

Feature-based motion control for near-repetitive structures

Citation for published version (APA):

Best, de, J. J. T. H. (2011). *Feature-based motion control for near-repetitive structures*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mechanical Engineering]. Technische Universiteit Eindhoven.
<https://doi.org/10.6100/IR716343>

DOI:

[10.6100/IR716343](https://doi.org/10.6100/IR716343)

Document status and date:

Published: 01/01/2011

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.

Feature-based motion control for near-repetitive structures

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op dinsdag 6 september 2011 om 14.00 uur

door

Jeroen Johannes Theodorus Hendrikus de Best

geboren te Wamel

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. M. Steinbuch

Copromotor:

dr.ir. M.J.G. van de Molengraft

disc

This dissertation has been completed in partial fulfillment of the requirements of the Dutch Institute of Systems and Control (DISC) for graduate study.

This research was financially supported by the IOP Precision Technology program of the Dutch Ministry of Economic Affairs.

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

Feature-based motion control for near-repetitive structures/ by Jeroen J.T.H. de Best. – Eindhoven : Technische Universiteit Eindhoven, 2011. Proefschrift. – ISBN: 978-90-386-2560-7

Copyright © 2011 by J.J.T.H. de Best. All rights reserved.

Typeset by the author with the pdfL^AT_EX documentation system.

Cover design: Ivo van Sluis, www.ivoontwerpt.nl, The Netherlands.

Reproduction: Ipskamp Drukkers B.V., Enschede, The Netherlands.

Summary

Feature-based motion control for near-repetitive structures

In many manufacturing processes, production steps are carried out on repetitive structures which consist of identical features placed in a repetitive pattern. In the production of these repetitive structures one or more consecutive steps are carried out on the features to create the final product. Key to obtaining a high product quality is to position the tool with respect to each feature of the repetitive structure with a high accuracy. In current industrial practice, local position sensors such as motor encoders are used to separately measure the metric position of the tool and the stage where the repetitive structure is on. Here, the final accuracy of alignment directly relies on assumptions like thermal stability, infinite machine frame stiffness and constant pitch between successive features. As the size of these repetitive structures is growing, often these assumptions are difficult to satisfy in practice.

The main goal of this thesis is to design control approaches for accurately positioning the tool with respect to the features, without the need of the aforementioned assumptions. In this thesis, visual servoing, i.e., using machine vision data in the servo loop to control the motion of a system, is used for controlling the relative position between the tool and the features. By using vision as a measurement device the relevant dynamics and disturbances are therefore measurable and can be accounted for in a non-collocated control setting.

In many cases, the pitch between features is subject to small imperfections, e.g., due to the finite accuracy of preceding process steps or thermal expansion. Therefore, the distance between two features is unknown a priori, such that setpoints can not be constructed a priori. In this thesis, a novel feature-based position measurement is proposed, with the advantage that the feature-based target position of every feature is known a priori. Motion setpoints can be defined from feature to feature without knowing the exact absolute metric position of the features before-

hand. Next to feature-to-feature movements, process steps involving movements with respect to the features, e.g., engraving or cutting, are implemented to increase the versatility of the movements. Final positioning accuracies of $10\ \mu\text{m}$ are attained.

For feature-to-feature movements with varying distances between the features a novel feedforward control strategy is developed based on iterative learning control (ILC) techniques. In this case, metric setpoints from feature to feature are constructed by scaling a nominal setpoint to handle the pitch imperfections. These scale varying setpoints will be applied during the learning process, while second order ILC is used to relax the classical ILC boundary of setpoints being the same every trial. The final position accuracy is within $5\ \mu\text{m}$, while scale varying setpoints are applied.

The proposed control design approaches are validated in practice on an industrial application, where the task is to position a tool with respect to discrete semiconductors of a wafer. A visual servoing setup capable of attaining a 1 kHz frame rate is realized. It consists of an xy-stage on which a wafer is clamped which contains the discrete semiconductor products. A camera looks down onto the wafer and is used for position feedback. The time delay of the system is 2.5 ms and the variation of the position measurement is $0.3\ \mu\text{m}$ (3σ).

Contents

Summary	iii
1 Introduction	1
1.1 Repetitive structures in high tech motion systems	1
1.2 Problem statement	2
1.3 Current vision-based control approaches	4
1.4 Research goal and approach	14
1.5 Research contributions	18
1.6 Outline of this thesis	19
2 One-dimensional feature-based motion control	21
2.1 Introduction	21
2.2 Measurement principle	24
2.3 Model-based prediction	25
2.4 Fast image processing implementation	28
2.5 Experimental setup	30
2.6 System identification	31
2.7 Integration	33
2.8 Stability analysis	34
2.9 Results	41
2.10 Conclusions	47
3 Two-dimensional feature-based motion control	49
3.1 Introduction	50
3.2 Notation	51
3.3 Feature-based positions	53
3.4 Relative feature movements	61
3.5 Experimental setup	63
3.6 Control design and stability analysis	65
3.7 Results	73

3.8	Conclusions	75
4	Iterative learning control for scale varying setpoints	77
4.1	Introduction	77
4.2	Standard ILC and normalized ILC	80
4.3	Existence of disturbances	84
4.4	Second order ILC	87
4.5	Results	96
4.6	Conclusions	101
5	Conclusions and recommendations	103
5.1	Conclusions	103
5.2	Recommendations	105
	Bibliography	107
A	Stability proof	117
B	Bilinear interpolation equivalence	119
C	Second order interpolation	121
D	Minimum and maximum Jacobian values	123
	Samenvatting	127
	Dankwoord	129
	Curriculum Vitae	131

Chapter 1

Introduction

I*N this chapter, an introduction is given on high tech motion systems which manufacture repetitive structures. Increasing demands on both accuracy and production speeds puts these machines to their limits, which leads to the problem statement of this work. Current motion control approaches of these machines are reviewed after which the goal of this thesis is defined. The research contributions are presented and finally, the outline of this thesis is given.*

1.1 Repetitive structures in high tech motion systems

In many manufacturing processes, production steps are carried out on repetitive structures consisting of identical features placed in a repetitive pattern. Examples of repetitive structures can be found in the flat panel display market like the organic light emitting diode (OLED) displays, see Fig. 1.1(a), and in the semiconductor industry, where Fig. 1.1(b) and 1.1(c) show diodes and transistors on a wafer, respectively. In the case of the OLED displays the features are the cups to be filled with organic compounds. The features on a wafer are the discrete semiconductors which need to be picked and placed for further processing. In general, the trend for this high tech motion area is to produce manufacturing systems that tend to produce 1) more accurate, 2) faster and 3) on larger surfaces. In the printing industry for example the printing resolution has become better and better through the years, the number of pages per minute has increased and the media sizes even include billboard size. In the semiconductor industry the wafer size has gradually

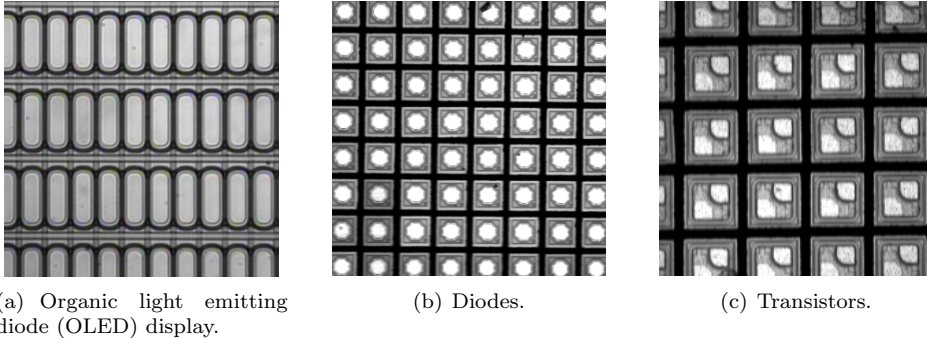


Figure 1.1: Examples of repetitive structures.

increased over time to improve the throughput and to reduce the costs, i.e., a larger wafer size results in less marginal space on the edges as a percentage of the total space and can significantly increase the yield per wafer. Moreover, less wafers need to be swapped. Regarding accuracy, Moore's law is obeyed, which describes the long-term trend in the history of manufacturing semiconductors. It states that the number of transistors that can be placed on a microchip doubles approximately every two years (Moore, 1965). In the manufacturing of displays, which consist of a repetitive grid of pixels (picture element), it is observed that there is an increase in resolution, see Fig. 1.2. At the same time a growing screen size is observed; whereas in the 1990s typical computer monitor sizes of 14" or 15" were common, nowadays 30" monitors are on the market. The increasing resolution and size is also present in the television market, where full high definition (HD) is becoming the standard and where the record-breaking (January 2010) full HD TV size is held by Panasonic with 152".

1.2 Problem statement

In the production of these repetitive structures one or more consecutive steps are carried out on the particular features of the repetitive structure to create the final product. Such production machines often consist of a tool and a stage or carrier on which the repetitive structure is to be processed. One of the possibilities for manufacturing OLED displays is using inkjet printing technology (Sturm et al., 2000) such that the tool in this case is a print head. For the production of discrete semiconductors a placement machine called a die bonder is used as a tool. Another tool, called a wire bonder, provides the electrical connection between the integrated circuit and the external leads of the semiconductor device to obtain the

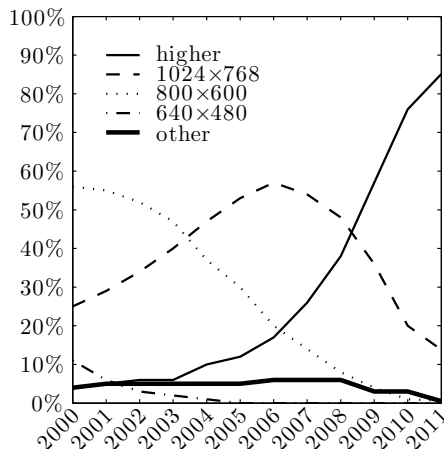


Figure 1.2: Display resolution statistics (W3schools.com, 2011).

final microchip. This work focusses on processes with point-to-point motion profiles as opposed to the continuous motions where operations are carried out during the movement. Key to obtaining a high product quality is to position the tool with respect to each feature of the repetitive structure with a high accuracy. In current industrial practice local position sensors such as motor encoders are used to measure the tool position x_t and the position of the stage x_o separately as shown in Fig. 1.3. This is referred to as an indirect measurement of $x_t - x_o$. Using such local measurements in a closed-loop control approach leads to a collocated control design. The final accuracy with which the tool can be positioned with respect to the features in this case is directly dependent on the following machine properties: 1) geometric accuracy of the mechanical construction, 2) stiffness of the mechanical construction and 3) thermal stability of the machine. Furthermore, the final accuracy also relies on assumptions with respect to the repetitive structure: 1) infinitely stiff connection between the supporting stage and the repetitive structure, 2) constant and known alignment of the repetitive structure with respect to the actuation axes, 3) infinite stiffness of the repetitive structure, 4) constant and known pitch between successive features of the repetitive structure and, finally, 5) thermal stability of the repetitive structure. In practice these assumptions are not valid when position accuracies of less than $10 \mu\text{m}$ are to be obtained. The linear thermal expansion coefficient of steel for example is approximately $15 \cdot 10^{-6} \text{ 1/K}$.

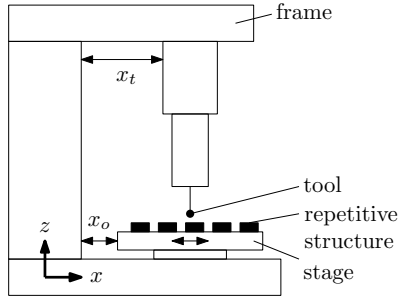


Figure 1.3: Conventional *indirect* measurement loop.

In that case, a temperature increase of only one degree in a machine part with a typical dimension in the order of one meter, results in an expansion of $15 \mu\text{m}$, which directly shows a significant influence on the attainable position accuracy. Therefore, in general the machine and the repetitive structure are *not* ideal, since the above assumptions are only partially met in practice. This leads to the following problem statement:

Investigate control design approaches for the relative positioning of a tool in a non-ideal machine with respect to the features of a non-ideal repetitive structure.

1.3 Current vision-based control approaches

The problem stated in Section 1.2 has two key ingredients, which are visualized in Fig. 1.4:

1. **non-ideal machine:** the system at hand cannot be considered ideal due to flexibilities, geometric imperfections and thermal expansion. Aligning the tool with respect to a feature poses the problem that *machine imperfections should be accounted for*,
2. **non-ideal repetitive structure:** an ideal repetitive structure is characterized by a perfectly repetitive pitch between successive features. However, small pitch imperfections cause the repetitive structure to become a non-ideal repetitive structure, such that the metric positions of the features are unknown beforehand. Aligning the tool with respect to a feature in this case poses the problem that the *metric reference is unknown a priori*.

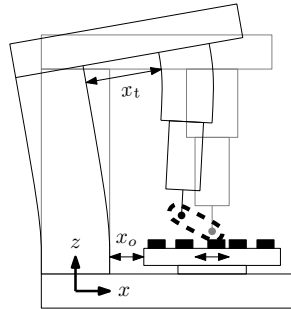


Figure 1.4: Due to, for example, geometric inaccuracies or thermal expansion the ideal machine (depicted in gray) results in a non-ideal machine (depicted in black), such that the assumed position of the tool is incorrect, emphasised by the dashed line. Also, the distance between successive features is not exactly repetitive, such that a non-ideal repetitive structure is to be considered.

In this work, machine vision (Jain et al., 1995; Sonka et al., 1999; Stegger et al., 2008) will be used to measure the relative position between the tool and the features of the repetitive structure. As opposed to the conventional indirect relative position measurement of Fig. 1.3, a direct relative position measurement of the tool relative to the feature can be obtained using vision, see Fig. 1.5. Furthermore, besides the use of the vision sensor as position measurement device, another advantage of vision is that quality inspection can be carried out, which however will not be addressed in this work. Other possible sensors to measure this relative position are inductive sensors, capacitive sensors, ultrasonic sensors or laser and fiber optic position sensors. However, some of these sensors require the repetitive structure to have specific properties such as conductivity. Ultrasonic position sensors

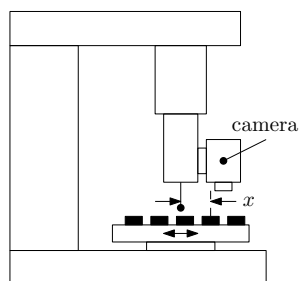


Figure 1.5: Visual *direct* measurement loop.

are directly dependent on the propagation velocity of the measurement medium, which might fluctuate as a function of temperature for example. Laser and fiber optic sensors are restricted due to the reachability of the features, i.e., the beam cannot reach the edges of all features. Machine vision is not hampered by these restrictions and is therefore used in this work.

The obtained machine vision data will be used as feedback signal in the control loop. Using machine vision data in the servo loop to control the motion of a system dates back to the 1970s (Shirai and Inoue, 1973) and is referred to as visual servo control (Chaumette and Hutchinson, 2006; Hutchinson et al., 1996) also known as visual servoing (Hill and Park, 1979), or vision-based robot control. Extensive reviews on visual servoing can be found in (Kragic and Christensen, 2002; Malis, 2002; Hutchinson et al., 1996; Corke, 2001; Hashimoto, 2003).

Many design choices are known within the field of visual servoing. Therefore, at this point an overview of the visual servoing taxonomy will be given including 1) direct visual servoing versus indirect visual servoing, 2) image-based visual servoing versus position-based visual servoing, 3) monocular visual servoing versus binocular/stereo visual servoing, 4) endpoint open-loop visual servoing versus endpoint closed-loop visual servoing and 5) eye-in-hand visual servoing versus eye-to-hand visual servoing. This visual servoing taxonomy is graphically depicted in Fig. 1.6. Later on in Section 1.4, the design choices regarding this visual servoing taxonomy are discussed, with the focus on the two issues at the beginning of this section: *non-ideal machine* and *non-ideal repetitive structure*.

