma_v^2 \mathbb{E}[F(q^{-1})H(q^{-1})z(k)][F(q^{-1})H(q^{-1})z(k)]^\top, \quad (7.44)$$

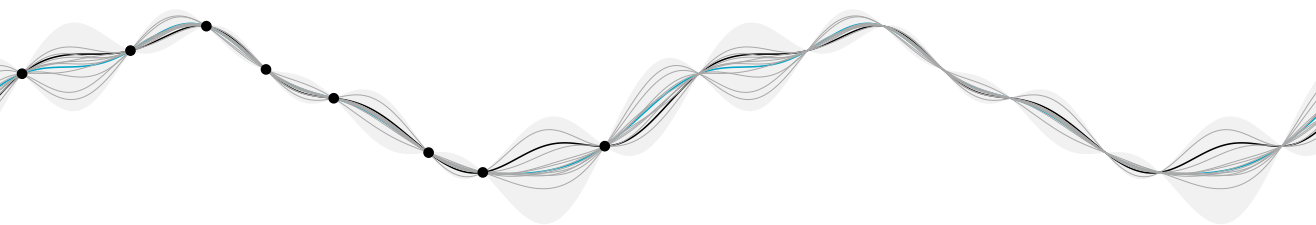
with $H(q^{-1}) = A(q^{-1}, \theta_0)$, a proof of this result can be found in (Söderström and Stoica, 1989, Appendix A8.1). In Söderström and Stoica (1989) it is also proven that

$$P_{\text{IV}} \geq P_{\text{IV}}^{\text{opt}} = [\sigma_v^2 \mathbb{E}([A_0^{-1} \phi_r(k)]^\top [A_0^{-1} \phi_r(k)])]^{-1}. \quad (7.45)$$

Finally, it remains to show that by substitution of (7.16) and (7.17) in (7.43) it follows that $P_{\text{IV}} = P_{\text{IV}}^{\text{opt}}$, which is outlined in detail in (Söderström and Stoica, 1989, p. 274), which completes the proof.

Part IV

Closing



CHAPTER

8

Conclusions and Recommendations

8.1 Conclusions

The control algorithms developed in this thesis enable improvement via the rejection of repeating disturbances and automated feedforward tuning for tracking of varying references up to the repeatable part of the error. The developed approaches exploit the intersection between control and machine learning techniques for mechatronic systems. The methods improve control of future mechatronic systems with ever-increasing performance and flexibility demands while at the same time having a strong desire for cost-efficient mechatronic designs. This leads to more complexity for controller design, for both tracking of varying reference tasks and the rejection of disturbance with underlying structures, for these challenges user-friendly control solutions are presented.

8.1.1 Learning for disturbance rejection

The first main contribution of this thesis addresses the rejection of unknown disturbances (research challenge R1), by developing a Gaussian-process-based internal model for repetitive control (RC). Disturbances often exhibit underlying periodic structures that are not directly visible in the time domain. In this situation, traditional temporal RC, which can compensate for time-domain periodic disturbances, is not directly applicable. The developed GP-based buffer for RC enables the elegant incorporation of these underlying periodic structures as prior knowledge in the form of a covariance function. The result is a new systematic design approach with a few high-level tuning knobs for RC that is applicable to a wide range of disturbances.

The first part of this contribution is tailored toward the rejection of a wide variety of temporal disturbances that are not necessarily periodic in the time

domain. The new design approach enables the rejection of single-period and multi-period disturbances, and disturbances with time-domain periods that do not fit in the traditional FIR buffers. Moreover, by appropriate selection of the covariance function the developed RC is robust for either non-repeating disturbances or variations in the disturbance period time, thereby recovering Steinbuch et al. (2007) as a special case in a single generic approach. A second major advantage is that due to the extrapolation capabilities of the GP-based buffer, compensation is enabled in the first period. This approach is presented in Chapter 2.

Rejection of disturbances that are not repeating in the time-domain but rather in, e.g., the position-domain, commutation-angle domain, or combinations of these at the same time, is enabled. This is encountered in many industrial applications, e.g., a non-perfect roller or a two-dimensional commutation disturbance in a magnetically levitated platform. The presented approach learns a continuous potentially multi-dimensional disturbance model from the inherently non-equidistant data points in the specific domains. This is established in the general framework by using spatially periodic prior knowledge or a multi-dimensional covariance function, see Chapter 3 and Chapter 4 respectively. In addition, a sparse approximation is presented to reduce the computation load of the GP in Chapter 3. The method is validated on an industrial substrate carrier with spatial disturbances.

The final part of the first contribution is the rejection of torque ripple disturbances that are spatially-periodic and do not require continuous updating over time, e.g., as encountered in the coarse pointing assembly (CPA) in Chapter 5. This is enabled with a computationally efficient two-step design routine. In the first step, GP-based RC outlined Chapter 2 is employed to learn a continuous temporal model of the disturbance at constant velocity. In the second step, the time-domain data is used to learn a GP-based model in the commutation-angle domain for compensation at arbitrary velocity variations. This is experimentally validated on the CPA resulting in an automated routine that is a key enabler for free-space optical communication with the CPA, see Chapter 5.

In summary, the combination of these four chapters is a complete and systematic framework to reject disturbances for future complex systems by employing an intelligent function estimator from the machine learning community in the traditional RC approach.

8.1.2 Feedforward tuning for flexible motion tasks

The second main contribution of this thesis enables on-line learning of feedforward parameters from data for the tracking of both batch-wise repeating reference tasks and varying reference tasks. The developed approach enables fast learning, even during a batch, by continuous estimation of feedforward parameters from data during closed-loop operation. This yields a big performance

improvement for future systems, where traditional approaches are either not applicable due to strict limitations on the task flexibility or require cumbersome and time-consuming tuning procedures.

A unified framework is developed for automatic feedforward controller tuning in both on-line and batch-wise settings for non-resetting tasks. A recursive least squares estimator optimizes the feedforward controller parameters in real-time using data. Furthermore, a detailed statistical analysis is provided in the presence of measurement noise, indicating the possible hazard of on-line feedforward parameter estimation in a closed-loop setting. A practical approach is presented to mitigate the effects of noise on the feedforward parameters. The theoretical claims and developed approach are experimentally validated on a benchmark motion system. This contribution is presented in Chapter 6.

The final contribution aims at the optimal tuning of feedforward parameters in a closed-loop setting and in the presence of noise by exploiting instrumental variable (IV) estimation. The approach is tailored towards varying and non-resetting point-to-point references, as encountered in the wafer-stage in Figure 1.1, to which traditional batch-wise learning approaches are not applicable. This is enabled with a recursive IV estimator which has an analytic solution. The approach approximates the optimal IV estimator by employing a recursive bootstrapping procedure, resulting in performance improvements within a batch even in the presence of noise as experimentally validated on the same benchmark motion system. This contribution is presented in Chapter 7.

The second contribution provides both theoretical analysis and a practical method for automatic tuning of feedforward parameters from data for complex systems.