Direct and indirect visual servoing

In 1980, Sanderson and Weiss (Sanderson and Weiss, 1980) introduced a taxonomy of visual servo systems. The first distinction is between *direct visual servoing* and *indirect visual servoing*. In the case of direct visual servoing, the visual controller *directly* computes the input to the system. In contrast, indirect visual servoing has a hierarchical or cascaded control architecture in which the vision system provides setpoints to low level joint controllers. The indirect visual servoing category is split up into *static look-and-move* and *dynamic look-and-move*.

Static look-and-move consists of a sequence of three independent steps (Weiss et al., 1987): 1) the system “looks” at the scene and measures the relative position between the tool and the feature, 2) the difference between its current position and where it should be is calculated and a trajectory to overcome this difference is applied to the independently closed-loop positioning system to “move” by this incremental distance, 3) the system moves to the new position. The first step is, however, *not* repeated until the system has completed the motion, i.e., during the execution of the move command, there is no feedback from the vision system. If the combined accuracy of the positioning system and vision measurement system are within the specified accuracy, this sequence needs to be executed only once.

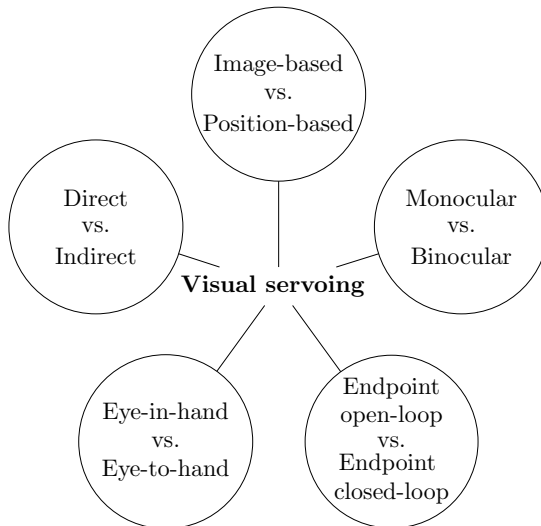


Figure 1.6: Visual servoing taxonomy.

If not, the sequence of operations is executed repeatedly until the specified accuracy is obtained. The static look-and-move approach demonstrates the concept of vision and system positioning, however it is not a dynamic control system, since each step is executed independently and in sequence. Therefore, the dynamics of each operation at each level of the hierarchy do not affect the overall system stability. Static look-and-move control approaches are found in practice for example to perform substrate alignment (Sakou et al., 1989; Nian and Tarng, 2005; Kuo et al., 2008). In many of those applications custom markers or fiducials on the substrate are searched under collocated control. Once these are found and the substrate is aligned, motor encoders define the position of the substrate from then on. With the assumption that the repetitive structure has a known predefined grid, the position of each feature can be reached by controlling the stage or tool according to the predefined distances between consecutive features using the on board motor encoders. With small variations in the distance between successive features, for example due to thermal expansion or local stretching of the repetitive structure, this assumption is not satisfied any more, which leads to a bad alignment. Therefore, an ideal repetitive structure is assumed in this case.

Other static look-and-move applications apply nominal trajectories based on the nominal distance between successive features. At the final position an image is captured of the feature of interest. From this image, the final displacement is calculated, which is translated to the current encoder values and an extra trajectory

is applied (Kolloor and lalmurugan, 1965; You et al., 1990; Verstegen et al., 2006). With a single image only a snapshot of the situation is taken. At the capture moment of the image, vibrations of the tool with respect to the particular feature can occur for example due to settling behavior or system flexibilities, such that based on the single image a wrong displacement is calculated.

Referring to the issues at the beginning of this section, the static look-and-move control approach *assumes an ideal machine* during the move commands.

In contrast to static look-and-move the *dynamic look-and-move* control approach is structured so that the three steps outlined above are executed in parallel. In this case the dynamic interaction between the levels of the hierarchy becomes critical. By far, most literature on visual servoing adopt this approach for several reasons (Espiau et al., 1992; Crétual and Chaumette, 1997; Corke and Hutchinson, 2001; Chaumette and Hutchinson, 2006, 2007). First, many applications already have an interface for accepting velocity or incremental position commands. This simplifies the construction of the visual servo system, and also makes the methods more portable. Second, the relatively low sampling rates available from vision (typically around 30-60 Hz) makes direct control of a system with complex dynamics an extremely challenging control problem. Using internal feedback with a high sampling rate generally presents the visual controller with idealized axis dynamics. Third, dynamic look-and-move separates the kinematic singularities of the mechanism from the visual controller, allowing the machine to be considered as an ideal motion device (Hutchinson et al., 1996). At this point, the two last assumptions of the machine being an ideal motion device with idealized axis dynamics is discussed in more detail.

Most dynamic look-and-move control approaches are designed to minimize an error function $e(t)$ given by

$$e(t) = s(t) - s^*, \quad (1.1)$$

where $s(t)$ is the image feature vector (most of the time a vector storing the pixel coordinates of detected points of the object) at time t and s^* is the (constant) desired image feature vector. Classically, the output of these visual controllers are reference velocities $v(t)$ to low level joint controllers. Under the assumption of rigid body dynamics, the velocities of these joints are related to the velocity of the features in the field of view by means of the image Jacobian J ,

$$\dot{s}(t) = J(s(t), Z(t))v(t). \quad (1.2)$$

This image Jacobian $J = J(s(t), Z(t))$ is dependent on the image feature vector $s(t)$ and the distance from the features to the camera $Z(t)$ (Chaumette and Hutchinson, 2006). This matrix is also called feature Jacobian (Feddemma and Mitchell, 1989), feature sensitivity matrix (Jang and Bien, 2002) and interaction matrix (Chaumette et al., 2002). Provided that the joint velocities are tracked

perfectly a control law can be derived

$$v(t) = -\lambda \hat{J}^\dagger s(t), \quad (1.3)$$

where \hat{J}^\dagger is the pseudo inverse of the estimate of the image Jacobian J (see (Chaumette and Hutchinson, 2006; Espiau et al., 1992; Chaumette and Hutchinson, 2006; Malis, 2004; Hosoda and Asada, 1994) for examples) and λ is a positive scalar. For a constant desired image feature vector, i.e., $\dot{s}^* = 0$ the following error dynamics can then be derived

$$\dot{e}(t) = -\lambda J(s(t), Z(t)) \hat{J}^\dagger e(t). \quad (1.4)$$

To assess the stability of the closed-loop system often Lyapunov analysis is used where the candidate Lyapunov function is given by $\frac{1}{2}e(t)^T e(t)$. This leads to the following condition to ensure global asymptotic stability

$$J(s(t), Z(t)) \hat{J}^\dagger \succ 0, \forall t. \quad (1.5)$$

The dynamic look-and-move control approach makes several assumptions. First, *rigid body* behavior is assumed in (1.2). Second, in (1.2) the velocity v is assumed to be the same as the one mentioned in (1.3). However, in practice the velocity in (1.3) is the applied reference velocity, whereas in (1.2) it is the current velocity, which in general are *not* the same due to the limited bandwidths of the low level velocity controllers. Third, the presence of delay (Vincze, 2000; Papanikolopoulos et al., 1993) due to image acquisition, data transfer and image processing is *not* included. Fourth, as many dynamic look-and-move applications are inherently multi rate systems, with high sample rates for the low level joint control loops and low sample rates for the high level vision control loop, the commonly used stability analysis is therefore discussable. Therefore, referring to the issues at the beginning of this section, dynamic look-and-move also *assumes an ideal machine*. Recognition of time delay and non-rigid body behavior and incorporating of these effects in the control design where done in (Corke and Good, 1992; Corke, 1995; Corke and Good, 1996).

Direct visual servoing, as opposed to indirect visual servoing, computes the input torques to the plant directly (Hutchinson et al., 1996; Malis, 2002). Sometimes it is confusing whether or not a proposed control design is direct or indirect visual servoing. In (Gangloff and de Mathelin, 2002, 2003) for example the authors state that they adopt a direct visual servoing control approach due to the absence of low-level *position* controllers. However, the proposed control design still uses a hierarchical control structure in which low-level *velocity* controllers are present. Also, (Kelly et al., 2000) uses a hierarchical control design where joint encoders are used in conjunction with the image features but claim to have a direct visual servo control approach.

For a good tracking performance and disturbance rejection a high bandwidth of the closed-loop control system is desirable. Franklin (Franklin et al., 1994) suggests that the sample rate of a digital control system must be at least four to twenty times the desired closed-loop bandwidth. With typical camera sample rates of around 50 Hz, the maximum bandwidth is therefore limited to approximately 10 Hz or even lower. In contrast, high-speed (1 kHz sample rate) direct visual servoing using massive parallel processing is reported in (Ishii et al., 1996; Ishikawa et al., 1992; Nakabo et al., 2000). The specially developed vision-chip is used in tracking micro organisms (Ogawa et al., 2005b,a) and in catching a ball in a high-speed multi-fingered hand (Namiki and Ishikawa, 2003a; Namiki et al., 2004). In (Komuro et al., 2009) a high-speed real-time vision system by integrating a CMOS image sensor and a massively parallel image processor is presented. In (Shimizu and Hirai, 2006) a specially developed CMOS in combination with a field-programmable gate array (FPGA) is used to obtain a 1 kHz direct visual servoing scheme capable of controlling a flexible link. However, nowadays standard commercially available cameras are also capable of reaching 1 kHz.

Image-based and position-based visual servoing

Next to indirect and direct visual servo control, a second distinction made by Sanderson and Weiss is between image-based visual servoing (IBVS) and position-based visual servoing (PBVS). In both concepts image features are extracted from the image. However, in PBVS a pose (position and orientation) estimation is carried out using these features in conjunction with a geometric model of the object under consideration and a known camera model (Wilson et al., 1996; Martinet and Gallice, 1999; Thuilot et al., 2002). The position error taken as the difference between the reference pose and the estimated pose is used for feedback for the vision controller. In IBVS the pose estimation is eliminated and control values are computed on the basis of the image features *directly* (Weiss et al., 1987; Espiau et al., 1992; Feddema and Mitchell, 1989; Hashimoto et al., 1991; Hashimoto, 2003).

Monocular and binocular visual servoing

A third classification is the monocular versus binocular or stereo vision. Monocular vision uses one camera, whereas in stereo vision two cameras are used. The advantage of stereo vision is that the distance of the features with respect to the camera can be estimated via triangulation. A disadvantage however is that the two cameras should be synchronized with a high accuracy in order to perform this estimation. Another disadvantage is that two images need to be processed, which is computational more demanding. Finally, two cameras are obviously twice as expensive as using only one.

Endpoint open-loop and endpoint closed-loop visual servoing

Endpoint open-loop versus endpoint closed-loop is another classification within visual servoing. In the considered applications the *tool* is to be positioned relative to the features of the repetitive structure. In most cases however the *camera* is positioned relative to the features. The position of the tool relative to the feature is determined indirectly by its known kinematic relationship with the camera. Errors in this kinematic relationship lead to positioning errors which cannot be observed by the system. Observing the tool directly makes it possible to sense and correct for such errors. In general, there is no guarantee on the positioning accuracy of the system unless *both* the tool and the feature are observed. To emphasize this distinction, we refer to systems that only observe the feature as endpoint open-loop (EOL) systems, and systems that observe both the tool and the feature as endpoint closed-loop (ECL) systems (Hutchinson et al., 1996).

Eye-in-hand and eye-to-hand visual servoing

A classification regarding the camera configuration is eye-in-hand versus eye-to-hand. Visual servo control systems typically use one of two camera configurations: mounted on the tool or fixed in the workspace. The first, also referred to as eye-in-hand configuration, has the camera mounted on the tool. Often there exists a known and constant relationship between the pose of the camera and the pose of the tool. The second configuration, the eye-to-hand configuration, has the camera mounted in the workspace. The eye-in-hand configuration has a precise sight of the scene relative to the camera, whereas the eye-to-hand configuration often has a more global sight which might be less precise.

The presented taxonomy is highly concerned with design choices for visual servo controllers where either an *ideal machine* or a *non-ideal machine* is considered. The second issue, regarding the *non-ideal repetitive structure*, will be discussed in more detail in the remainder of this section. Therefore, consider a repetitive structure, where the task is to align the tool to an arbitrary feature, the target feature. As stated earlier, the tendency is that the size of these repetitive structures is increasing, while at the same time the number of features per unit area is increasing, see the example below.

Example: product and feature size

Consider a wafer with a diameter of 200 mm, also referred to as an 8" wafer. The size of a semiconductor diode is approximately $250 \times 250 \mu\text{m}$. The number of diodes on a single row can therefore easily reach as much as 800. The required position accuracy in this type of applications is typically in the order of $10 \mu\text{m}$. Assuming the diodes can be recognized with pixel accuracy, this means that at least 20000 pixels are needed for a single row. In order to have the full wafer in the field of view a vision sensor would be needed of 20000×20000 pixels. With a readout rate of 1 kHz and with typically 255 gray levels (8-bits) per pixel, this would lead to a data stream of 0.4 TB/s.

This simple example illustrates that in many applications it is impossible to have the full product in the field of view and process the data stream in time, while meeting the position accuracy demands. Therefore, for the sake of resolution, the field of view is restricted to only a small part of the repetitive structure, i.e., not the whole repetitive structure is in the field of view. Therefore, the target feature might be outside the field of view. Since the target feature can be outside the field of view and the pitch between consecutive features may vary due to manufacturing tolerances or temperature fluctuations, the metric position of the target feature measured in units of pixels is unknown. As a result a pixel-based reference cannot be prescribed a priori. In order to still use pixel-based references there is a strong need for an adaptive pixel-based reference. This can be done via trajectory generation. Literature concerning online trajectory generation within visual servoing can be found in (Feddemma and Mitchell, 1989; Mezouar and Chaumette, 2000, 2002; Schramm et al., 2005; Schramm and Morel, 2006). However, most of these works are concerned with how to plan a trajectory from an initial pose to its target pose which is known *a priori*. In our case however, the final target position is *unknown*. Therefore, dependent on the *current* measurements a new trajectory should be calculated online and applied to the closed-loop control system, also referred to as online or adaptive trajectory generation (Broquère et al., 2008; Kröger and Wahl, 2010; Kröger et al., 2006; Haschke et al., 2008; Zheng et al., 2009). This approach is schematically depicted in Fig. 1.7. It shows a standard control loop consisting of a plant G and a feedback controller K . The measured output of the plant is given by y . This measurement together with the target t , which typically is the feature number to be processed, is directly used in the online trajectory generator R . The output r of this trajectory generator is a trajectory leading to the target t . The corresponding units of the signals are given in Table 1.2. This approach is present in many applications, like for example the previous mentioned robotic hand trying to catch a ball (Namiki and Ishikawa, 2003b). Another area is the RoboCup (Kitano et al., 2002; RoboCup, 2010) soccer league (Kalmár-Nagy et al., 2004; Sherback et al., 2006; Kalmár-

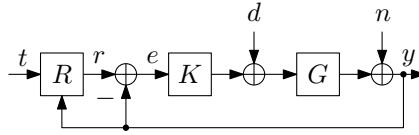


Figure 1.7: Conventional control approach with an online or adaptive trajectory generator R .

Nagy et al., 2002; Purwin and D’Andrea, 2006). Also in the field of automatic milking robots (Honderd et al., 1991; Frost et al., 1993; Wittenberg, 1993) the same strategy is adopted.

In general, the measured output y is corrupted by measurement noise n , see Fig. 1.7. Therefore, it is expected that the resulting trajectory is affected by this noise, which might lead to a poor performance. Furthermore, by introducing a trajectory generator a dual or cascaded control loop is created. For this cascaded system closed-loop stability should be satisfied. Many trajectory generators generate second or higher order position profiles in which for example the maximum acceleration and maximum velocity can be incorporated (Kalmár-Nagy et al., 2002, 2004; Sherback et al., 2006; Purwin and D’Andrea, 2006). Other trajectory generators are based on for example splines (Bazaz and Tondu, 1999), Bezier curves (Hwang et al., 2003) or potential fields (Tsourveloudis et al., 2002). In general, these trajectory generators are highly non-linear systems, such that stability of the closed-loop system is hard to prove. Moreover, guarantees about when to arrive at the target feature is hard to implement, since the distance to be covered is initially unknown.

A different approach to position the tool with respect to non-ideal repetitive structures and which is closely related to this work is given in (Brier et al., 2007). It describes the implementation of a visual position estimation algorithm, using an FPGA in conjunction with a line-scan sensor positioned at an angle over a part of a two-dimensional repetitive structure. A Fourier transform is used with direct interpretation of the phase information at the two fundamental frequencies of the repetitive structure, i.e., phase correlation in the frequency domain (Kuglin and Hines, 1975). A condition needed for this approach is that the two fundamental frequencies do not coincide with each other or with one of their harmonics. The position is determined by the phase of each frequency, which is accumulated every 2π and added to the measured phase. This means that positions are incremented when a feature is passed, while in between features an interpolated position is obtained. This basic idea of incrementing and interpolating will be implemented in this work. A disadvantage however of the approach in (Brier et al., 2007), is that

the phase obtained from the Fourier analysis only provides an *average* position of the features that are *locally* observed. With the knowledge that the repetitive structure is not ideal, the peaks in the amplitude of the Fourier transform suffer from leakage leading to “lobes” instead of sharp “peaks”, which raises the question what the real fundamental frequency is. Moreover, if the orientation of the features deviates the same effect will be present. Therefore, the position of each specific feature is still unknown.

1.4 Research goal and approach

The goal of this work is to design a control approach for positioning a tool with respect to the features of a non-ideal repetitive structure using visual servoing, while a non-ideal machine is considered. Two tasks will be considered in this work, which are

- positioning the tool with respect to arbitrary features of a non-ideal repetitive structure, and
- positioning the tool from one feature to its neighboring feature of a non-ideal repetitive structure.

In our research approach visual servoing will be used to align the tool with respect to the features of the non-ideal repetitive structure. Therefore, in this section first the visual servoing control design choices will be presented according to the taxonomy described in Section 1.3. Although many challenges are present in the field of visual servoing, like the optical design and the applied image processing techniques, the focus of this work is on the control approach. Second, to deal with the problem of unknown metric feature positions, we will discuss the use of position measurements in the novel feature domain, i.e., feature-based positions will be introduced for positioning the tool with respect to arbitrary features, while taking into account the pitch imperfections. Finally, for the special task where the tool is to be positioned from one feature to the neighboring feature a novel feedforward algorithm will be given based on the well known iterative learning control technique, which will deal with the pitch imperfections.

The first category of the visual servoing taxonomy is indirect versus direct visual servoing. As opposed to indirect visual servoing, direct visual servoing makes no assumptions regarding rigid body behavior, perfect velocity tracking control or delay. All these machine imperfections are directly present in the plant to be controlled by the vision controller. Therefore, the direct visual servoing control strategy considers a *non-ideal machine* and will be adopted in this work.

The second category of the taxonomy is image-based visual servoing versus position-

based visual servoing. In this work the features of the repetitive structure will be used for positioning. As previously stated, a *non-ideal repetitive structure* will be considered. This means that due to pitch imperfections there is no geometric model on beforehand of the target, being the repetitive structure, that can be used for position reconstruction or pose estimation. Therefore, in this work we will adopt the image-based visual servoing (IBVS) control strategy. One of the advantages of the IBVS approach is that it may reduce computational delay since no pose estimation is needed in this strategy. Another advantage of image-based control is that there is no need to interpret the image. Keeping the features in the field of view in IBVS is reported to be more easy as opposed to PBVS (Chesi et al., 2004). A last advantage with respect to position-based visual control is that errors due to camera calibration and sensor modeling are eliminated.

The third category of the taxonomy is monocular versus binocular or stereo vision. The distance from the repetitive structure to the camera is assumed to be constant in this work, since planar motion will be considered. Therefore, monocular vision instead of stereo vision will be used throughout the work. Moreover, the use of more cameras is more expensive, more computational demanding, and puts extra constraints on the placement of the cameras.

The fourth category distinguishes endpoint open-loop from endpoint closed-loop visual servoing. Since ECL systems must track the tool as well as the feature, the implementation of an ECL controller often requires a solution of a more demanding vision problem and places field of view constraints on the system that cannot always be satisfied. Moreover, since both the feature and tool must be tracked simultaneously, both should be detected, which requires a more elaborate image processing algorithm, which is likely to be more computational demanding leading to longer image processing times and performance degradation. Therefore, in this work, it is assumed that there is a known kinematic relation between the camera and the tool, such that the problem of positioning a tool with respect to a feature is transformed into positioning the camera with respect to the feature. More specifically, we assume the tool is located at the center of the image of the camera.

The final category of the taxonomy considers eye-in-hand versus eye-to-hand visual servoing. The latter assumption of a known kinematic relation between the camera and the tool is easier to realize with an eye-in-hand camera configuration than the eye-to-hand camera configuration. In the case of the eye-in-hand configuration a stiff connection between the tool and camera is needed which typically are located near to each other, whereas in the eye-to-hand configuration the distance between the camera and the tool is much larger, possibly involving flexible machine elements such that it is harder to assume a known relative position between the camera and the tool. Moreover, the eye-in-hand camera configuration has a precise sight of the scene relative to the camera, whereas the eye-to-hand configuration often has a more global sight which might be less precise. Since in this work we

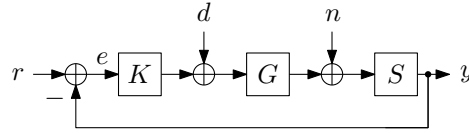


Figure 1.8: Feature-based control approach.

assume the tool is located at the center of the image, we have inherently adopted the eye-in-hand control structure.

The visual servoing control design choices explained above are summarized in Table 1.1.

The control approach shown in Fig. 1.7 represents a tracking problem, where the reference is based on noisy measurements. These references are created by a highly non-linear trajectory generator. The online implementation of a trajectory generator as in Fig. 1.7 leads to a cascaded control architecture, for which stability is difficult to prove due to the non-linear dynamics of the trajectory generator.

In this work we propose a control scheme which is depicted in Fig. 1.8. In this approach the output of the plant enters the block S which generates a so-called *feature-based position* y . This feature-based position takes integer values when the center of the image is perfectly aligned with the features, and interpolates when the center of the image is at a position between features. Therefore, the feature-based position will be expressed in unit of features, denoted by f . The units of the signals of Fig. 1.8 are given in Table 1.2.

A first observation regarding our approach is that the feature-based position of each feature is *known a priori*, whereas the metric or pixel-based position of each feature in Fig. 1.7 is not due to pitch imperfections. The feature-based position of the features are namely the feature numbers or labels. For the one-dimensional case, the first feature takes the feature-based value of 1 f . In between feature one

Table 1.1: Visual servoing choices.

<input checked="" type="checkbox"/> Direct visual servoing	vs.	<input type="checkbox"/> Indirect visual servoing
<input checked="" type="checkbox"/> Image-based visual servoing	vs.	<input type="checkbox"/> Position-based visual servoing
<input checked="" type="checkbox"/> Monocular visual servoing	vs.	<input type="checkbox"/> Binocular visual servoing
<input checked="" type="checkbox"/> Endpoint open-loop	vs.	<input type="checkbox"/> Endpoint closed-loop
<input checked="" type="checkbox"/> Eye-in-hand	vs.	<input type="checkbox"/> Eye-to-hand

and two the feature-based position takes values on the interval $\langle 1, 2 \rangle$ f. Similarly, feature ten has feature-based position 10 f and so on. A second observation between the proposed control approach given in Fig. 1.8 and the one given in Fig. 1.7 is that in our approach a servo problem is to be solved instead of a tracking problem. There is no need to implement a trajectory generator. Third, the feature-based position representation is very intuitive for operators. Aligning the camera with respect to feature $N \in \mathbb{Z}$ is translated into controlling the position to feature-based position N .

The main question of the proposed design approach is how to design the block S . This block maps metric positions to feature-based positions. Different pixel-based pitches between neighboring features all map to a feature-based pitch of one. As a consequence, the block involves a non-linear mapping between pixel-based positions and feature-based positions, leading to a non-linear system from input u to feature-based output y . The non-linear system influences the design of the controller K in terms of robust stability and performance. In this work a stability analysis will be presented in order to guarantee closed-loop stability for a predefined range of metric pitches.

In many applications the motion task is to move from one feature to its neighboring feature of the repetitive structure. In those cases, the next feature is assumed to be already in the field of view, which is not the case if the task is to go to an *arbitrary* feature of the repetitive structure. As a consequence, the pixel-based distance to be traveled can be determined *a priori*, for example via one snapshot. Moreover, in Fig. 1.7 the trajectory r towards the next feature can be generated once without online adaptation. This feature-to-feature task can be seen as a repetitive task. However, the metric distance to be traveled is prone to pitch imperfections, such that a different trajectory is needed every time the system is to be aligned with the next feature. Iterative learning control (ILC) (Moore, 1993) is a well known technique for handling repetitive tasks. In this work the ILC principle is used,

Table 1.2: Signal units.

Signal	Symbol	Unit in Fig. 1.7	Unit in Fig. 1.8
Target	t	[f]	n/a ($t \equiv r$)
Reference	r	[m]	[f]
Error	e	[m]	[f]
Input disturbance	d	[N]	[N]
Measurement noise	n	[m]	[m]
Measured output	y	[m]	[f]

but in our approach setpoints with different magnitudes are applied *during* the learning process such that the tracking performance of different travel distances related to the pitch imperfections are improved iteratively.

1.5 Research contributions

The goal of this work is to design control approaches for positioning the center of the camera with respect to the features of a non-ideal repetitive structure using visual servoing in a non-ideal machine. Two tasks will be considered in this work, which are 1) positioning with respect to arbitrary features of a non-ideal repetitive structure, and 2) positioning from one feature to its neighboring feature of a non-ideal repetitive structure. The contributions of this work are fourfold:

1. The first contribution involves the development of a one-dimensional feature-based position, that is used for positioning with respect to arbitrary features. More specifically, the blocks S and K in Fig. 1.8 will be designed for a one-dimensional repetitive structure with pitch imperfections. Closed-loop stability will be proven for the one-dimensional feature-based control approach.
2. Second, the proposed feature-based control approach for positioning with respect to arbitrary features of the repetitive structure is extended to a full two-dimensional case. Again, pitch imperfections will be considered together with small rotations of the repetitive structure. Different interpolations for obtaining inter-feature positions are considered in this case to improve the performance and simple programmable metric movements with respect to the features will be implemented without having to transform these metric movements into feature-based movements.
3. The third contribution is related to the feature-to-feature task. The ILC principle is used, but in our approach setpoints with different magnitudes are applied *during* the learning process such that the tracking performance of different travel distances related to the pitch imperfections are improved iteratively. Second order ILC will be used in this work to estimate 1) the part of the error that is independent of the magnitude of the setpoint and 2) the part of the error that is directly related to the magnitude of the setpoint.
4. Finally, the proposed control approaches will be validated in practice on an industrial xy-wafer stage.

1.6 Outline of this thesis

This thesis consists of three research chapters. Each chapter is submitted for journal publication and is therefore self contained and can be read independently. Chapter 2 is based on (De Best et al., 2011a) and will present the development of a one-dimensional feature-based position measurement in the novel feature domain, which will be used as feedback signal in the servo control loop. The robustness with respect to pitch imperfections of the repetitive structure will be proven by a stability analysis. The proposed control design will be validated in practice on an academic visual servoing setup.

Chapter 3 is based on (De Best et al., 2011b) and will extend the feature domain to two dimensions. High order interpolation for obtaining inter-feature positions will be considered next to linear interpolation in order to improve the performance. Next to movements from one feature to another we will also discuss the implementation of metric movements with respect to the feature to increase the versatility of programmable movements. An industrial xy-wafer stage will be used in combination with a commercially available off-the-shelf camera to experimentally validate the proposed control approach.

Chapter 4 is based on (De Best et al., 2011c) and will discuss the use of second order iterative learning control for the specific motion task of positioning the tool from one feature to its neighboring feature of the repetitive structure. Different types of disturbances will be considered, identified and compensated where possible. The xy-wafer stage is again used as a testbed for the proposed control design. Finally, in Chapter 5 the main conclusions of this work will be given together with recommendations for future work.

Chapter 2

One-dimensional feature-based motion control

THIS chapter focusses on direct dynamic visual servoing at high sampling rates in machines used for the production of products that consist of equal features placed in a repetitive pattern. The word “direct” means that the system at hand is controlled on the basis of vision only. More specifically, the motor inputs are driven directly by a vision-based controller without the intervention of low level joint controllers. The considered motion task is to position the repetitive structure in order to align the center of the camera with respect to the features. The vision based controller is designed using classical loop shaping techniques. Robustness with respect to imperfections of the repetitiveness is investigated. The combination of fast image processing and a Kalman-filter based predictor results in a 1 kHz visual servoing setup. The design approach is validated on an experimental setup.

2.1 Introduction

Many production processes take place on repetitive structures, for example in ink jet printing technology where droplets are placed in a repetitive pattern, or in pick and place machines used in the production of discrete semiconductors. In each of these processes one or more consecutive steps are carried out on the particular features of the repetitive structure to create the final product. Such production

This chapter is based on: J.J.T.H. de Best, M.J.G. van de Molengraft and M. Steinbuch. High Speed Visual Motion Control Applied to Products with Repetitive Structures. *Accepted for publication in IEEE Trans. Control Syst. Technol.*

machines often consist of a tool, for example a print head, and a stage or carrier on which the repetitive structure is to be processed. Key to obtaining a high product quality is to position the tool with respect to each feature of the repetitive structure with a high accuracy. In current industrial practice, local position sensors such as motor encoders are used to measure the tool position x_t and the position of the stage x_o separately as shown in Fig. 2.1(a). Often the absolute reference points of these measurements do not coincide. This is referred to as an *indirect* measurement of $x_t - x_o$. Using such local measurements in a closed-loop control approach often leads to a collocated control design. The final accuracy of alignment in this case is directly dependent on the following machine properties:

- geometric accuracy of the mechanical construction,
- stiffness of the mechanical construction and
- thermal stability of the machine.

Furthermore, the final accuracy of alignment also relies on assumptions with respect to the repetitive structure:

- infinitely stiff connection between the supporting stage and the repetitive structure,
- constant and known alignment of the repetitive structure with respect to the actuation axes,
- infinite stiffness of the repetitive structure,
- constant and known pitch between successive features of the repetitive structure and
- thermal stability of the repetitive structure.

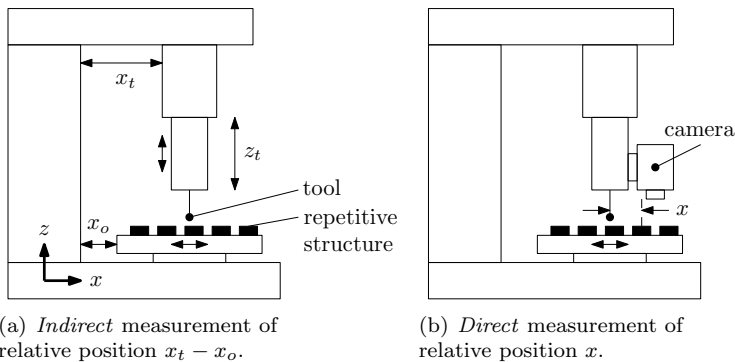
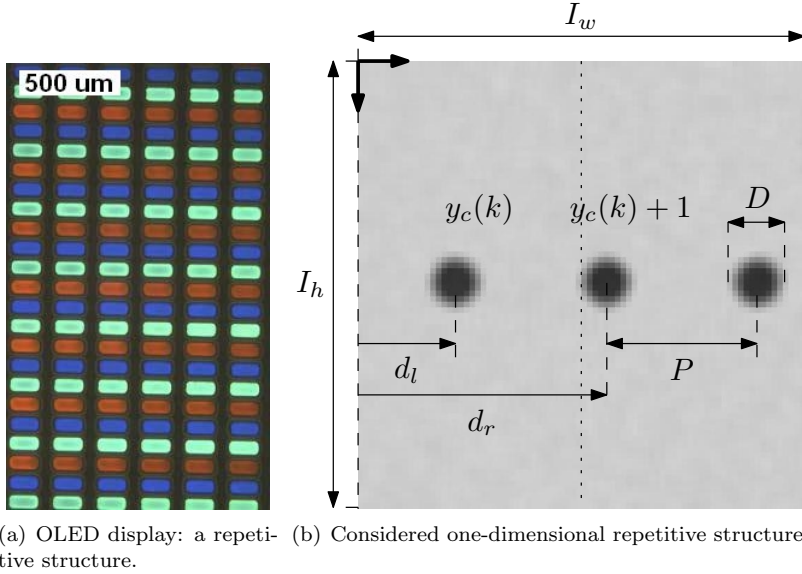


Figure 2.1: Direct versus indirect measurement.