8.2 Recommendations

Based on the insight that is obtained from the research in this thesis, the following further improvements are envisioned for the control of future mechatronic systems.

A general observation is that the research in this thesis confirms that the field of machine learning has a lot to offer for control of future systems, to improve both performance as well as user-friendliness. This is confirmed in many other related studies, see, e.g., Aarnoudse and Oomen (2020); Alam et al. (2020); Balta et al. (2021); Banka and Devasia (2018); Oomen (2018); Oomen and Rojas (2017); Van Meer et al. (2021). It is recommended to further investigate the potential benefit of machine learning and artificial intelligence approaches for control of future mechatronic systems, for example with neural networks along the lines of Aarnoudse et al. (2021), Gaussian processes in various control related approaches (Poot et al., 2022), reinforcement learning as in Berkenkamp et al. (2017); Poot et al. (2020), and deep learning for system identification as

in Ljung et al. (2020). Consider the following more specific recommendations in line with the research in this thesis.

Uncertainties in repetitive control: The internal disturbance model that is learned using GP-based RC results in a posterior distribution, i.e., a mean and variance. In the approach presented in this thesis, the posterior mean is utilized for compensation. A recommendation is to utilize the posterior variance, which is a measure of confidence for the learned disturbance model, see Chapter 2-4. It is envisioned that the confidence bounds obtained with the GP-based buffer can be used as a time-varying or position-varying learning gain for RC, e.g., if the estimate is unreliable then the learning gain is preferably small and vice versa.

Automated tuning of hyperparameters: In the current GP RC approach the hyperparameters, e.g., smoothness and periodicity, are tuned manually. To make this approach even more user-friendly, data of an a priori performed experiment can be used to automatically tune these hyperparameters, e.g., by optimizing the marginal likelihood with respect to the hyperparameters, see Pilonetto et al. (2014); Williams and Rasmussen (2006).

Model-free repetitive control: Traditional repetitive control (Hara et al., 1988) requires a parametric learning filter that contains a model of the closed-loop system. If the model quality is poor, then a robustness filter is required for stability leading to non-perfect learning. On the other hand, data-driven approaches appeared where (inverse) GP-based models are learned from data (Blanken and Oomen, 2020; de Rozario and Oomen, 2019; Lataire and Chen, 2016; Pilonetto et al., 2014). Moreover, recursive approaches appeared to further improve the computational efficiency of GP regression in an on-line setting (Bijl et al., 2017). It is envisioned that combining these approaches for estimating the learning filter, in combination with the GP-based RC approach in this thesis can potentially contribute to a model-free RC approach along the lines of, e.g., De Rozario and Oomen (2019).

Direct learning for optimal commutation functions: An automated procedure is developed to learn a static commutation function for torque ripple compensation. In practice, there is one commutation function per coil that has to be designed while taking into account the design trade-offs, e.g., power efficiency versus performance. Taken together these may result in a torque ripple. Direct learning of these commutation functions as a GP from data can potentially improve the performance by exploiting more freedom and automatically taking into account these trade-offs, while at the same time automating the commutation function design.

Adaptive feedforward for parameter varying systems: Model-based feedforward controller design for positioning systems that exhibit parameter varying dynamics often leads to a cumbersome design procedure. These linear parameter varying (LPV) systems can potentially benefit from learning of position-dependent feedforward parameters (Bloemers et al., 2018; de Rozario et al., 2017) using instrumental variables, but also Gaussian processes are promising for LPV feedforward, see Van Haren et al. (2022). By sufficiently fast adaptation or direct learning of an LPV model for feedforward as in Tóth et al. (2012), it is envisioned that user-friendly feedforward tuning and performance for LPV systems can be improved.

Selection of basis functions: It is shown in Chapter 6 and Chapter 7 that the basis functions for feedforward control are currently selected using physical insights. Moreover, the basis functions have a significant impact on the estimation performance if measurement noise is present, i.e., in such a situation it might be favorable to choose different basis functions. A topic for future research is the automatic selection or optimization of basis functions as in, e.g., Oomen and Rojas (2017). Addressing a trade-off between over-fitting and bias due to under-modeling, and by altering basis functions to reduce the effect of noise at the cost of introducing a modeling bias, might be beneficial.

Bibliography

- Aarnoudse, L., Ohnishi, W., Poot, M., Tacx, P., Srijbosch, N., and Oomen, T. (2021). Control-Relevant Neural Networks for Intelligent Motion Feed-forward. In *2021 IEEE International Conference on Mechatronics (ICM)*. IEEE, pages 1–6.
- Aarnoudse, L. and Oomen, T. (2020). Model-Free Learning for Massive MIMO Systems: Stochastic Approximation Adjoint Iterative Learning Control. *IEEE Control Systems Letters*, 5(6):1946–1951.
- Ai, W., Wu, M., Li, X., and Li, X. (2021). Active Disturbance Rejection Based Iterative Learning Control for Direct Torque Control of Switched Reluctance Motor Drive. In *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS)*. IEEE, pages 1379–1384.
- Akesson, B. and Goossens, K. (2011). *Memory controllers for real-time embedded systems*. Springer.
- Alam, M. F., Shtein, M., Barton, K., and Hoelzle, D. J. (2020). Autonomous Manufacturing Using Machine Learning: A Computational Case Study With a Limited Manufacturing Budget. In *International Manufacturing Science and Engineering Conference*. Volume 84263. American Society of Mechanical Engineers, V002T07A009.
- Almada-Lobo, F. (2015). The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES). *Journal of innovation management*, 3(4):16–21.
- Altın, B., Willems, J., Oomen, T., and Barton, K. (2017). Iterative learning control of iteration-varying systems via robust update laws with experimental implementation. *Control Engineering Practice*, 62:36–45.
- Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation of robots by learning. *Journal of Robotic systems*, 1(2):123–140.
- Åström, K. J. and Wittenmark, B. (2013). *Adaptive Control: Second Edition*. Dover Publications, INC. Mineola, New York.
- Balta, E. C., Barton, K., Tilbury, D. M., Rupenyan, A., and Lygeros, J. (2021). Learning-based repetitive precision motion control with mismatch compensation. *arXiv preprint arXiv:2111.10246*.

- Banka, N. and Devasia, S. (2018). Application of iterative machine learning for output tracking with magnetic soft actuators. *IEEE/ASME Transactions on Mechatronics*, 23(5):2186–2195.
- Beltman, A.-J., Plak, R., Hazenberg, J., Pulles, R., Brals, B., and van Ooik, G. (2012a). Improved ink registration through advanced steel belt steering. In *NIP & Digital Fabrication Conference*. Volume 2012. 1. Society for Imaging Science and Technology, pages 222–225.
- Beltman, A.-J., Plak, R., Hazenberg, J., Pulles, R., Brals, B., and van Ooik, G. (2012b). Improved Ink Registration through Advanced Steel Belt Steering. In *NIP & Digital Fabrication Conference*. Volume 2012. 1. Society for Imaging Science and Technology, pages 222–225.
- Beltman, A.-J. J., Brals, A., Plak, R., and Simons, J. A. F. M. (2014). Belt conveyor with an actuator for moving the belt in a lateral direction. U.S. Patent 8,807,331.
- Berkenkamp, F., Moriconi, R., Schoellig, A., and Krause, A. (2016). Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4661–4666. DOI: 10.1109/CDC.2016.7798979.
- Berkenkamp, F., Turchetta, M., Schoellig, A. P., and Krause, A. (2017). Safe model-based reinforcement learning with stability guarantees. *arXiv preprint arXiv:1705.08551*.
- Bernstein, D. (2009). *Matrix mathematics: theory, facts, and formulas*. Princeton university press.
- Bijl, H., Schön, T., van Wingerden, J., and Verhaegen, M. (2017). System identification through online sparse Gaussian process regression with input noise. *IFAC Journal of Systems and Control*, 2:1–11.
- Bijl, H., van Wingerden, J., Schön, T., and Verhaegen, M. (2015). Online sparse Gaussian process regression using FITC and PITC approximations. *IFAC-PapersOnLine*, 48(28):703–708.
- Bitmead, R and Anderson, B. (1980). Performance of adaptive estimation algorithms in dependent random environments. *IEEE Transactions on Automatic Control*, 25(4):788–794.
- Blanken, L., Bevers, P., Koekebakker, S., and Oomen, T. (2020a). Sequential Multiperiod Repetitive Control Design With Application to Industrial Wide-Format Printing. *IEEE/ASME Transactions on Mechatronics*, 25(2):770–778.
- Blanken, L., Boeren, F., Bruijnen, D., and Oomen, T. (2017a). Batch-to-Batch Rational Feedforward Control: From Iterative Learning to Identification Approaches, With Application to a Wafer Stage. *IEEE/ASME Transactions on Mechatronics*, 22(2):826–837.

- Blanken, L., Hazelaar, T., Koekebakker, S., and Oomen, T. (2017b). Multivariable repetitive control design framework applied to flatbed printing with continuous media flow. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, pages 4727–4732.
- Blanken, L., Koekebakker, S., and Oomen, T. (2020b). Multivariable Repetitive Control: Decentralized Designs with Application to Continuous Media Flow Printing. *IEEE/ASME Transactions on Mechatronics*.
- Blanken, L. and Oomen, T. (2019). Multivariable iterative learning control design procedures: From decentralized to centralized, illustrated on an industrial printer. *IEEE Transactions on Control Systems Technology*.
- Blanken, L. and Oomen, T. (2020). Kernel-based identification of non-causal systems with application to inverse model control. *Automatica*, 114:108830.
- Bloemers, T., Proimadis, I., Kasemsinsup, Y., and Tóth, R. (2018). Parameter-dependent feedforward strategies for motion systems. In *2018 Annual American Control Conference (ACC)*. IEEE, pages 2017–2022.
- Boeren, F., Bareja, A., Kok, T., and Oomen, T. (2016). Frequency-Domain ILC Approach for Repeating and Varying Tasks: With Application to Semiconductor Bonding Equipment. *IEEE/ASME Transactions on Mechatronics*, 21(6):2716–2727.
- Boeren, F., Bruijnen, D., van Dijk, N., and Oomen, T. (2014). Joint input shaping and feedforward for point-to-point motion: Automated tuning for an industrial nanopositioning system. *Mechatronics*.
- Boeren, F., Blanken, L., Bruijnen, D., and Oomen, T. (2018). Optimal estimation of rational feedforward control via instrumental variables: with application to a wafer stage. English. *Asian Journal of Control*, 20(1):1–18.
- Boeren, F., Bruijnen, D., and Oomen, T. (2017). Enhancing feedforward controller tuning via instrumental variables: with application to nanopositioning. *International Journal of Control*, 90(4):746–764.
- Boeren, F., Oomen, T., and Steinbuch, M. (2015). Iterative motion feedforward tuning: A data-driven approach based on instrumental variable identification. *Control Engineering Practice*, 37:11–19.
- Boerlage, M., Steinbuch, M., Lambrechts, P., and van de Wal, M. (2003). Model-based feedforward for motion systems. In *Proceedings of 2003 IEEE Conference on Control Applications*. Volume 2. IEEE, pages 1158–1163.
- Bolder, J. and Oomen, T. (2015). Rational Basis Functions in Iterative Learning Control—With Experimental Verification on a Motion System. *IEEE Transactions on Control Systems Technology*, 23(2):722–729.
- Bolder, J., van Zundert, J., Koekebakker, S., and Oomen, T. (2017). Enhancing Flatbed Printer Accuracy and Throughput: Optimal Rational Feedforward Controller Tuning Via Iterative Learning Control. *IEEE Transactions on Industrial Electronics*, 64(5):4207–4216.

- Bristow, D. A., Tharayil, M., and Alleyne, A. G. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114.
- Butler, H. (2012). Adaptive feedforward for a wafer stage in a lithographic tool. *IEEE Transactions on Control Systems Technology*, 21(3):875–881.
- Butterworth, J. A., Pao, L. Y., and Abramovitch, D. Y. (2008). The effect of nonminimum-phase zero locations on the performance of feedforward model-inverse control techniques in discrete-time systems. In *2008 American control conference*. IEEE, pages 2696–2702.
- Cao, Z. and Ledwich, G. (2002). Adaptive repetitive control to track variable periodic signals with fixed sampling rate. *IEEE/ASME Transactions on Mechatronics*, 7(3):378–384.
- Čech, M., Beltman, A.-J., and Ozols, K. (2019). I-MECH-Smart System Integration for Mechatronic Applications. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, pages 843–850.
- Chang, W., Suh, I., and Oh, J. (1998). Synthesis and analysis of digital multiple repetitive control systems. In *Proceedings of the 1998 American Control Conference. ACC*. Volume 5. IEEE, pages 2687–2691.
- Chen, C.-L. and Chiu, G. T.-C. (February 2008). Spatially periodic disturbance rejection with spatially sampled robust repetitive control. *Journal of Dynamic Systems, Measurement, and Control*, 130(2). DOI: 10.1115/1.2837306.
- Chen, C.-L. and Yang, Y.-H. (2007). Adaptive repetitive control for uncertain variable-speed rotational motion systems subject to spatially periodic disturbances. In *2007 American Control Conference*. IEEE, pages 564–569.
- Chen, T., Ohlsson, H., and Ljung, L. (2012). On the estimation of transfer functions, regularizations and Gaussian processes - Revisited. *Automatica*, 48(8):1525–1535.
- Chen, X. and Tomizuka, M. (2011). A minimum parameter adaptive approach for rejecting multiple narrow-band disturbances with application to hard disk drives. *IEEE Transactions on Control Systems Technology*, 20(2):408–415.
- Chen, X. and Tomizuka, M. (2013). New repetitive control with improved steady-state performance and accelerated transient. *IEEE Transactions on Control Systems Technology*, 22(2):664–675.
- Cuiyan, L., Dongchun, Z., and Xianyi, Z. (2004). A survey of repetitive control. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*(IEEE Cat. No. 04CH37566). Volume 2. IEEE, pages 1160–1166.
- De Rozario, R., Fleming, A., and Oomen, T. (2019). Finite-time learning control using frequency response data with application to a nanopositioning stage. *IEEE/ASME Transactions on Mechatronics*, 24(5):2085–2096.