Some of the above issues may result in reproducible errors, especially geometric imperfections and dynamic flexibilities. These errors usually require expensive mechanical measures, with respect to both the machine itself and the repetitive structure. As an example, a priori unknown pitch variations in the repetitive structure will limit the attainable accuracy and prevent the use of absolute motion setpoints in high-accuracy applications. Such limitations due to imperfections can be overcome by adopting a different design paradigm where a camera is used for a direct measurement of the relative position between product and tool. In this paradigm the imperfections of machine and product will be dealt with by non-collocated feedback control. This work exploits the potential of this approach by constructing a feature-based position measurement on the basis of camera images, such that motion setpoints can be defined from feature to feature without knowing the exact absolute position of the features beforehand, while achieving a high positioning accuracy. Controlling a mechanical system by means of camera measurements is referred to as visual servoing (Hashimoto, 2003; Hutchinson et al., 1996; Malis, 2002). Kinematic visual control (Chaumette and Hutchinson, 2006; Hashimoto, 2003; Hutchinson et al., 1996) assumes rigid body dynamic behavior and can not be used in our dynamic, non-collocated control approach. Indirect dynamic visual control (Corke, 1995; Corke and Good, 1992, 1996; Sequeira Goncalves, 2001; Sequeira Goncalves and Caldas Pinto, 2003) does account for dynamic effects but still relies on the presence of collocated position sensors. Therefore, we will adopt the concept of direct dynamic visual control (Ishii et al., 1996; Ishikawa et al., 1992; Nakabo et al., 2000) with eye-in-hand camera configuration, where we assume that the tool is located in the center of the image. The main contributions of this work compared to the above literature are the following:

- feature-based position sensing enabling a direct dynamic visual control paradigm that is robust against machine imperfections and deviations in the pitch between successive features of the repetitive structure,
- stability analysis of the controlled system with respect to the allowable deviations in the pitch between features of the repetitive structure, and
- validation of the proposed methods on a practical direct visual control setup using a commercially available and cost-effective camera with a 1 kHz update rate.

The rest of the chapter is organized as follows. In Section 2.2 the measurement principle to create a feature-based position sensor using the repetitive structure in combination with a camera will be given followed in Section 2.3 by the design of a model-based predictor that is needed when moving at high velocities and for speeding up the image processing steps. The image processing algorithm will be discussed in Section 2.4. The practical setup used for validation of the proposed algorithm will be described in Section 2.5, the system identification in Section 2.6, and Section 2.7 will discuss the final integration. The stability analysis in com-



(a) OLED display: a repetitive structure. (b) Considered one-dimensional repetitive structure.

Figure 2.2: Repetitive structures.

bination with the controller design will be given in Section 2.8, followed by the experimental validation in a closed-loop visual servoing control setting. Finally, conclusions and suggestions for future work will be given.

2.2 Measurement principle

Within this research we focus on machines used for the production of structures that inherently consist of identical features placed in a repetitive pattern such as OLED displays, see Fig. 2.2(a). At this point we restrict the focus of the work to a one-dimensional repetitive structure for ease of explanation. In many manufacturing machines, production steps are carried out row by row or column by column, so in practice we need a two-dimensional position measurement. In our case the second dimension is however restricted by the field of view of the camera. The focus in this work will be on the feature-based position measurement along the repetitive structure in order to apply feature-based control. For now we will consider the features to be circular objects as shown in Fig. 2.2(b), with a diameter of D pixels. The height and width of the image captured by the camera

are I_h and I_w pixels, respectively. The repetitiveness is characterized by the pitch P between the features, which satisfies $\bar{P} - \Delta P \leq P \leq \bar{P} + \Delta P$, where \bar{P} is the nominal pitch and ΔP is the maximum pitch variation. The number of features that are completely within the field of view for the presented method must be at least two, and must be located at *different* sides of the center of the field of view. Therefore, the required field of view is determined by the pitch of the repetitive structure together with the feature size. In the case of a different pitch either the height of the camera can be adjusted which influences the resolution, or a differently sized area of pixels can be read out leading to different acquisition and processing times. Within the image, the horizontal pixel positions d_l and d_r of the two features that are located nearest to the opposite sides of the image center are measured, see Fig. 2.2(b). These features are labeled $y_c(k)$ and $y_c(k) + 1$, with $y_c(k) \in \mathbb{Z}$, *irrespective* of the mutual pixel distance. Here, the time step is indicated by k . The measured position y_v that will be used in the closed-loop visual control setting is now given by

$$y_v(k) = y_c(k) + y_f(k), \quad (2.1)$$

with y_c being the coarse position, i.e., the integer feature label. The fine position y_f is the linear interpolation between the left and right feature label and is calculated as

$$y_f(k) = \frac{0.5I_w - d_l(k)}{d_r(k) - d_l(k)} \leq 1. \quad (2.2)$$

The output $y_v(k)$ indicates the position of the center of the image in feature units. So, $y_v(k) = 1.0$ indicates that the feature labeled 1 is exactly in the center of the image, whereas $y_v(k) = 0.5$ indicates that the center of the image is exactly between the features with labels 0 and 1. Therefore, we define the feature label, denoted by f , as a measurement unit. Pitch variations, i.e., $\bar{P} - \Delta P \leq d_r - d_l \leq \bar{P} + \Delta P$, cause this measurement to become piecewise linear, i.e., the gain of the process varies along the structure. Section 2.8 will discuss this in detail, where the feature unit f also appears.

2.3 Model-based prediction

Key to obtaining the correct position is determining the value of $y_c(k)$ within the field of view. When, for example, the velocity is one pitch per sample the camera will record identical images every time step. Based on that information only, the measurement y_v as described in the previous section gives the same value if y_c is not incremented, i.e., we measure a velocity of zero while the structure is moving with the high velocity of one pitch per sample. If the velocity is increased further aliasing effects cause the features to appear to move slowly in the *wrong* direction.

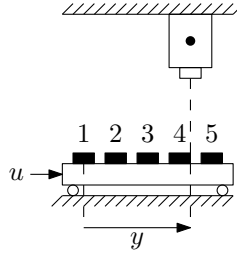


Figure 2.3: Single mass system. The input is denoted by u . The output, denoted by y , is the position of the repetitive structure measured by the camera.

To tackle the problem of incrementing the value of y_c , a model-based solution will be applied. More specifically, we will design a stationary Kalman filter (Kalman, 1960), from which the one step ahead prediction will be used to estimate the value of y_c for the next time step. Moreover, next to incrementing the value of y_c , the one step ahead prediction will also be used to estimate where the features will be located in the field of view in the next time step. Therefore, we will model the input-output behavior of the motion drive carrying the repetitive structure as a mass system, see Fig. 2.3. The input of the system u is the force applied to the mass and the output is the position y . The state space representation of the discrete time system is given by

$$\underline{x}(k+1) = A\underline{x}(k) + B(u(k) + w(k)) \quad (2.3)$$

$$y(k) = C\underline{x}(k), \quad (2.4)$$

where $\underline{x} = (y \dot{y})^T$ is the state vector containing the position y and the velocity \dot{y} , with $\underline{x}(0) = \underline{x}_0$, u is the known applied force and w is the process noise, being the unmodeled forces. The matrices A , B and C are the system, input and output matrices, respectively. The specific matrices for our model are straightforward, and expanded, time-delay versions are given in Section 2.6 by (2.18). In this section a stationary Kalman filter will be given that estimates the output y given the known input u and the measurement y_v given by

$$y_v(k) = C\underline{x}(k) + v(k), \quad (2.5)$$

where v represents the measurement noise. For the process and measurement noise we assume

$$E(Bw^2B^T) = BB^TE(w^2) = Q_w, \quad E(v^2) = Q_v, \quad E(wv) = 0, \quad (2.6)$$

where $E(\cdot)$ is the expected value operator. The Kalman filter consists of a 1) prediction step

$$\hat{\underline{x}}(k+1|k) = A\hat{\underline{x}}(k|k) + Bu(k), \quad (2.7)$$

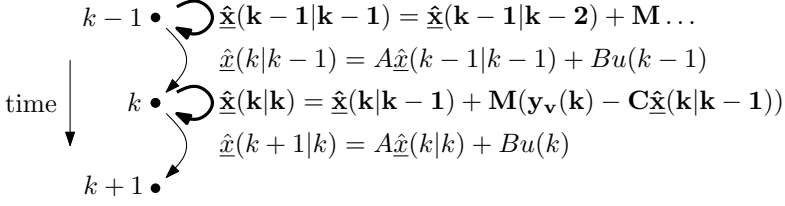


Figure 2.4: The Kalman filter consists of a 1) prediction step (normal) given by (2.7) and a 2) correction step (bold) given by (2.8).

and a 2) 'no steps ahead' correction step

$$\hat{\underline{x}}(k|k) = \hat{\underline{x}}(k|k-1) + M(y_v(k) - C\hat{\underline{x}}(k|k-1)), \quad (2.8)$$

where M is the Kalman gain obtained from solving the steady state Riccati equation. Here, the prediction of the state at time step $k+1$ on the basis of measurements up to time step k is denoted by $\hat{\underline{x}}(k+1|k)$. The two steps are graphically depicted in Fig. 2.4. Combined the prediction and correction step lead to

$$\hat{\underline{x}}(k+1|k) = A(I-MC)\hat{\underline{x}}(k|k-1) + Bu(k) + AMy_v(k). \quad (2.9)$$

The one step ahead output prediction uses this one step ahead state prediction and is given by

$$\hat{y}(k+1|k) = C\hat{\underline{x}}(k+1|k), \quad (2.10)$$

where, $\hat{y}(k+1|k)$ is the estimate of $y(k+1)$ on the basis of measurements up to time step k . This prediction is used to get an estimate \hat{y}_c of the position of the repetitive structure in the next time step $k+1$:

$$\hat{y}_c(k+1|k) = \lfloor \hat{y}(k+1|k) \rfloor, \quad (2.11)$$

where $\lfloor \cdot \rfloor$ is the floor function, which rounds $\hat{y}(k+1|k)$ to the nearest lower integer. In the prediction step explained above it is assumed that the pitch P is constant and equal to the nominal pitch \bar{P} . If this is not satisfied we cannot associate the right label to the feature if $n\Delta P > \bar{P}$, where ΔP is the deviation of the nominal pitch \bar{P} and n is the number of features that has passed within one time step. In this work it is assumed that at every time step a position is measured. In the case the image processing fails to detect the features resulting in an invalid position, the Kalman filter can also be used to predict the position. This is however not considered in this work.

2.4 Fast image processing implementation

Although the focus of this work is on the control approach, this section discusses the image processing algorithm used for detecting the pixel positions d_l and d_r , which in our case comprises straightforward thresholding and calculating the center of gravity. Since the features are assumed to be identical, thresholding and calculating the center of gravity is a low cost primitive image processing technique that indicates a fixed position of the feature, irrespective of its shape or orientation. If however, features are partially occluded, by for example a dust flake, an incorrect position is calculated. In those cases, image registration techniques based on correlation or hough transform could be used which are more computationally demanding. At this point we will introduce search areas around each of the features within the field of view with a width and height of S_w and S_h pixels respectively, such that only one feature is completely present within one search area as shown in Fig. 2.5. In our case we have chosen $S_w = S_h = \bar{P}$. The goal is to search for only one feature within one search area such that labeling implementations to distinguish between multiple features in the image processing algorithms, which cause overhead, can be eliminated. Furthermore, we introduce \hat{d} , which is a pixel position estimate of the feature that is closest to the image center. By using a better prediction the search area can be reduced, which in turn again leads to a smaller computation time of the image processing algorithms. The size of the search area depends on 1) the feature size D 2) the variation of the feature position and 3) the quality of the prediction \hat{d} . Naturally, this size should be larger if 1) the feature size is large, 2) the variation of the feature position is large or 3) the prediction quality is low. The size of the features and the variation of the position are characteristics of the machine which cannot be altered. However, the estimate \hat{d} can be influenced. The pixel position estimation \hat{d} can be obtained from the one step ahead prediction, discussed in the Section 2.3, as follows

$$\hat{d}(k+1|k) = \begin{cases} 0.5I_w + (1 - (\hat{y}(k+1|k) - \hat{y}_c(k+1|k))\bar{P}) & \text{if } \hat{y}(k+1|k) - \hat{y}_c(k+1|k) \geq 0.5 \\ 0.5I_w - (\hat{y}(k+1|k) - \hat{y}_c(k+1|k))\bar{P} & \text{if } \hat{y}(k+1|k) - \hat{y}_c(k+1|k) < 0.5. \end{cases} \quad (2.12)$$

Given this estimate together with the search area, the position of the feature within the search area is calculated. This is done as follows.

First, the image is thresholded within the search area. Global optimal thresholding is performed using Otsu's thresholding method (Gonzalez and Woods, 2008), which determines the optimal threshold level TH . The thresholding is done while reversing salient intensities as follows

$$T(i, j, k) = \begin{cases} TH - I(i, j, k) & \text{if } I(i, j, k) \leq TH \\ 0 & \text{if } I(i, j, k) > TH. \end{cases} \quad (2.13)$$

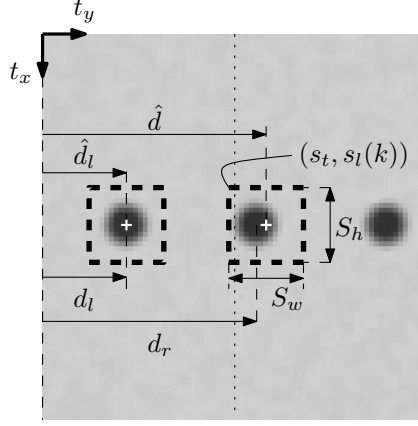


Figure 2.5: Measurements of d_r and d_l using the search areas.

Here, the image data is denoted by $I(i, j, k)$, with indices $i \in \{s_t, \dots, s_t + S_h\}$, $j \in \{s_l(k), \dots, s_l(k) + S_w\}$ indicating the row and column pixel elements, respectively, and k indicating the time step. The position $(s_t, s_l(k))$ indicates the top left corner of the search area, see Fig. 2.5. This position is given by $s_l(k) = \hat{d}(k) - 0.5S_w$ and $s_t = 0.5(I_h - S_h)$. Therefore, we assume that the t_x positions of the features only vary within $S_h - D$ with respect to the center of the image in t_x direction. As a result, we can also measure the t_x position within a limited range. This position can be used in a feedback loop to keep the features within the field of view. However in the remainder we will focus on the horizontal position measurement. The resulting thresholded image is given by $T(i, j, k)$.

Secondly, the center of gravity in the t_y direction within the search area of the thresholded image $T(i, j, k)$ is calculated as

$$d(k) = \frac{\sum_{i=s_t}^{s_t+S_h} \sum_{j=s_l(k)}^{s_l(k)+S_w} iT(i, j, k)}{\sum_{i=s_t}^{s_t+S_h} \sum_{j=s_l(k)}^{s_l(k)+S_w} T(i, j, k)}. \quad (2.14)$$

If $d(k) \geq 0.5I_w$ we have found the center of the feature at the right of the center of the image and we call this distance $d_r(k) = d(k)$. From Fig. 2.5 it can be seen that $d_r(k)$ can be slightly different from $\hat{d}(k)$ indicating the estimation error. Next, if the center of the feature is found at the right of the center of the image, the center

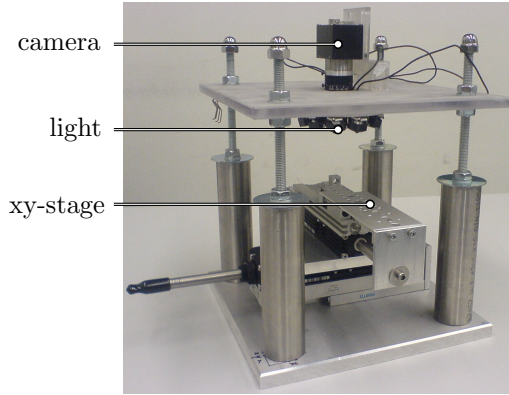


Figure 2.6: Experimental visual servoing setup.

of another feature is searched for at the left of the image center with an estimate given by $\hat{d}_l(k) = d_r(k) - P$. Conversely, if $d(k)$ was found to satisfy $d(k) < 0.5I_w$ we have found the left feature with position $d_l(k) = d(k)$ and we search for the right feature with an estimate given by $\hat{d}_r(k) = d_l(k) + P$. We end up having two positions $d_r(k)$ and $d_l(k)$. These positions are used to determine y_f in (2.2), which together with y_c leads to the feature-based position y_v of (2.1) that will be used for feedback.