- De Rozario, R. and Oomen, T. (2019). Data-driven iterative inversion-based control: Achieving robustness through nonlinear learning. *Automatica*, 107:342–352.
- De Rozario, R., Oomen, T., and Steinbuch, M. (2017). Iterative learning control and feedforward for LPV systems: Applied to a position-dependent motion system. In *2017 American Control Conference (ACC)*. IEEE, pages 3518–3523.
- Desoer, C. A. and Vidyasagar, M. (2009). *Feedback systems: input-output properties*. SIAM.
- Devasia, S. (2002). Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Transactions on Automatic Control*, 47(11):1865–1871.
- Devasia, S. (2017). Iterative machine learning for output tracking. *IEEE Transactions on Control Systems Technology*, 27(2):516–526.
- Durrande, N., Ginsbourger, D., and Roustant, O. (2011). Additive kernels for Gaussian process modeling. *arXiv preprint*.
- Duvenaud, D. (2014). Automatic model construction with Gaussian processes. PhD thesis, University of Cambridge.
- Evers, E., van Tuijl, N., Lamers, R., and Oomen, T. (2019). Identifying thermal dynamics for precision motion control. *IFAC-PapersOnLine*, 52(15):73–78.
- Francis, B. and Wonham, W. (1976). The internal model principle of control theory. *Automatica*, 12(5):457–465.
- Franklin, G. F., Powell, J. D., and Emami-Naeini, A. (2015). *Feedback control of dynamic systems*. Prentice hall Upper Saddle River, NJ.
- Freedman, D. A. (2009). *Statistical models: theory and practice*. cambridge university press.
- Frigola, R., Lindsten, F., Schön, T. B., and Rasmussen, C. E. (2013). Bayesian inference and learning in Gaussian process state-space models with particle MCMC. In *Advances in Neural Information Processing Systems*, pages 3156–3164.
- Fujimoto, H. (2009). RRO compensation of hard disk drives with multirate repetitive perfect tracking control. *IEEE Transactions on Industrial Electronics*, 56(10):3825–3831.
- Fujimoto, H. and Yao, B. (2005). Multirate adaptive robust control for discrete-time non-minimum phase systems and application to linear motors. *IEEE/ASME Transactions on mechatronics*, 10(4):371–377.
- Gao, X. and Mishra, S. (2014). An iterative learning control algorithm for portability between trajectories. In *2014 American Control Conference*, pages 3808–3813.

- Gawronski, W. K. (2004). *Advanced Structural Dynamics & Active Control of Structures*. Edited by F. F. Ling. Volume 411 p. Springer.
- Gilson, M., Garnier, H., Young, P. C., and Van den Hof, P. M. (2011). Optimal instrumental variable method for closed-loop identification. *IET control theory & applications*, 5(10):1147–1154.
- Gilson, M. and Van Den Hof, P. (2005). Instrumental variable methods for closed-loop system identification. *Automatica*, 41(2):241–249.
- Goodwin, G. and Sin, K. (2014). *Adaptive filtering prediction and control*. Courier Corporation.
- Gorinevsky, D., Torfs, D. E., and Goldenberg, A. A. (1997). Learning approximation of feedforward control dependence on the task parameters with application to direct-drive manipulator tracking. *IEEE Transactions on Robotics and Automation*, 13(4):567–581.
- Gregory, M., Heine, F. F., Kämpfner, H., Lange, R., Lutzer, M., and Meyer, R. (2012). Commercial optical inter-satellite communication at high data rates. *Optical Engineering*, 51(3):031202.
- Griñó, R. and Costa-Castelló, R. (2005). Digital repetitive plug-in controller for odd-harmonic periodic references and disturbances. *Automatica*, 41(1):153–157.
- Haghi, M., Yao, Y., Goswami, D., and Goossens, K. (2020). Parallel implementation of iterative learning controllers on multi-core platforms. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, pages 1704–1709.
- Hara, S., Yamamoto, Y., Omata, T., and Nakano, M. (1988). Repetitive control system: A new type servo system for periodic exogenous signals. *IEEE Transactions on Automatic Control*, 33(7):659–668.
- Hillerstrom, G. (1996). Adaptive suppression of vibrations - a repetitive control approach. *IEEE Transactions on Control Systems Technology*, 4(1):72–78. DOI: 10.1109/87.481769.
- Hoelzle, D. J. and Barton, K. L. (2016). On Spatial Iterative Learning Control via 2-D Convolution: Stability Analysis and Computational Efficiency. *IEEE Transactions on Control Systems Technology*, 24(4):1504–1512.
- Hoelzle, D., Alleyne, A., and Johnson, A. W. (2011). Basis Task Approach to Iterative Learning Control With Applications to Micro-Robotic Deposition. *IEEE Transactions on Control Systems Technology*, 19(5):1138–1148.
- Holz, M., Reuter, C., Reum, A., Ahmad, A., Hofmann, M., Ivanov, T., Rangelow, I., Stauffenberg, J., Manske, E., Du, C., et al. (2019). High throughput AFM inspection system with parallel active cantilevers. In *Photomask Technology 2019*. Volume 11148. International Society for Optics and Photonics, 111481E.

- Huo, X., Wang, M., Liu, K.-Z., and Tong, X. (2019). Attenuation of Position-Dependent Periodic Disturbance for Rotary Machines by Improved Spatial Repetitive Control With Frequency Alignment. *IEEE/ASME Transactions on Mechatronics*, 25(1):339–348.
- Ioannou, P. A. and Sun, J. (1996). *Robust adaptive control*. Volume 1. PTR Prentice-Hall Upper Saddle River, NJ.
- Janot, A., Vandanjon, P.-O., and Gautier, M. (2013). A Generic Instrumental Variable approach for Industrial Robot Identification. *IEEE Transactions on Control Systems Technology*, 22(1):132–145.
- Jidling, C., Hendriks, J., Wahlström, N., Gregg, A., Schön, T. B., Wensrich, C., and Wills, A. (2018). Probabilistic modelling and reconstruction of strain. *Nuclear Instruments and Methods in Physics Research*, 436:141–155.
- Khalil, H. K. (2002). *Nonlinear systems*. Third Edition edition. Prentice Hall.
- Kramer, L., Peters, J., Voorhoeve, R., Witvoet, G., and Kuiper, S. (2020). Novel motorization axis for a Coarse Pointing Assembly in Optical Communication Systems. *IFAC-PapersOnLine*, 53(2):8426–8431.
- Kramer, L., van den Dool, T., and Witvoet, G. (2019). Demonstrator for nano-precision multi-agent MagLev positioning platform for high throughput metrology. *IFAC-PapersOnLine*, 52(15):471–476.
- Kurniawan, E., Cao, Z., and Man, Z. (2014). Design of robust repetitive control with time-varying sampling periods. *IEEE Transactions on Industrial Electronics*, 61(6):2834–2841.
- Lambrechts, P., Boerlage, M., and Steinbuch, M. (2005). Trajectory planning and feedforward design for electromechanical motion systems. *Control Engineering Practice*, (13):145–157.
- Lataire, J. and Chen, T. (2016). Transfer function and transient estimation by Gaussian process regression in the frequency domain. *Automatica*, 72:217–229.
- Li, P. Y. (2015). Prototype angle-domain repetitive control-affine parameterization approach. *Journal of Dynamic Systems, Measurement, and Control*, 137(12).
- Li, S., Yang, J., Chen, W., and Chen, X. (2014). *Disturbance observer-based control: methods and applications*. CRC press.
- Lieder, M. and Rashid, A. (2016). Towards circular economy implementation: a comprehensive review in context of manufacturing industry. *Journal of cleaner production*, 115:36–51.
- Liu, H., Ong, Y.-S., Shen, X., and Cai, J. (2020). When Gaussian process meets big data: A review of scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems*.

- Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR.
- Ljung, L., Andersson, C., Tiels, K., and Schön, T. B. (2020). Deep learning and system identification. *IFAC-PapersOnLine*, 53(2):1175–1181.
- Longman, R. W. (2010). On the theory and design of linear repetitive control systems. *European Journal of Control*, 16(5):447–496.
- Mack, C. (2008). *Fundamental principles of optical lithography: the science of microfabrication*. John Wiley & Sons.
- Manayathara, T. J., Tsao, T.-C., and Bentsman, J. (1996). Rejection of unknown periodic load disturbances in continuous steel casting process using learning repetitive control approach. *IEEE Transactions on Control Systems Technology*, 4(3):259–265.
- Mazzoleni, M., Scandella, M., Formentin, S., and Previdi, F. (2020). Enhanced kernels for nonparametric identification of a class of nonlinear systems. In *2020 European Control Conference (ECC)*, pages 540–545. DOI: 10.23919/ECC51009.2020.9143785.
- McHutchon, A. and Rasmussen, C. (2011). Gaussian process training with input noise. *Advances in Neural Information Processing Systems*, 24:1341–1349.
- Merry, R., Kessels, D., Heemels, W., van de Molengraft, M., and Steinbuch, M. (2011). Delay-varying repetitive control with application to a walking piezo actuator. *Automatica*, 47(8):1737–1743.
- Miller, T. J. E. (2001). *Electronic control of switched reluctance machines*. Elsevier.
- Moore, G. E. (1995). Lithography and the future of Moore’s law. In *Integrated Circuit Metrology, Inspection, and Process Control IX*. Volume 2439. International Society for Optics and Photonics, pages 2–17.
- Mooren, N., Witvoet, G., and Oomen, T. (2022a). Gaussian Process Repetitive Control with Application to an Industrial Substrate Carrier Systems with Spatial Disturbances. (*Submitted for journal publication*).
- Mooren, N., van Meer, M., Witvoet, G., and Oomen, T. (2022b). Compensating Torque Ripples in Coarse Pointing Mechanism for Free-Space Optical Communication: A Gaussian Process Repetitive Control Approach. (*In preparation for journal submission*).
- Mooren, N., Witvoet, G., Açan, I., Kooijman, J., and Oomen, T. (2020a). Suppressing position-dependent disturbances in repetitive control: With application to a substrate carrier system. In *2020 IEEE 16th International Workshop on Advanced Motion Control (AMC)*. IEEE, pages 331–336.
- Mooren, N., Witvoet, G., and Oomen, T. (2019a). Feedforward motion control: From batch-to-batch learning to online parameter estimation. In *2019 American Control Conference (ACC)*. IEEE, pages 947–952.

- Mooren, N., Witvoet, G., and Oomen, T. (2019b). From Batch-to-Batch to online learning control: experimental motion control case study. *IFAC-PapersOnLine*, 52(15):406–411.
- Mooren, N., Witvoet, G., and Oomen, T. (2020b). Gaussian process repetitive control for suppressing spatial disturbances. *IFAC-PapersOnLine*, 53(2):1487–1492.
- Mooren, N., Witvoet, G., and Oomen, T. (2022c). A Gaussian Process Approach to Multiple Internal Models in Repetitive Control. In *IEEE 17th International Conference on Advanced Motion Control*.
- Mooren, N., Witvoet, G., and Oomen, T. (2022d). Gaussian process repetitive control: Beyond periodic internal models through kernels. *Automatica*, 140:110273.
- Mooren, N., Witvoet, G., and Oomen, T. (2022e). On-line Instrumental Variable-Based Feedforward Tuning for Non-Resetting Motion Tasks. (*Submitted for journal publication*).
- Murphy, K. (2012). *Machine learning: a probabilistic perspective*. MIT press.
- Nagahara, M. and Yamamoto, Y. (2016). Digital repetitive controller design via sampled-data delayed signal reconstruction. *Automatica*, 65:203–209.
- Oomen, T. (2018). Advanced motion control for precision mechatronics: Control, identification, and learning of complex systems. *IEEEJ Journal of Industry Applications*, 7(2):127–140.
- Oomen, T. (2019). Control for precision mechatronics. In *Encyclopedia of Systems and Control*. Springer.
- Oomen, T. and Rojas, C. R. (2017). Sparse iterative learning control with application to a wafer stage: Achieving performance, resource efficiency, and task flexibility. *Mechatronics*, 47:134–147.
- Park, S., Ko, B., Lee, H., and So, H. (2021). Rapid manufacturing of micro-drilling devices using FFF-type 3D printing technology. *Scientific Reports*, 11(1):1–9.
- Pérez-Arancibia, N., Tsao, T., and Gibson, J. (2010). A new method for synthesizing multiple-period adaptive-repetitive controllers and its application to the control of hard disk drives. *Automatica*, 46(7):1186–1195.
- Phung, T. H., Gafurov, A. N., Kim, I., Kim, S. Y., Kim, K. M., and Lee, T.-M. (2021). IoT device fabrication using roll-to-roll printing process. *Scientific Reports*, 11(1):1–11.
- Pillonetto, G. and De Nicolao, G. (2010). A new kernel-based approach for linear system identification. *Automatica*, 46(1):81–93.
- Pillonetto, G., Dinuzzo, F., Chen, T., de Nicolao, G., and Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, 50(3):657–682.