2.5 Experimental setup

The setup that will be used later on for experimental validation is depicted in Fig. 2.6. It consists of two stacked linear motors forming an xy-stage. The data-acquisition is realized using an EtherCAT (Jansen and Buttner, 2004) data-acquisition system, where DAC, I/O, and ADC modules are installed, respectively, to drive the current amplifiers of the motors, to enable amplifiers and to measure the position of the xy-stage on the motor side. Hence, this position is only used for comparison and is *not* used in the final control algorithm as such. A Prosilica GC640M high-performance machine vision camera (Prosilica, 2009) with Gigabit Ethernet interface (GigE Vision) which supports jumbo frames and is capable of reaching a frame rate of 197 Hz full frame (near VGA, 659×493) is mounted above the stage. The GigE interface allows for fast frame rates and long cable lengths. The captured images are monochrome images with 8 bit intensity values. To obtain a frame rate of 1 kHz we make use of a region of interest (ROI): we read out only

a part of the image sensor as large as 80×80 pixels. The GigE network controller is able to process frame sizes up to 9200 bytes. The frame size is the number of bytes per packet and the larger the frame size, the less the CPU will be loaded due to the processing of incoming packets. Since 80×80 pixels results in 6400 bytes fitting into a single packet, only one packet is needed. This region is centered at the principal point located near the center of the sensor. The used objective is a Fujinon DF6HA-1 lens, specified with a focal length f of 6 mm. According to the data sheet, the camera has a Micron MT9V403 sensor with a square pixel size p of $9.9 \mu\text{m}$. The camera is calibrated (Heikkila and Silven, 2002) with a calibration grid, which first shows that distortion is negligible due to the small ROI around the principal point. Therefore the pinhole camera model, which maps the metric positions X_i, Y_i, Z_i of the repetitive structure to the image coordinates u_i, v_i , can be applied

$$u_i = \frac{f}{pZ_i} X_i \quad (2.15)$$

$$v_i = \frac{f}{pZ_i} Y_i. \quad (2.16)$$

The focal length f' which is expressed in pixel units, i.e., $f' = \frac{f}{p}$ was calibrated and found to be 623 ± 11 pixels. The height of the camera h was also calibrated and is 0.1132 m. Therefore one pixel in the image coincides with $181.7 \mu\text{m} \pm 3.3 \mu\text{m}$ on the repetitive structure. With a height h of 0.1132 m between the camera and the stage and a focal length f' of 623 pixels we can calculate the resulting field of view as

$$\frac{I_w h}{f'} \times \frac{I_h h}{f'}, \quad (2.17)$$

which in this case is 14.5×14.5 mm. The repetitive structure consists of circular black dots with a diameter of 2 mm and a nominal pitch of 4 mm. The region of interest is therefore large enough such that at least two features are completely in the field of view. The exposure time is set to its minimum, which is $10 \mu\text{s}$. The illumination is realized using power LEDs and set such that all pixel values are within the dynamic range, i.e., between 0 and 255, to avoid clipping. The data-acquisition is integrated in a Linux environment running a 2.6.28.3 preemptible low-latency kernel and the real-time executable is built using the real-time workshop (RTW) of Matlab/Simulink.

2.6 System identification

For the horizontal direction frequency response functions (FRFs) are measured. Two different FRFs are measured: one from the motor input u to the position

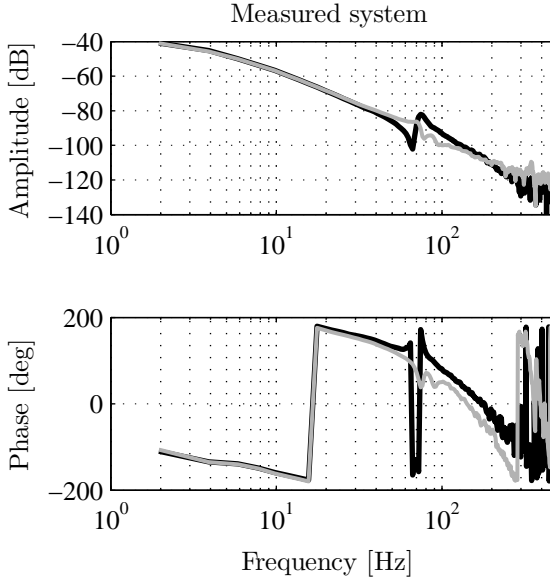


Figure 2.7: Measured FRFs from motor input u to position output y , **black**: using motor encoder, **gray**: using camera (scaled by the pitch of 4 mm for comparison).

output y_{mot} using the motor encoder (collocated control) and one from the motor input u to the position output y_{cam} using the camera with the position measurement as described in the previous sections (non-collocated control). The pitch was constant during the measurement and is 4 mm and the sampling frequency was 1 kHz in both cases. In the ideal case, both y_{mot} and y_{cam} would represent a measurement of the product position y . The result is given in Fig. 2.7, where the camera measurement is scaled by the pitch for comparison. Different dynamics are present if the position measurement from the camera is used instead of the motor encoder. In the case of using the camera as sensor all relevant dynamics are measured including vibrations caused by the limited stiffness of the frame. Furthermore, from the phase plot it can be seen that different time delays are present when using the camera in the feedback loop instead of the motor encoder. The time delay when using the camera is larger, due to the necessary image acquisition and image processing time. When the camera is used the time delay is 3.5 ms, where the readout of the sensor according to the data sheet takes 816 μs , the image transport takes 1.1 ms, the image processing takes 60 μs , the data

acquisition for driving the motors takes 1 ms and finally the zero order hold effect results in 0.5 ms delay.

At 70 Hz a resonance of the system is observed caused by the finite stiffness of the camera mounting. For frequencies below 50 Hz the two FRFs are quite similar and can be modeled by a single mass system as depicted in Fig. 2.3. This model is used in the one step ahead prediction as explained in Section 2.3 . The time delay is incorporated in the discrete time model of (2.3) and (2.4) by adding three additional states $\underline{x} = (y(k-3) \ y(k-2) \ y(k-1) \ y(k) \ \dot{y}(k))^T$ to model a 3 ms delay. The zero-order hold (ZOH) effect introduces another 0.5 ms delay, which in total leads to the 3.5 ms delay present in the system. The system, input and output matrices A , B and C , are given as follows

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0.5T^2/m \\ T/m \end{pmatrix}, \quad (2.18)$$

$$C = (1 \ 0 \ 0 \ 0 \ 0),$$

where m is the dimensionless mass (including all motor and amplifier gains) of the system and $T = 0.001$ s is the sampling time. The value of m is estimated to 0.184. For that the measured FRFs are approximated by a single mass system $G(j\omega) = -\frac{1}{m\omega^2}$. The values of $G(j\omega)$ and ω are known, so m can therefore be estimated for low frequencies.

2.7 Integration

The integration of all blocks within the control scheme is depicted in Fig. 2.8. The steady-state Kalman filter is designed using the matrices A , B and C given in the previous section together with

$$Q_w = BB^T E(w^2) = BB^T q_w, \quad Q_v = E(v^2) = q_r, \quad (2.19)$$

with q_r the variance of the measurement noise v . This variance is determined by measuring the position while the system is at a standstill. The noise of this measurement has a variance of $q_r = 2.8 \cdot 10^{-8}$ f², with f being the unit features as explained in Section 2.2. The value q_w , which is the variance of the unmodeled input w , is used as a tuning variable such that the innovation signal, defined as $y_v(k) - \hat{y}(k|k-1)$, is minimized. The optimal value for the variance of w is found to be 0.07 V. Note that the unit of u and w are given here in volts. The output

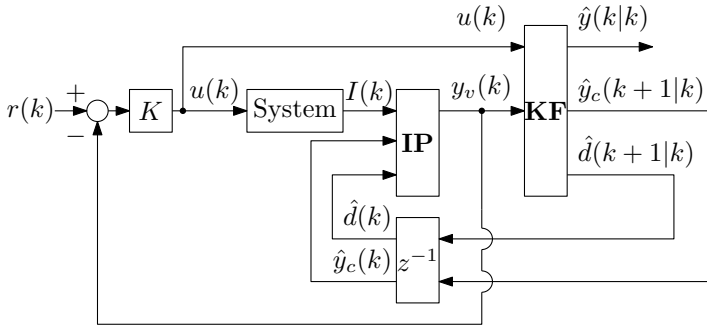


Figure 2.8: Control scheme. The controller K controls the system. The image I captured by the system is processed in the image processing block \mathbf{IP} , resulting in the feature-based position y_v . The Kalman filter \mathbf{KF} is used to estimate *which* features will be in the field of view in the next time sample ($\hat{y}_c(k+1|k)$) and *where* they will be located $\hat{d}(k+1|k)$.

voltage of the controller is applied to a current amplifier which generates a current through the motor leading to a force. In Fig. 2.8 the controller K is connected to the system with input u and image output I . This image is processed in the image processing block \mathbf{IP} using the estimates $\hat{y}_c(k)$ and $\hat{d}(k)$. These estimates are the previous outputs of the Kalman filter \mathbf{KF} . The Kalman filter is used only for 1) incrementing the value of $y_c(k)$ and 2) estimating the position of the feature closest to the image center in the next time step. The filtered position output of the Kalman filter $\hat{y}(k|k)$ is *not* used for feedback since in that case the dynamics of the Kalman filter would attribute to the dynamics to be controlled, i.e., by using $\hat{y}(k|k)$ for feedback, the plant as seen by the controller is the series connection of the system and the Kalman filter, see Fig. 2.8. Therefore, based on the Kalman filter tuning, a different plant is observed by the controller. In our case, we use the output y_v of the system as feedback, such that the dynamics of the Kalman filter do not have to be considered in the controller design.

2.8 Stability analysis

When going from one feature to the next, a different pitch $d_r - d_l \neq \bar{P}$ is considered after each transition in (2.2). This results in a switching gain of the system. In this section the switching behavior will be modeled. Furthermore, a stability analysis will be given that can be carried out to ensure stability of the closed-loop

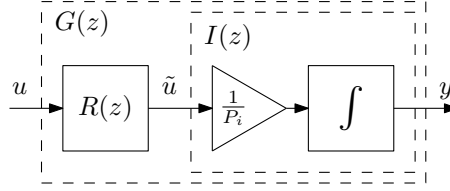


Figure 2.9: Representation of switched system $G(z)$. The input to the system u is mapped to the metric velocity \tilde{u} by the block $R(z)$. In the block $I(z)$, this metric velocity is converted to its feature-based velocity according to the momentary pitch P_i . After integration the feature-based position y is obtained.

system if a single controller is used to control the switching system.

We model the switching behavior of the system $G(z) = R(z)I(z)$ as given in Fig. 2.9. The first system $R(z)$ relates the applied forces to metric velocities measured in m/s. The gain $\frac{1}{P_i}$ converts these metric velocities into feature-based velocities measured in f/s according to the momentary pitch P_i between features. After integration the output of the system $G(z)$ is the position in the feature domain. The system $R(z)$, containing for example the flexibilities of the system, can be written in state space as

$$\underline{x}_R(k+1) = A_R \underline{x}_R(k) + B_R u(k), \quad (2.20a)$$

$$\tilde{u}(k) = C_R \underline{x}_R(k) + D_R u(k), \quad (2.20b)$$

with \underline{x}_R representing the system states for modeling the flexibilities. This system is connected in series with the integrator $I(z)$, that models the rigid body behavior. Therefore, the output $\tilde{u}(k)$ of $R(z)$, being the metric velocity measured in m/s, is an input to the integrator $I(z)$, which can be written in state space as

$$\underline{x}_I(k+1) = A_I \underline{x}_I(k) + B_{I,i} \tilde{u}(k), \quad (2.21a)$$

$$y(k) = C_I \underline{x}_I(k), \quad (2.21b)$$

with $A_I = 1$, $B_{I,i} = \frac{T}{P_i}$ and $C_I = 1$, with T the sampling time of 0.001 s and $P_i \in \mathcal{P}$. The set \mathcal{P} is defined as $\mathcal{P} = \{P_i | \bar{P} - \Delta P \leq P_i \leq \bar{P} + \Delta P\}$, with \bar{P} the nominal pitch. The value ΔP can be related directly to the deviations of the spacing between successive features. By defining the state $\underline{x}_G(k) = (\underline{x}_R(k)^T \ \underline{x}_I(k)^T)^T$ the series connection form the total nominal system $G(z)$ and is given by

$$\underline{x}_G(k+1) = A_{G,i} \underline{x}_G(k) + B_{G,i} u(k), \quad (2.22a)$$

$$y(k) = C_G \underline{x}_G(k), \quad (2.22b)$$

with

$$A_{G,i} = \begin{pmatrix} A_R & 0 \\ B_{I,i}C_R & A_I \end{pmatrix}, B_{G,i} = \begin{pmatrix} B_R \\ B_{I,i}D_R \end{pmatrix}, \quad (2.23)$$

$$C_G = (0 \quad C_I), D_G = 0. \quad (2.24)$$

The above system can switch in time, caused by the non-constant inter spacing between features, i.e., $d_r - d_l \neq P$. The switching results in a different gain of the above system, which is modeled by a switching $B_{I,i}$ matrix. It is chosen to have a switching $B_{I,i}$ matrix since in that case the feature position is continuous, whereas the feature velocity is not. For the switching plant G mentioned above a nominal controller K is designed based on the matrix $B_{I,i}$ containing the value $\frac{T}{P}$. This controller is given in state space representation as

$$\underline{x}_K(k+1) = A_K \underline{x}_K(k) + B_K e(k) \quad (2.25a)$$

$$u(k) = C_K \underline{x}_K(k) + D_K e(k), \quad (2.25b)$$

with \underline{x}_K is the state of the controller. Since the time of switching is assumed to be unknown the question we want to answer is whether the (arbitrary) switching system G controlled by the single controller K is stable given a pitch deviation ΔP . By defining the error $e(k)$ as $e(k) = r(k) - y(k)$ and $\underline{x}(k) = (\underline{x}_G(k)^T \quad \underline{x}_K(k)^T)^T$, substitution leads to the following closed-loop system

$$\underline{x}(k+1) = A_{cl,i} \underline{x}(k) + B_{cl,i} r(k), \quad (2.26a)$$

$$y(k) = C_{cl} \underline{x}(k), \quad (2.26b)$$

with

$$A_{cl,i} = \begin{pmatrix} A_{G,i} - B_{G,i}D_K C_G & B_{G,i}C_K \\ -B_K C_G & A_K \end{pmatrix}, \quad (2.27)$$

$$B_{cl,i} = \begin{pmatrix} B_{G,i}D_K \\ B_K \end{pmatrix}, \quad (2.28)$$

$$C_{cl} = (C_G \quad 0). \quad (2.29)$$

The above system is described by a linear differential inclusion (LDI) (Boyd et al., 1994). For two values of P_i , say P_1 and P_2 , the stability of the system under arbitrary switching can be checked by solving the following linear matrix inequalities (LMIs) (Boyd et al., 1994),

$$E - A_{cl,1}^T E A_{cl,1} \succ 0 \quad (2.30)$$

$$E - A_{cl,2}^T E A_{cl,2} \succ 0 \quad (2.31)$$

$$E \succ 0, \quad (2.32)$$

with the variable E to be solved. However, in this case the stability is only checked for two fixed values of P_i . For any value P_i in between $\bar{P} - \Delta P$ and $\bar{P} + \Delta P$ we proceed as follows. Define $A_{cl,\min}$ containing $P_i = \bar{P} - \Delta P$ as

$$A_{cl,\min} = \begin{pmatrix} A_{G,\min} - B_{G,\min}D_K C_G & B_{G,\min}C_K \\ -B_K C_G & A_K \end{pmatrix}. \quad (2.33)$$

Furthermore define $A_{cl,\max}$ containing $P_i = \bar{P} + \Delta P$ as

$$A_{cl,\max} = \begin{pmatrix} A_{G,\max} - B_{G,\max}D_K C_G & B_{G,\max}C_K \\ -B_K C_G & A_K \end{pmatrix}. \quad (2.34)$$

With these matrices we can write any matrix $A_{cl,i}$ as the convex combination of $A_{cl,\min}$ and $A_{cl,\max}$ given by

$$A_{cl,i} = \alpha_1 A_{cl,\min} + \alpha_2 A_{cl,\max}, \quad (2.35)$$

where $\alpha_1 > 0$, $\alpha_2 > 0$ and $\alpha_1 + \alpha_2 = 1$. Using this definition we can search for a common quadratic Lyapunov function $V(\underline{x}(k)) = \underline{x}^T(k)E\underline{x}(k)$, with $E = E^T \succ 0$ such that $V(\underline{x}(k)) - V(\underline{x}(k+1)) > 0, \forall \underline{x}(k+1) = A_{cl,i}\underline{x}(k)$ or equivalently

$$\underline{x}^T(k)(E - A_{cl,i}^T E A_{cl,i})\underline{x}(k) \succ 0. \quad (2.36)$$

Since the inequality has to hold for every \underline{x} it is sufficient to check

$$E - (\alpha_1 A_{cl,\min} + \alpha_2 A_{cl,\max})^T E (\alpha_1 A_{cl,\min} + \alpha_2 A_{cl,\max}) \succ 0, \quad (2.37)$$

where we substituted (2.35). This can be solved by checking

$$E - A_{cl,\min}^T E A_{cl,\min} \succ 0, \quad (2.38)$$

$$E - A_{cl,\max}^T E A_{cl,\max} \succ 0, \quad (2.39)$$

since α_1 and α_2 are always positive. The proof is given in Appendix A. Therefore, we can conclude that if the systems with $\bar{P} - \Delta P$ and $\bar{P} + \Delta P$ are simultaneously stable, the arbitrary switching system is stable for any value P_i within the bound given by $\bar{P} - \Delta P \leq P_i \leq \bar{P} + \Delta P$.