- Pipeleers, G., Demeulenaere, B., de Schutter, J., and Swevers, J. (2008). Robust high-order repetitive control: optimal performance trade-offs. *Automatica*, 44(10):2628–2634.
- Poot, M., Portegies, J., Mooren, N., van Haren, M., van Meer, M., and Oomen, T. (2022). Gaussian Processes for Advanced Motion Control. *IEEE Journal of Industry Applications*.
- Poot, M., Portegies, J., and Oomen, T. (2020). On the role of models in learning control: Actor-critic iterative learning control. *IFAC-PapersOnLine*, 53(2):1450–1455.
- Poot, M., Portegies, J., and Oomen, T. (2021). Kernel-Based Learning Control for Iteration-Varying Tasks Applied to a Printer With Friction. In *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1052–1057. DOI: 10.1109/AIM46487.2021.9517554.
- Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959.
- Reinders, J., Verkade, R., Hunnekens, B., van de Wouw, N., and Oomen, T. (2020). Improving mechanical ventilation for patient care through repetitive control. *IFAC-PapersOnLine*, 53(2):1415–1420.
- Rivera, D., Morari, M., and Skogestad, S. (1986). Internal model control: PID controller design. *Industrial & engineering chemistry process design and development*, 25(1):252–265.
- Rohrig, C. and Jochheim, A. (2001). Identification and compensation of force ripple in linear permanent magnet motors. In *Proceedings of the 2001 American Control Conference*. (Cat. No. 01CH37148). Volume 3. IEEE, pages 2161–2166.
- Saathof, R., Kuiper, S., Crowcombe, W., de Man, H., de Lange, D., van der Valk, N., Kramer, L., and Fritz, E. (2019). Opto-mechatronics system development for future intersatellite laser communications. In *Free-Space Laser Communications XXXI*. Volume 10910. International Society for Optics and Photonics, 109101B.
- Sadeghian, H., Herfst, R., Dekker, B., Winters, J., Bijnagte, T., and Rijnbeek, R. (2017). High-throughput atomic force microscopes operating in parallel. *Review of Scientific Instruments*, 88(3):033703.
- Santacreu, A. M. and Zhu, H. (2018). Manufacturing and service sector roles in the evolution of innovation and productivity. *Economic Synopses*, (2):1–3.
- Scampicchio, A., Chiuso, A., Formentin, S., and Pillonetto, G. (2019). Bayesian Kernel-Based Linear Control Design. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 822–827. DOI: 10.1109/CDC40024.2019.9029466.

- Schiff, J. (1999). *The Laplace transform: theory and applications*. Springer Science & Business Media.
- Schmidt, R. M., Schitter, G., and Rankers, A. (2020). *The design of high performance mechatronics-: high-Tech functionality by multidisciplinary system integration*. Ios Press.
- Schölkopf, B., Smola, A. J., Bach, F., et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*.
- Shan, Y. and Leang, K. (2012). Accounting for hysteresis in repetitive control design: Nanopositioning example. *Automatica*, 48(8):1751–1758.
- Shi, Y., Longman, R., and Nagashima, M. (2014). Small gain stability theory for matched basis function repetitive control. *Acta Astronautica*, 95:260–271.
- Skogestad, S. and Postlethwaite, I. (2007). *Multivariable feedback control: analysis and design*. Volume 2. Citeseer.
- Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264.
- Snelson, E. and Ghahramani, Z. (2007). Local and global sparse Gaussian process approximations. In *Artificial Intelligence and Statistics*, pages 524–531.
- Söderström, T. and Stoica, P. (1989). *System Identification*. Prentice Hall, Hemel Hempstead UK.
- Söderström, T. and Stoica, P. (2002). Instrumental variable methods for system identification. *Circuits, Systems and Signal Processing*, 21(1):1–9.
- Steinbuch, M. (2002). Repetitive control for systems with uncertain period-time. *Automatica*, 38(12):2103–2109.
- Steinbuch, M., Weiland, S., and Singh, T. (2007). Design of noise and period-time robust high-order repetitive control, with application to optical storage. *Automatica*, 43(12):2086–2095.
- Steinbuch, M. and Norg, M. L. (1998). Advanced motion control: An industrial perspective. *European Journal of Control*, 4(4):278–293.
- Strijbosch, N., Tacx, P., Verschueren, E., and Oomen, T. (2019). Commutation angle iterative learning control: Enhancing piezo-stepper actuator Waveforms. *IFAC-PapersOnLine*, 52(15):579–584.
- Strijbosch, N., Tiels, K., and Oomen, T. (2021). Hysteresis Feedforward Compensation: A Direct Tuning Approach using Hybrid-MEM-Elements. *IEEE Control Systems Letters*.
- Sun, Z. (2004). Tracking or rejecting rotational-angle dependent signals using time varying repetitive control. In *Proceedings of the 2004 American Control Conference*. Volume 1. IEEE, pages 144–149.

- Sun, Z., Zhang, Z., and Tsao, T.-C. (2009). Trajectory tracking and disturbance rejection for linear time-varying systems: Input/output representation. *Systems & Control Letters*, 58(6):452–460.
- Tjarnstrom, F. and Ljung, L. (2002). Using the bootstrap to estimate the variance in the case of undermodeling. *IEEE transactions on automatic control*, 47(2):395–398.
- Tomizuka, M. (1987). Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems, Measurement, and Control*, 109(1):65–68.
- Tomizuka, M. (2008). Dealing with periodic disturbances in controls of mechanical systems. *Annual Reviews in Control*, 32(2):193–199.
- Tóth, R., Laurain, V., Gilson, M., and Garnier, H. (2012). Instrumental variable scheme for closed-loop LPV model identification. *Automatica*, 48(9):2314–2320.
- Umlauft, J., Beckers, T., Capone, A., Lederer, A., and Hirche, S. (2020). Smart forgetting for safe online learning with Gaussian processes. In *Learning for Dynamics and Control*. PMLR, pages 160–169.
- Vaidya, S., Ambad, P., and Bhosle, S. (2018). Industry 4.0—a glimpse. *Procedia manufacturing*, 20:233–238.
- Valencia, J., Goswami, D., and Goossens, K. (2015). Composable platform-aware embedded control systems on a multi-core architecture. In *2015 Euromicro Conference on Digital System Design*. IEEE, pages 502–509.
- Van Haren, M., Poot, M., Kostic, D., van Es, R., Portegies, J., and Oomen, T. (2022). Gaussian Process Position-Dependent Feedforward: With Application to a Wire Bonder. In *Accepted for publication in IEEE 17th International Conference on Advanced Motion Control (AMC 2022)*.
- Van Meer, M., Poot, M., Portegies, J., and Oomen, T. (2021). Learning nonlinear feedforward: a Gaussian Process Approach Applied to a Printer with Friction. *arXiv preprint arXiv:2112.03805*.
- Van Zundert, J. and Oomen, T. (2017). Inverting nonminimum-phase systems from the perspectives of feedforward and ILC. *20th World Congress of the International Federation of Automatic Control*:12607–12612.
- Van Zundert, J. and Oomen, T. (2018). On inversion-based approaches for feedforward and ILC. *Mechatronics*, 50:282–291.
- Van de Wijdeven, J. and Bosgra, O. (2010). Using basis functions in iterative learning control: analysis and design theory. *Int. Journal of Control*, 83(4):661–675.
- Van den Braembussche, P., Swevers, J., and Van Brussel, H. (1998). Linear motor ripple compensation using position-triggered repetitive control. *IFAC Proceedings Volumes*, 31(27):353–357.

- Van den Hof, P. M. and Schrama, R. J. (1994). Identification and control-closed loop issues. *IFAC Proceedings Volumes*, 27(8):311–323.
- Van der Meulen, S., Tousain, R., and Bosgra, O. (2008). Fixed Structure Feed-forward Controller Design Exploiting Iterative Trials: Application to a Wafer Stage and a Desktop Printer. *Journal of Dynamics Systems, Measurements and Control*, 130.
- Van der Veen, G., Stokkermans, J., Mooren, N., and Oomen, T. (2020). How Learning Control Improves Product Quality in Semiconductor Assembly Equipment. In *ASPE 2020 Spring-Design and Control of Precision Mechatronic Systems*, pages 1–5.
- Veldman, D., Fey, R., Zwart, H., van de Wal, M., van den Boom, J., and Nijmeijer, H. (2019). Optimal thermal actuation for mitigation of heat-induced wafer deformation. *IEEE Transactions on Control Systems Technology*.
- Veronesi, M. and Visioli, A. (2014). Automatic tuning of feedforward controllers for disturbance rejection. *Industrial & Engineering Chemistry Research*, 53(7):2764–2770.
- Visioli, A. (2006). *Practical PID control*. Springer Science & Business Media.
- Vujičić, V. P. (2011). Minimization of torque ripple and copper losses in switched reluctance drive. *IEEE transactions on power electronics*, 27(1):388–399.
- Williams, C. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. Volume 2. 3. MIT press Cambridge, MA.
- Yao, W.-S., Tsai, M.-C., and Yamamoto, Y. (2013). Implementation of repetitive controller for rejection of position-based periodic disturbances. *Control Engineering Practice*, 21(9):1226–1237.
- Zhou, K., Wang, D., Zhang, B., Wang, Y., Ferreira, J., and de Haan, S. (2007). Dual-mode structure digital repetitive control. *Automatica*, 43(3):546–554.

List of publications

Peer-reviewed journal articles

- Mooren, N., Witvoet, G., and Oomen, T. (2022a). Gaussian Process Repetitive Control with Application to an Industrial Substrate Carrier Systems with Spatial Disturbances. *(Submitted for journal publication)*
- Mooren, N., Witvoet, G., and Oomen, T. (2022d). Gaussian process repetitive control: Beyond periodic internal models through kernels. *Automatica*, 140:110273
- Mooren, N., van Meer, M., Witvoet, G., and Oomen, T. (2022b). Compensating Torque Ripples in Coarse Pointing Mechanism for Free-Space Optical Communication: A Gaussian Process Repetitive Control Approach. *(In preparation for journal submission)*
- Mooren, N., Witvoet, G., and Oomen, T. (2022e). On-line Instrumental Variable-Based Feedforward Tuning for Non-Resetting Motion Tasks. *(Submitted for journal publication)*
- Poot, M., Portegies, J., Mooren, N., van Haren, M., van Meer, M., and Oomen, T. (2022). Gaussian Processes for Advanced Motion Control. *IEEE Journal of Industry Applications*

Peer-reviewed articles in conference proceedings

- Mooren, N., Witvoet, G., and Oomen, T. (2019a). Feedforward motion control: From batch-to-batch learning to online parameter estimation. In IEEE 2019 American Control Conference (ACC), pages 947-952, Philadelphia, PA, USA
- Mooren, N., Witvoet, G., and Oomen, T. (2019b). From Batch-to-Batch to online learning control: experimental motion control case study. In Preprints Joint Conference 8th IFAC Symposium on Mechatronic Systems, and 11th IFAC Symposium on Nonlinear Control Systems, pages 1013-1018, Vienna, Austria

- Goubej, M., Meeusen, S., Mooren, N., and Oomen, T. (2019). Iterative learning control in high-performance motion systems: from theory to implementation. In 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pages 851-856, Zaragoza, Spain
- Van der Veen, G., Stokkermans, J., Mooren, N., and Oomen, T. (2020). How Learning Control Improves Product Quality in Semiconductor Assembly Equipment. In Proceedings of ASPE 2020 Spring-Design and Control of Precision Mechatronic Systems, pages 1-5, Cambridge, Massachusetts, USA
- Mooren, N. Witvoet, G., Oomen, T. (2020). Compensating position-dependent disturbances in mechatronic systems: a new repetitive control framework with applications to a substrate carrier. In Proceedings of the First Euspen Special Interest Group Meeting on Precision Motion Systems & Control, Veldhoven, The Netherlands
- Mooren, N., Witvoet, G., Açı, I., Kooijman, J., and Oomen, T. (2020a). Suppressing position-dependent disturbances in repetitive control: With application to a substrate carrier system. In Proceedings of the IEEE 16th International Workshop on Advanced Motion Control (AMC), pages 331-336, Agder, Norway
- Mooren, N., Witvoet, G., and Oomen, T. (2020b). Gaussian process repetitive control for suppressing spatial disturbances. In Proceedings of the IFAC 21st Triennial World Congress, pages 1513-1518, Berlin, Germany
- Mooren, N., Witvoet, G., and Oomen, T. (2022c). A Gaussian Process Approach to Multiple Internal Models in Repetitive Control. In IEEE 17th International Conference on Advanced Motion Control, Padova, Italy

Unrelated peer-reviewed articles in conference proceedings

- Dirx, N., Mooren, N., and Oomen, T. (2021). Suppressing non-collocated disturbances in inferential motion control: with application to a wafer stage. In IEEE 2021 American Control Conference (ACC), pages 4333-4338, New Orleans, Louisiana, USA
- Mooren, N., Dirx, N., Voorhoeve, R., and Oomen, T. (2018). Compensating quasi-static disturbances for inferential control: an observer-based approach applied to a wafer stage. In Proceedings of the 2018 IEEE International Workshop on Sensing, Actuation, Motion Control, and Optimization, Tokyo, Japan

Dankwoord

Promoveren is vergelijkbaar met een berg omhoog fietsen (iets waar ik ook van kan genieten ;), de eerste kilometers vliegen voorbij, eenmaal op de helft angekommen denk je ‘hoe ga ik boven komen’, daarna wisselen gedachten als ‘wat is dit gaaf’ zich af met ‘waarom wilde ik dit ook alweer?’, zodra de laatste bocht in zicht is kan er nog een tandje bij en ga je voldaan naar de top. Na ruim 4 jaar van onderzoeken, ontdekken, schrijven, feedback ontcijferen, experimenteren, ‘opnieuw beginnen’, ‘F5-en’, ‘Matlab-en’ en ettelijke vierkantenmeters whiteboard vol gekliederd te hebben met goede (en minder goede) ideeën, aangevuld met een hoop dartscores, ligt hier nu een mijn proefschrift waar ik trots op ben!