In our case we investigate the stability for pitch deviations of up to 1 mm with respect to the nominal pitch of 4 mm. The system under consideration is modeled with $I(z)$ defined by (2.21) with

$$A_I = 1, \quad B_{I,i} = \frac{T}{P_i}, \quad C_I = 1, \quad (2.40)$$

and with $R(z)$ defined in (2.20) with

$$A_R = \begin{pmatrix} 5.4 & -3.1 & 2.0 & -0.7 & 0.3 & -0.1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (2.41)$$

$$B_R = (0.1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)^T, \quad (2.42)$$

$$C_R = 1 \cdot 10^{-3} \cdot (0 \ 0 \ 5 \ -8 \ 9 \ 1 \ -18 \ 15 \ -4), \quad (2.43)$$

$$D_R = 0. \quad (2.44)$$

These matrices are determined by fitting a transfer function on the measured frequency response. From the obtained model the integrator I is extracted leading to the system R . This system is converted into a state space model leading to the matrices above. The result of the modeling is given in Fig. 2.10, where for $B_{I,i}$ the value $\frac{T}{P}$ was used. The controller for this system (2.25) consists of a lead filter to create a phase lead with the zero of the lead filter at 7.5 Hz and the pole at 120 Hz. A second order low-pass at 100 Hz with a damping of 0.6 is added such that high frequencies are not amplified. The discrete time controller is given by

$$A_K = \begin{pmatrix} 1.67 & -1.04 & 0.44 \\ 1 & 0 & 0 \\ 0 & 0.50 & 0 \end{pmatrix}, \quad B_K = (64 \ 0 \ 0)^T, \quad (2.45)$$

$$C_K = (26.86 \ -9.09 \ -31.54), \quad D_K = 0. \quad (2.46)$$

The open-loop is given in Fig. 2.11. The achieved closed-loop bandwidth is 20 Hz. Given the matrices, $A_{cl,\min}$ and $A_{cl,\max}$ can be calculated. The LMIs can be solved efficiently using commercially available software (Gahinet et al., 1994). The LMIs were solved and a feasible solution was found, from which it can be concluded that the closed-loop arbitrary switching system is stable. Stated otherwise, the controlled system is robust against pitch deviations within 1 mm.

In practice a number of additional straightforward constraints should be satisfied, which are related to the pitch deviations in combination with the search area width. First, as mentioned before, there always have to be at least two features within the field of view. This constraint can be formulated as

$$2(\bar{P} + \Delta P) + D < I_w. \quad (2.47)$$

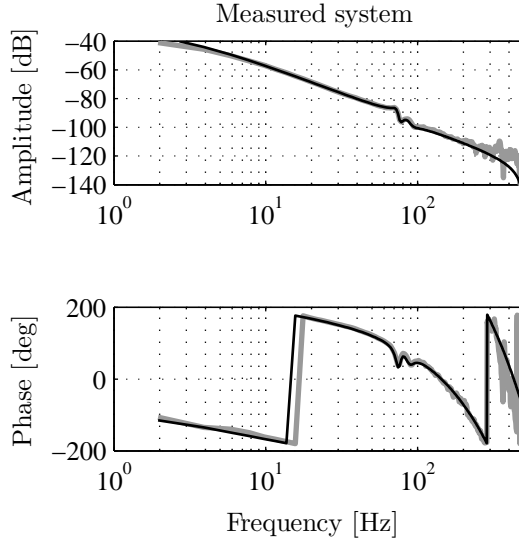


Figure 2.10: Measured FRF from motor input u to position output y using the camera (gray) and the corresponding model (black).

The second obvious constraint that should hold is that features do not overlap

$$\bar{P} > 2\Delta P + D. \quad (2.48)$$

For the width of the search region S_w two more constraints are imposed

$$S_w > D + 2n\Delta P, \quad (2.49)$$

$$S_w < \bar{P} - n\Delta P, \quad (2.50)$$

where n is the number of features that has passed the field of view within one sample. The first constraint states that there is always one complete feature within the search area, whereas the second one states that there may be only one feature within the search region. If the velocity is so high that more than one feature has passed within one sample the number n gets larger than one, hence smaller pitch deviations can be allowed.

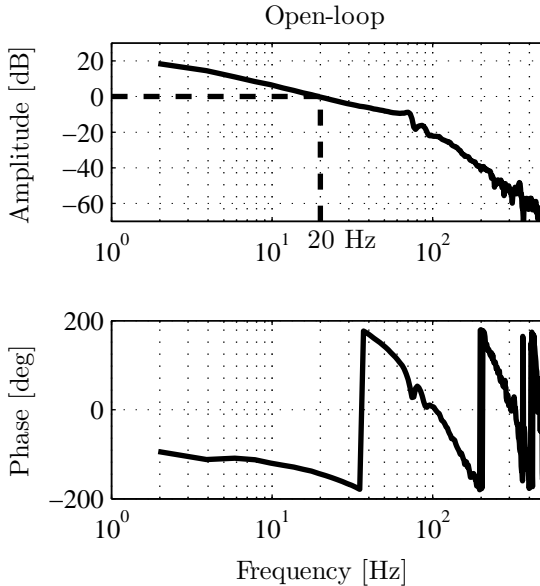


Figure 2.11: Open-loop in y direction.

Remark: The above stability result could also be used to prove stability with respect to lens distortions. This makes the tedious task of camera calibration (Tsai, 1987; Zhang, 2000) superfluous. It is known from literature that image-based visual servoing is more robust against lens distortions than position-based visual servoing (Hutchinson et al., 1996). Our analysis can also be applied for radial lens distortions. When a feature is traveling at a constant velocity, radial lens distortions cause the feature to move at different velocities near the edges of the image than near the center of the image. The velocities can be higher or lower, dependent on whether a pincushion or barrel distortion is present. For that case, we can state that if a feature is near the edge of the image the gain of the system is larger or smaller than the gain near the center of the image. This gain varies during the movement of the feature. Therefore, the same type of analysis can be carried out where the value ΔP depends on the distortion coefficients that describe the radial distortion.

2.9 Results

First, the proposed non-collocated visual servoing solution is compared to the classical collocated solution. Two experiments have been conducted for that, one with collocated feedback using the motor encoder and one with non-collocated feedback using the camera. The non-collocated controller of the previous section is used which has a bandwidth of 20 Hz, see Fig. 2.11. The collocated controller is tuned with a bandwidth of 40 Hz, which is twice as high as in the non-collocated case. In both experiments a repetitive structure is placed on the stage, such that a feature is exactly centered in the field of view of the camera. The control task in both experiments is to simply keep the feature centered in the field of view. The setpoint is therefore zero. During the experiments a disturbance F is applied for one second, see Fig. 2.12. This disturbance might originate for example from other moving parts of the production machine. As a consequence of this disturbance and due to the finite mechanical frame stiffness the camera will move with respect to the feature. Hence, the feature will not be exactly in the center of the field of view any more. Contrary to the collocated case, in the non-collocated case this displacement, denoted by x_c in Fig. 2.12, can be measured and compensated for using feedback control. The results of the two experiments are given in Fig. 2.13. The measured motor encoder x_m is given on the left hand side. The camera measurement x_c is given on the right hand side. The black lines show these measurements under collocated control, whereas the gray lines show the measurements under non-collocated control. Under collocated control, the position error at the motor side is small. However, the performance at the camera

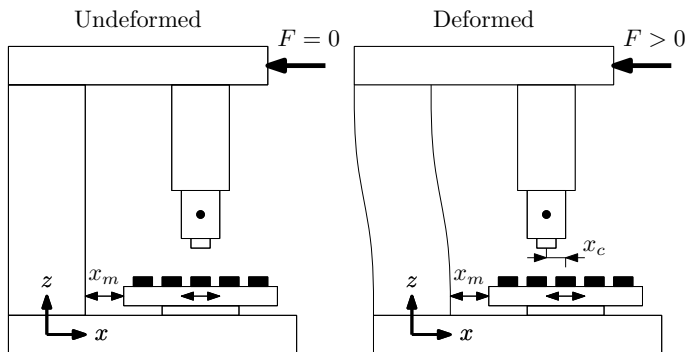


Figure 2.12: The disturbance F results in a relative movement of the camera with respect to the stage, which cannot be observed by the motor encoder. Contrary, this displacement can be observed by the camera.

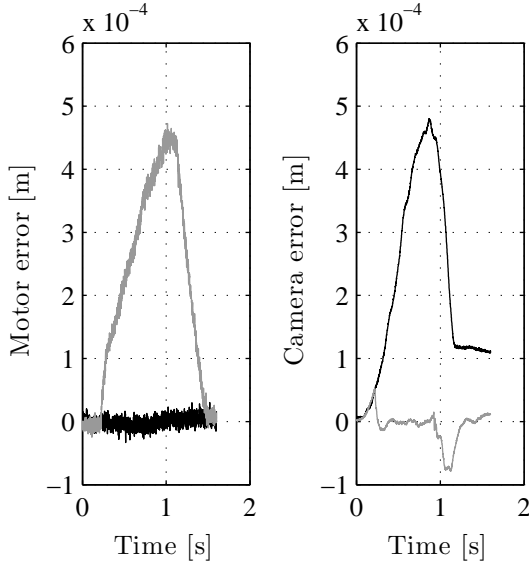


Figure 2.13: Motor error and camera error measured during collocated (**black**) and non-collocated feedback (**gray**).

side, which is our performance variable of interest, shows an error of up to 0.48 mm. There is even a steady state error of 0.1 mm. The performance at the camera side is much better under non-collocated control with maximum error values of 80 μm , which is a factor six better than in the collocated case. Also the steady state error is reduced significantly to less than 10 μm . The cumulative power spectrum (CPS) of the error signals measured by the camera, which are given in Fig. 2.14, also shows a significant error decrease in the case of non-collocated control.

Next, two non-collocated experiments have been carried out with a constant pitch of 4 mm. Throughout the remainder of this section the measurement unit will be features and is denoted by f . In the first experiment a reference with a constant velocity of 25 f/s is applied with the final position at feature 10 f , which is well outside the field of view. In the second experiment the reference to be tracked is a sine wave with an amplitude of 4 f and a frequency of 2 Hz. The outputs $y_v(k)$ are given in the top figures of Fig. 2.15 and Fig. 2.16. During these experiments a one step ahead prediction of the output y_v is calculated as $\hat{y}(k+1|k)$, i.e., using the prediction given in Section 2.3. In the lower figures of Fig. 2.15 and Fig. 2.16

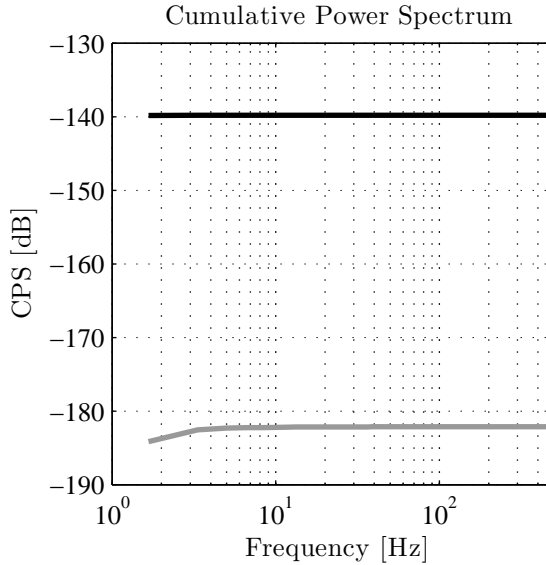


Figure 2.14: Cumulative power spectrum of the error measured by the camera under collocated control (**black**) and under non-collocated control (**gray**) .

this estimate is compared to the measured value of $y_v(k+1)$ and this prediction error is depicted in gray. In Fig. 2.15 we see that during the movement there is an offset in the prediction error. Also a sinusoidal prediction error is present in Fig. 2.16. These errors are probably caused by the friction present in the system. This can be explained by the fact that the innovation signal of Fig. 2.15 is slightly negative. Hence, the prediction is ahead of the actual position, due to the unmodeled friction. This friction is not incorporated in the model of the Kalman filter, therefore leading to a prediction error. Still, in both figures it can be seen that the prediction error is smaller than $0.005 f$. Using the pitch of 4 mm , the metric prediction error can be calculated and is therefore at most $20 \mu\text{m}$. The quality of the position measurement is characterized by a 3σ of $3\sqrt{q_r} = 0.0005 f$ or $2 \mu\text{m}$, with σ the standard deviation.

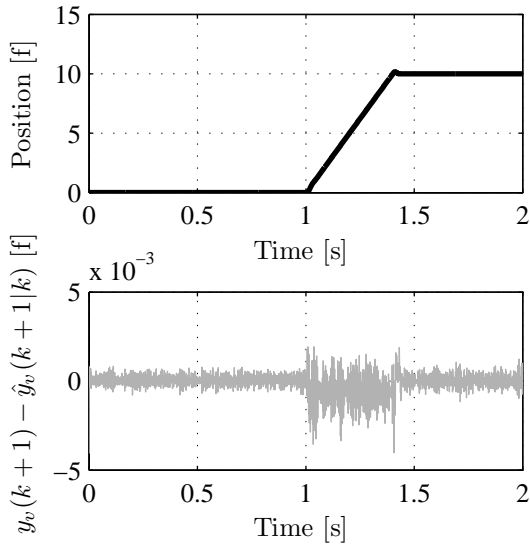


Figure 2.15: Prediction results for ramp reference, **black**: y_v , **gray**: innovation signal $y_v(k+1) - \hat{y}_v(k+1|k)$.