Ik wil beginnen met diegenen bedanken die dit fantastische avontuur mogelijk gemaakt hebben, Tom en Gert! Tom, door jouw kritische blik op mijn werk, inspirerende ideeën, vele (soms onleesbare) feedback en op zijn tijd de ‘effe boos maken’ is de kwaliteit van dit proefschrift naar een hoger niveau getild. Daarnaast hebben de informele gesprekken, BBQ’s en je aanwezigheid op conferenties mij altijd gemotiveerd, mooi dat hier ook ruimte voor is. Bedankt voor het creëren van deze prettige sfeer binnen ‘Groep Oomen’, hier mag je trots op zijn! Gert, jouw onuitputtelijke enthousiasme verbaast me nog elke week! Jouw feedback op mijn werk en alles wat ik van je heb geleerd met betrekking tot de applicaties bij TNO heb ik zeer gewaardeerd. Je inzet om deze applicaties in te zetten voor mijn onderzoek heeft me een hoop kennis en motivatie opgeleverd. Tijdens onze wekelijkse meetings ben je altijd in staat geweest om mijn motivatie op peil te houden zodat ik er weer met een frisse blik tegenaan kon, bedankt voor deze prettige samenwerking!

I want to thank the members of my defense committee. Professor Antonio Visioli, Professor Kees Goossens, and Professor Kira Barton thank you for your time and effort to read my thesis and participate in the defense. Maarten, jou wil ik ook graag bedanken voor de prettige sfeer binnen CST. Gijs, bedankt voor je bijdrage vanuit Nexperia, het deelnemen in mijn commissie en het samen begeleiden van studenten.

This research is part of I-MECH, I want to thank everyone who has been involved. Both directly and indirectly during one of the ‘face-to-face’ meetings in Eindhoven, Pilsen, Madrid, and Athens which I enjoyed. I want to thank

Arend-Jan Beltman for the great organisation of I-MECH, Martin Goubey and Martin Cech for inviting me to Pilsen, Joep Kooijman and Gijs van der Veen for the collaborative work at Sioux and Nexperia. Een deel van de praktische resultaten in dit proefschrift heb ik ook te danken aan de samenwerking met studenten, daarom wil ik graag Ibrahim, Sven, Corn  , Raoul, Joost, Berend, Bilge, Tom, Abdullah en Thijs bedanken voor jullie bijdragen en de prettige samenwerking.

Ik wil mijn (oud)collega's van CST en D&C bedanken, in het bijzonder, Alex, Camiel, Fahim, Joey, Nic, Nard, Masahiro, Maurice, Enzo, Lennart, Johan, Leontine, Koen, Wataru, Mathyn, Max (2x) en Paul voor de gezelligheid op de gang, de vele koffie pauzes, lunch wandelingen en vrijdag middag borrels. Niet te vergeten de conferenties zoals de Benelux meetings, waar een flinke delegatie vanuit Eindhoven rond het middaguur in de 'aqua mundo' te vinden was voor een sessie team building. Specifiek wil ik 'Groep Oomen' bedanken voor de fantastische tijd die ik met jullie heb gehad, van Karaoke in Tokyo en Philadelphia tot de met rook gevulde hotel kamer feestjes op de Benelux.

Dan misschien wel het belangrijkste, ons kantoor, waar ik met heel veel plezier ben geweest de afgelopen jaren, o.a. voor, beleggings/klus/verf/hypotheekadvies, darten en het boeken van conferentie tripjes (met onze kosten functie voor de ideale hotel locatie). Ik heb vooral een hoop gelachen met jullie en af en toe werd er ook hard gewerkt. Nic, Max en Nard als harde kern, bedankt voor jullie bijgedrage aan dit proefschrift, zonder jullie was het niet gelukt! Van discolampen en rookmachines op conferenties tot ons gedeelde enthousiasme voor (gratis) eten, dat ga ik ontzettend missen! Ook wil ik onze burens bedanken, specifiek Fahim voor de vele 'Botjes' die we de afgelopen jaren gespeeld hebben, mogelijk had er een extra hoofdstuk in dit boekje kunnen zitten als we dat niet gedaan hadden, maar ik denk dat het essentieel is geweest ;)

I also want to thank our visiting office mates, in particular, Wataru and Masahiro. I had a great time with you teaching me a lot about Japanese cooking skills, ranging from Nissin Raoh ramen, to various types of okonomiyaki and trips to Takumi, which were oishii,          !

Mijn vrienden van 'Caf   Mooren' wil ik bedanken voor de ontspanning tijdens bijvoorbeeld vriendenweekenden en de vele BBQ's met 'kookclub Schultenbrau'. Daarnaast ook degenen waarmee ik de afgelopen jaren vele KM's op de fiets heb doorgebracht, Dave, Bram en Lennart bedanken. Bijna als laatste bedank ik mijn familie; mam, pap, Mechy, Joep, opa's, oma's, jullie support heeft er mede voor gezorgd dat ik dit kon doen afgewisseld met ontspannen. Nikki, jou bedankt ik als laatste, je hebt altijd voor me klaar gestaan en me gemotiveerd, van de lieve briefjes als ik thuis kwam, tot aan fietsen, duiken en kamperen op vakantie die momenten ga ik nooit vergeten. Zonder jou was dit niet gelukt en dat wil ik nog vele jaren voorzetten, ik hou van je!

Noud Mooren
Eindhoven, Maart 2022

About the author

Noud Mooren was born on November 21th, 1992 in Venlo, the Netherlands. After finishing secondary education in 2010 at the Bouwens van der Boijecollege in Panningen, the Netherlands, he studied Mechatronics at Fontys University of Applied Sciences in Eindhoven, the Netherlands, where he received the Bachelor of Science degree in 2017. Consequently, he studied Systems and Control at the Eindhoven University of Technology, where he received his Master of Science degree (cum laude) in 2017. During his studies, he performed internships at several industrial companies, including Frencken and Contiweb in the field of mechatronic design, dynamic modeling, and motion controller design. The research for his master's thesis, entitled 'Controlling structural deformations of a wafer stage: a disturbance-observer based approach' was conducted in cooperation with ASML Research under the supervision of Tom Oomen and Nic Dirkx.



In December 2017, Noud started his Ph.D. research in the Control Systems Technology group at the Department of Mechanical Engineering at the Eindhoven University of Technology, under the supervision of Tom Oomen and Gert Witvoet. The main results of this research are included in this thesis and focus on intelligent controller design for future mechatronic systems by combining control and machine learning. The research is part of the European project I-MECH and included a collaboration with industrial partners, including Sioux Technologies and Nexperia.