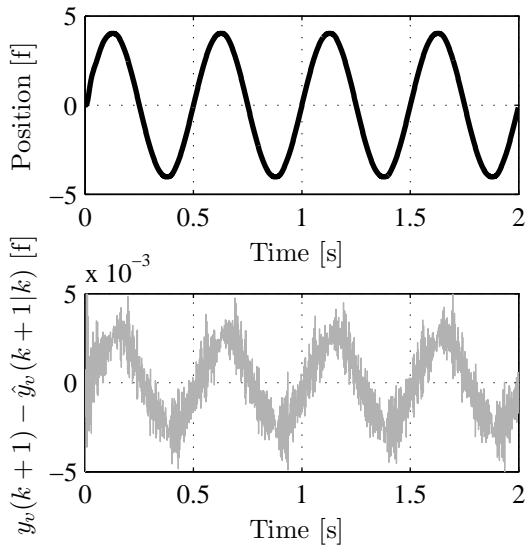


Figure 2.16: Prediction results for sine reference, **black**: y_v , **gray**: innovation signal $y_v(k+1) - \hat{y}_v(k+1|k)$.

Table 2.1: Pitches of non repetitive structure.

Number [-]	Pitch [m]	Measured pitch (mean) [m]
1	0.00376	0.00377
2	0.00359	0.00359
3	0.00470	0.00469
4	0.00377	0.00378
5	0.00366	0.00366
6	0.00436	0.00435
7	0.00411	0.00411
8	0.00347	0.00347
9	0.00388	0.00389
10	0.00472	0.00472

To show the robustness against pitch imperfections of the closed-loop system a structure is placed on the stage with a non-constant pitch. The pitch between successive features varies between 3 mm and 5 mm. The pitches are given in Table 2.1. The experiment with the ramp reference is performed again with this structure. The pitch that was measured during the experiment is given in Fig. 2.17. As can be seen from this figure, the pitches coincide with the ones given in Table 2.1. Furthermore, during the experiment the position error is measured and is given in Fig. 2.18. In this figure the vertical dotted lines indicate when a new feature is at the center of the field of view. Every time a new feature passes the center of the field of view the system switches and obtains a new gain as was explained in the previous section. The switching behavior can be recognized by the small transients in the error after each switching point. When larger pitch imperfections are present, obviously these transients tend to increase leading to a larger position error. However from this figure, we observe that a stable closed-loop system is obtained as was expected.

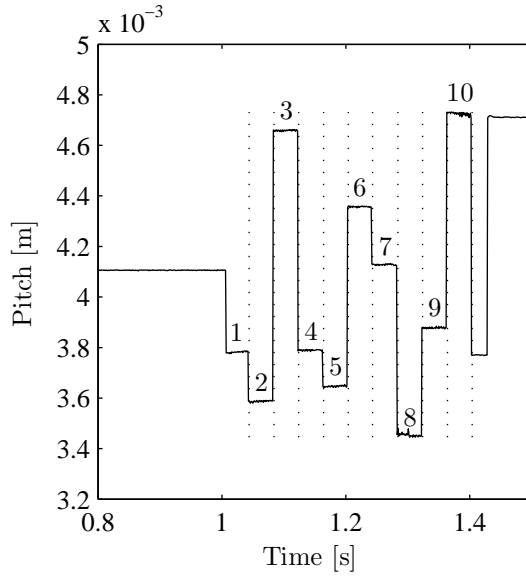


Figure 2.17: Non repetitive pitches.

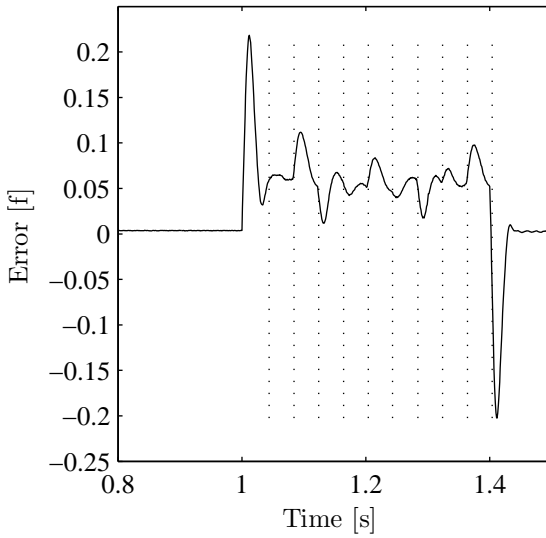


Figure 2.18: Error measured in features.

2.10 Conclusions

In this chapter a direct dynamic visual servoing setup has been created that controls a motion system with 1 kHz visual feedback, without the intervention of low level joint controllers. Different dynamics have been observed when using the motor encoder or measurements of the camera. By using the camera all the relevant dynamics between the camera and the features of the repetitive structure have been measured. The control design accounts for these dynamics. Secondly, an algorithm has been developed that uses the repetitive structure to create a feature-based position measurement with a measurement accuracy of $2 \mu\text{m}$ and capable of sampling at 1 kHz in combination with velocities up to 0.2 m/s. The sampling rate of 1 kHz has been realized by reading out only a part of the vision sensor to reduce the data flow. In the image processing steps a Kalman filter based prediction is used to further reduce the amount of data to be analyzed. The advantage of the proposed method is that feature-based positions can be used instead of metric positions that will drift due to the cumulative sum of the deviations of the pitches. A stability analysis via LMIs and writing the system as a linear differential inclusion shows that the closed-loop system is robustly stable for pitch deviations of the repetitive structure up to 1 mm.

Future work will concentrate on expanding the feature-based position to a full two-dimensional feature-based position including possible structure rotations and developing appropriate feature-based control algorithms. Furthermore, the integration of metric movements of the tool with respect to each feature in combination with feature-to-feature movements based on the presented feature-based control approach will be investigated.

Chapter 3

Two-dimensional feature-based motion control

THIS chapter focuses on the motion control for machines used for the production of products that inherently consist of equal features placed in a repetitive pattern. In many cases the repetitiveness of these structures is prone to imperfections, for example due to thermal expansion, such that the distance between successive features deviates. As a consequence the metric positions of the features of such near-repetitive structures are unknown a priori such that setpoints can not be created a priori. The considered motion task in this chapter is to position a tool relative to the features of a near-repetitive structure with an accuracy of $<10\ \mu\text{m}$. Instead of metric positions we will use novel two-dimensional feature-based positions obtained from a camera capturing images at 1 kHz for feedback, resulting in a direct visual servoing control approach. The robustness with respect to imperfections in the repetitiveness is investigated and the design is validated on an experimental setup.

3.1 Introduction

Many production processes take place on repetitive structures. In each of these processes one or more consecutive steps are carried out on the particular features of the repetitive structure to create the final product. Such production machines often consist of a tool and a stage or carrier on which the repetitive structure is to be processed. The considered control task is therefore to position the tool relative to the features of the repetitive structure. In current industrial practice, local position sensors such as motor encoders are used to measure the position of the tool and the stage separately. Often the absolute reference points of these measurements do not coincide, such that the final accuracy of the alignment of the tool directly relies on properties such as thermal stability, mechanical stiffness and assumptions on the pitch between successive features of the repetitive structure. Any falsification of these assumptions results in a poor alignment.

In (De Best et al., 2009, 2011a) a new control design paradigm was introduced to cope with this problem. Instead of the aforementioned collocated metric position measurements, feature-based position measurements on the basis of camera images in combination with non-collocated visual feedback is used, also referred to as visual servo control (Chaumette and Hutchinson, 2006; Hutchinson et al., 1996). As such, motion setpoints can be defined from feature to feature without knowing the exact absolute metric position of the features beforehand, while still achieving a high positioning accuracy. The proposed method was restricted to the one-dimensional case. In practical applications the repetitive structure in general will contain a two-dimensional grid pattern, like for example the repetitive structure depicted in Fig. 3.1(a), which shows diodes on a wafer. Therefore, in this chapter the feature domain is extended towards two dimensions. Furthermore, in (De Best et al., 2009) the feature-based position is constructed by piecewise linear interpolation between successive features. When passing a feature, a different pitch between the current features is considered. Due to the piecewise linear interpolation the feature-based position is continuous when passing a feature but the feature-based velocity is not and switches instantaneously. As a result, undesired transient responses are observed. In this chapter, higher order interpolation will be implemented to reduce these undesired transient responses. The introduction of feature-based positions results in a straightforward setpoint creation from one feature to another target feature, referred to as *feature-to-feature movements*, without having to know the absolute metric position of the target feature. However, besides these feature-to-feature movements, many production processes require metric movements of the tool *with respect to* the feature, like for example engraving text on each feature. These movements are referred to as *relative feature movements*. Typical movements in such applications are therefore constructed by repeatedly alternating between 1) *feature-to-feature movements* from the current

feature to the target feature and 2) *metric relative feature movements* with respect to the target feature. These relative feature movements will be implemented in the feature-based control approach, so the contributions of this chapter are fourfold: 1) the feature-based position measurement is extended towards two dimensions, 2) the piecewise linear interpolation is extended to higher order interpolation to reduce the transient responses when passing features, 3) next to feature-to-feature movements, relative feature movements are implemented, to increase the versatility of programmable movements and 4) a stability analysis is presented to prove robust stability of the closed-loop system.

The rest of the chapter is organized as follows. In Section 3.2 the notation with respect to the repetitive structure and the different coordinate representations will be presented. Section 3.3 will first introduce two-dimensional feature-based positions, followed by higher order feature interpolation. At the end of Section 3.3 the implementation of relative feature movements will be discussed. The experimental setup that will be used for validation will be given in Section 3.5. The control design and stability analysis will be given in Section 3.6. Finally, conclusions will be given.

3.2 Notation

Throughout this chapter we will use a repetitive structure that consists of equal features ordered in a near-rectangular repetitive pattern. A practical example is depicted in Fig. 3.1(a) which shows diodes on a wafer. A schematic representation of such a repetitive structure is given in Fig. 3.1(b) where the features are circular black dots on a white background. The image captured by the camera, denoted as I , has a height I_h and width I_w pixels and captures only a part of the repetitive structure. The features have a diameter of D pixels and are placed in a rectangular repetitive pattern. The nominal pitch between features is \bar{P} pixels both in horizontal and vertical direction. In this work pitch imperfections will be considered, which can occur for example due to inaccurate preceding process steps, local stretching of the structure when flexible plastic or metal foil is used as product carrier or thermal expansion of the structure. The pitch imperfection is denoted by ΔP , with $0 < \Delta P \ll \bar{P}$, such that each pitch P satisfies

$$\bar{P} - \Delta P \leq P \leq \bar{P} + \Delta P. \quad (3.1)$$

More specifically, once a feature is found, the other features are expected in the shaded areas in Fig. 3.1(b). The position and size of the shaded area are related to the nominal pitch \bar{P} and the pitch imperfection ΔP . We will allow and take into account these pitch imperfections such that precision requirements of the repetitiveness of the pattern can be less strict, while at the same time being able

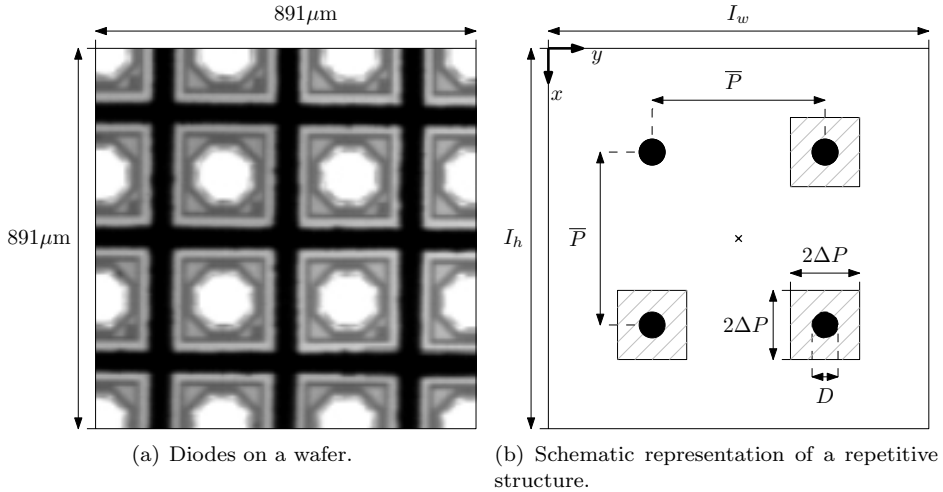


Figure 3.1: A part of a two-dimensional repetitive structure.

to position accurately with respect to the features.

Throughout this chapter two different coordinate representations will be used, which are metric pixel coordinates and feature coordinates. A point p within the image that is expressed in pixel coordinates is represented as \underline{p}^p . Similarly, \underline{p}^f represents the point p in feature coordinates. In the remainder of this chapter subscripts given in Table 3.1 will be used. Using this notation we denote the pixel coordinates of the center of the top left feature for example as $\underline{p}_{tl}^p = (x_{tl}^p \ y_{tl}^p)^T$. Similar notations are used for the top right feature \underline{p}_{tr}^p , the bottom left feature \underline{p}_{bl}^p and the bottom right feature \underline{p}_{br}^p . The point of interest, which initially is taken as the center of the image sensor, is denoted by p_i .

Table 3.1: Subscript definitions.

Subscript	Definition
l	left
r	right
t	top
b	bottom
i	interest

3.3 Feature-based positions

This section will first introduce the feature domain. Next, the detailed steps in obtaining feature-based positions in the feature domain will be discussed, which are the feature detection, bilinear feature interpolation and the high order feature interpolation. Finally, relative feature movements will be presented by combining the metric domain with the feature domain.

3.3.1 The feature domain

Since we assume the tool is located at the center of the image, the control task is to position the center of the image with respect to the target feature. The target feature however might be outside the field of view, since for resolution purposes only a small part of the repetitive structure is observed. Due to a priori unknown pitch imperfections, the absolute position of the target feature in the pixel or *metric domain* is therefore not known on beforehand. Hence, metric setpoint creation cannot be done offline. To solve this problem, we introduce position measurements in the *feature domain*. The advantage of feature-based positions over metric positions is that the positions of each feature in the feature domain are known *a priori*, such that there is no need for online trajectory generation. Therefore, instead of using metric position measurements, novel feature-based position measurements will be used for feedback, see Fig. 3.2. The system captures images I of the scene, which together with the pixel coordinates of the point of interest \underline{p}_i^p are fed into the image processing block **IP** that gives a two-dimensional feature-based position \underline{p}_i^f . At this point, we take the point of interest as the center of the image $\underline{p}_i^p = (\frac{I_h}{2} \quad \frac{I_w}{2})^T$, since this point is to be positioned with respect to the features. The feature-based position \underline{p}_i^f is compared to the feature-based reference \underline{r}_i^f and fed to a controller K that generates the input \underline{u} to the system.

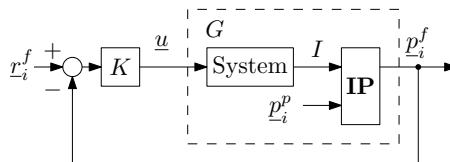


Figure 3.2: Feature-based control approach.

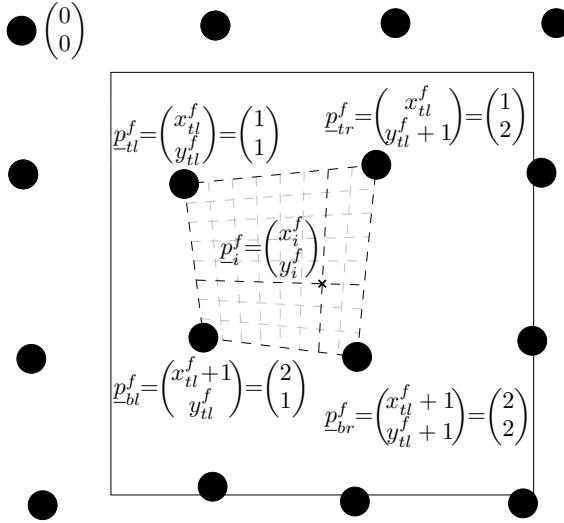


Figure 3.3: Feature-based positions. The frame indicates the field of view.

The two-dimensional measurement principle of the feature-based positions will be explained using Fig. 3.3. The figure shows a part of a near-repetitive structure that is captured by the camera. The pitches between neighboring features are such that they satisfy (3.1) in both directions. For the proposed measurement method it is required that four features will be completely within the field of view, such that they enclose the point of interest p_i indicated by the cross, see Fig. 3.3. Each feature is assigned an integer feature-based position, *irrespective* of their mutual pixel distance, i.e., feature-based positions are determined by counting features starting from the top left corner of the image. So, $\underline{p}_{tl}^f = (x_{tl}^f \ y_{tl}^f)^T \in \mathbb{Z}^2$ is the feature-based position of the top left feature which in this example is $(1 \ 1)^T$. The feature-based positions of successive features can be obtained by simply incrementing the values of x_{tl}^f and/or y_{tl}^f .

Since the goal is to position the center of the image with respect to the target feature, we want to express the inter-feature position of the center of the image, here denoted by the point of interest p_i , in the feature-based coordinates. More specifically, we want to find $\underline{p}_i^f = (x_i^f \ y_i^f)^T \in \mathbb{R}^2$. To obtain a unique feature-based position we required the positions x_i^f and y_i^f increase monotonic between

$$x_{tl}^f \leq x_i^f \leq x_{tl}^f + 1, \quad (3.2)$$

$$y_{tl}^f \leq y_i^f \leq y_{tl}^f + 1. \quad (3.3)$$

Therefore, the idea of (De Best et al., 2009) is extended towards two dimensions as follows. If the point of interest is perfectly aligned with one of the features, its position is the integer two-dimensional feature-based position. In between features we use an interpolation as is indicated by the gray dashed grid lines in Fig. 3.3. Here, the horizontal and vertical grid lines indicate lines of equal x and y feature-based positions, respectively. More specifically, they connect equal fractions of the horizontal and vertical lines between the features. The feature-based position of the point of interest in Fig. 3.3 is given by $\underline{p}_i^f = \left(1\frac{5}{8} \quad 1\frac{3}{4}\right)^T$.

To obtain the correct feature-based position it is essential to determine *which* features are in the field of view. A problem arises when the repetitive structure moves with a velocity larger than one pitch per sample. In that case it is desired to track which features are in the field of view. A steady state Kalman filter (Kalman, 1960) will be used for that, from which *only* a one step ahead prediction will be used to predict 1) *which* features will be in the field of view and 2) *where* these features will be located in the field of view. The first point assures that the feature-based position is incremented when new features enter the field of view, whereas the second point will generate the initial pixel position estimate $\hat{\underline{p}}^p$ of the feature closest to the point of interest, which is used in the feature detection, see Section 3.3.2.

In the next sections the consecutive steps needed for obtaining the feature-based position will be explained in more detail, which involve 1) feature detection, 2) bilinear feature interpolation and 3) higher order feature interpolation.

3.3.2 Feature detection

This section will describe the first step towards obtaining a feature-based position, which is the accurate detection of the pixel coordinates of the four features that enclose the point of interest. These coordinates will be used in the second step as described in the next section, which involves the bilinear feature interpolation.

Initially, assume that a pixel position estimate $\hat{\underline{p}}^p = \left(\hat{x}^p \quad \hat{y}^p\right)^T$ generated by the Kalman filter is available of the feature that is expected to be closest to the point of interest. In Fig. 3.4 the bottom right feature is closest to the point of interest. The pixel position estimate of this feature is given by the gray cross.

A rectangular search area is defined around the estimate $\hat{\underline{p}}^p$, with a width of S_w pixels and a height of S_h pixels. The search area should be such that it completely confines a single feature. The size of the search area is directly dependent on 1) the feature size, 2) the feature position variation (pitch imperfections) and 3) the quality of the estimate $\hat{\underline{p}}^p$. With the introduction of the search area it is possible to search for a single feature within the search area, such that labeling implementations to distinguish between multiple features, which cause a computational overhead, can be eliminated in the image processing steps. Stated otherwise, we

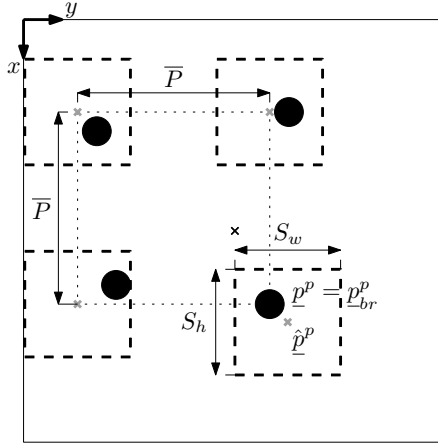


Figure 3.4: Feature detection. Around the initial pixel position estimate \hat{p}^p a search area is defined in which the feature is detected and its position p^p is measured. The other features can be found similarly.

can speed up the image processing by using a priori knowledge about the repetitiveness of the structure.

In our case, we determine the pixel position of the center of the feature p_p by first thresholding the search region followed by a center of gravity calculation (Van Assen et al., 2002). For more complex feature shapes more elaborated image processing techniques can be applied such as hough transforms or template matching. From Fig. 3.4 it can be seen that the measured p^p can be different from \hat{p}^p indicating the estimation error. In this case the bottom right feature is found, i.e., $p^p = p_{br}^p$, since it is located at the bottom right of the point of interest. The pixel position of the three remaining features can be estimated as follows

$$\hat{p}_{bl}^p = p_{br}^p - (0 \quad \bar{P})^T, \quad (3.4)$$

$$\hat{p}_{tr}^p = p_{br}^p - (\bar{P} \quad 0)^T, \quad (3.5)$$

$$\hat{p}_{tl}^p = p_{br}^p - (\bar{P} \quad \bar{P})^T. \quad (3.6)$$

Similar to finding the pixel coordinates of the feature p_{br}^p , search areas can be defined around these estimates and the pixel coordinates of the remaining features p_{bl}^p , p_{tr}^p and p_{tl}^p can be calculated. During these operations it is important to check

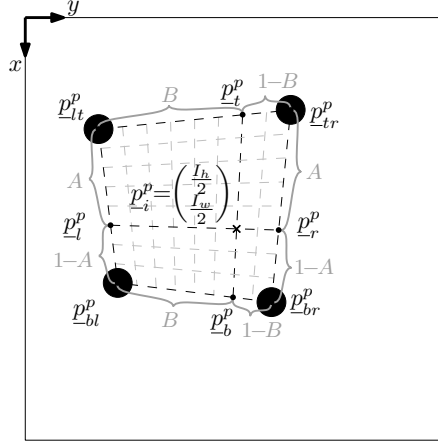


Figure 3.5: Bilinear feature interpolation.

that the four features are *enclosing* the point of interest, i.e.,

$$\underline{p}_i^p \in \left\{ \sum_{l \in L} \alpha_l \underline{p}_l^p \mid \forall \alpha_l \in \mathbb{R}, \alpha_l \geq 0, \sum_l \alpha_l = 1 \right\}, \quad (3.7)$$

with $L = \{tl, tr, bl, br\}$. At the end of this step the pixel positions \underline{p}_{tl}^p , \underline{p}_{tr}^p , \underline{p}_{bl}^p and \underline{p}_{br}^p are known. These coordinates will be used in the second step, the bilinear feature interpolation.

3.3.3 Bilinear feature interpolation

This section will describe the second step for obtaining the feature-based position, which is the bilinear feature interpolation. The bilinear feature interpolation will use the detected pixel coordinates of the four features that enclose the point of interest as explained in the previous section. The bilinear feature interpolation will be extended in the next section leading to second order feature interpolation. The bilinear feature interpolation is based on the similar idea as in (De Best et al., 2009) and will be explained using Fig. 3.3 and 3.5. In Fig. 3.5 two lines intersect the point of interest. The vertical line connects the point $\underline{p}_i^p = (x_i^p \ y_i^p)^T$ with the point $\underline{p}_b^p = (x_b^p \ y_b^p)^T$, whereas the horizontal line connects the point $\underline{p}_i^p = (x_i^p \ y_i^p)^T$ with the point $\underline{p}_l^p = (x_l^p \ y_l^p)^T$. This horizontal line is constructed

such that

$$A = \frac{x_l^p - x_{tl}^p}{x_{bl}^p - x_{tl}^p} = \frac{y_l^p - y_{tl}^p}{y_{bl}^p - y_{tl}^p} = \frac{x_r^p - x_{tr}^p}{x_{br}^p - x_{tr}^p} = \frac{y_r^p - y_{tr}^p}{y_{br}^p - y_{tr}^p}. \quad (3.8)$$

In the case of Fig. 3.5, the value of A is $\frac{5}{8}$. For the vertical line the same reasoning holds. It connects equal B fractions of the top and bottom line, so

$$B = \frac{x_t^p - x_{tl}^p}{x_{tr}^p - x_{tl}^p} = \frac{y_t^p - y_{tl}^p}{y_{tr}^p - y_{tl}^p} = \frac{x_b^p - x_{bl}^p}{x_{br}^p - x_{bl}^p} = \frac{y_b^p - y_{bl}^p}{y_{br}^p - y_{bl}^p}. \quad (3.9)$$

The value for B in Fig. 3.5 is $\frac{3}{4}$. Moreover, it can be shown (see Appendix B) that

$$A = \frac{x_t^p - x_{tl}^p}{x_b^p - x_{tl}^p} = \frac{y_t^p - y_{tl}^p}{y_b^p - y_{tl}^p}, \quad B = \frac{x_i^p - x_{tl}^p}{x_r^p - x_{tl}^p} = \frac{y_i^p - y_{tl}^p}{y_r^p - y_{tl}^p}, \quad (3.10)$$

such that the values of A and B can be expressed as a function of the four pixel coordinates of the enclosing features $\underline{p}_{tl}^p, \underline{p}_{tr}^p, \underline{p}_{bl}^p, \underline{p}_{br}^p$ and the pixel coordinates of the point of interest \underline{p}_i^p . The analytic expressions for A and B are not given here due to space limitations. The two-dimensional feature-based position can now be written as

$$x_i^f = x_{tl}^f + A, \quad y_i^f = y_{tl}^f + B, \quad (3.11)$$

with $A \in \mathbb{R}, 0 \leq A \leq 1, B \in \mathbb{R}$ and $0 \leq B \leq 1$.

3.3.4 Second order feature interpolation

In this section the third step in obtaining the feature-based position measurement will be explained which involves a second order interpolation. When positioning the camera from one feature to another, the point of interest will leave the current area spanned by the four features and enter the next area spanned by four different features, which we refer to as *feature frame transitions* in the remainder of this work. Due to pitch imperfections the mutual distances between the features can be different after feature frame transitions. Using the bilinear interpolation of (3.11) the feature-based position is continuous during feature frame transitions, however the feature-based velocity is not and switches instantaneously. These switching velocities cannot be tracked by the controller and will result in transient position responses whenever feature frame transitions occur. These responses are comparable to the responses obtained when first order trajectories are applied to a closed-loop system. In the remainder of this section, the switching behavior will be explained in more detail and second order functions will be incorporated in the feature interpolation such as to prevent the feature-based velocity from switching

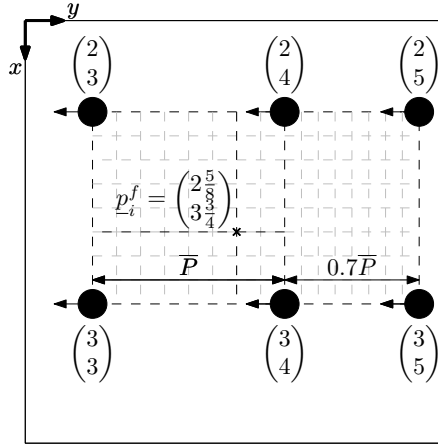


Figure 3.6: The features in the image are moving to the left with a velocity of \bar{P} pixels/s. The feature-based velocity switches when the $y_i^f \geq 4$. This can be seen in the image by the interpolated grid lines that are closer to each other, when $y_i^f \geq 4$.

and therefore to reduce the undesired transient position responses.

To explain the switching feature-based velocity, Fig. 3.6 will be used, where the current feature-based position is $\underline{p}_i^f = (2\frac{5}{8} \quad 3\frac{3}{4})^T$. Suppose the repetitive structure is moving with a constant pixel velocity $\underline{v}^p = (0 \quad -\bar{P})^T$ pixels/s. As a result the feature-based position in the y direction is increasing. If $y_i^f < 4$, the pitch in y direction is $P = \bar{P}$ and therefore the feature-based velocity in \dot{y}_i^f is 1 f/s. However if $y_i^f \geq 4$, the pitch is $P = 0.7\bar{P}$ such that the feature-based velocity instantaneously becomes $\frac{1}{0.7}$ f/s. By introducing a different feature-based position measurement as

$$x_i^f = x_{il}^f + g(A), \quad y_i^f = y_{il}^f + h(B), \quad (3.12)$$

we can design the functions g and h such that the feature-based velocity does not switch but changes smoothly. In the remainder we will focus on the design of the function h , while the design of g can be done in a similar way. The following constraints can be constructed for designing the function h

$$h(0) = 0, \quad h(1) = 1, \quad \frac{dh(0)}{dB} = \frac{P}{\bar{P}}, \quad \frac{dh(1)}{dB} = \frac{P}{\bar{P}}, \quad (3.13)$$

where P is the momentary pitch, i.e., the length of the horizontal grid line intersecting the point of interest, which is \bar{P} if $y_i^f < 4$ and $0.7\bar{P}$ in the case $y_i^f \geq 4$. The first two constraints imply that the feature-based position is continuous across feature frame transitions and is composed of two integer values when the point of interest p_i is aligned with a feature. The second two constraints imply that the feature-based velocity is constant, i.e., does *not* switch, across the feature frame transitions. Fig. 3.7 shows an example of the function $h(B)$ for the cases where $P = \bar{P}$ and $P = 0.7\bar{P}$. It can be seen that if $P = \bar{P}$ the function is simply

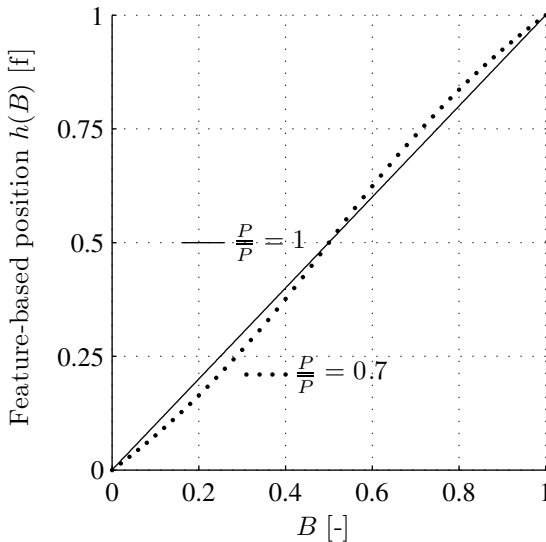


Figure 3.7: Second order interpolation.

$h(B) = B$. If $P = 0.7\bar{P}$ the function $h(B)$ satisfies the aforementioned conditions. In this case the function $h(B)$ is a piecewise quadratic function (see Appendix C, i.e.,

$$h(B) = \begin{cases} \left(2 - \frac{2P}{\bar{P}}\right) B^2 + \frac{P}{\bar{P}} B & \text{if } B < 0.5, \\ \left(-2 + \frac{2P}{\bar{P}}\right) B^2 + \left(4 - \frac{3P}{\bar{P}}\right) B - 1 + \frac{P}{\bar{P}} & \text{if } B \geq 0.5. \end{cases} \quad (3.14)$$

In the choice of the function $h(B)$ a trade-off is made. The function $h(B)$ in this case is chosen to have a non switching velocity when features are passed, so as to reduce the transient response caused by this switching. If for example higher order functions would be designed, even the feature-based acceleration and jerk can be

made continuous. However, one should take into account that this interpolation has a high influence on the gain of the system, i.e., larger values of $\frac{dh(B)}{dB}$ lead to a higher momentary gain of the system and vice versa. As a result the control design needs to cope with high gain variation to prevent potential stability problems as will be explained in Section 3.6.

3.4 Relative feature movements

In the production of repetitive structures, the tool is typically moved from one feature to the next feature, where at every feature a processing step is executed. This processing step can for example be a pick and place action or jetting droplets using ink jet printing technology. For these processing steps it is sufficient to move from one feature to the next, i.e., no additional movements of the tool with respect to the feature have to be carried out. In processing steps, like for example engraving or cutting, additional movements of the tool are necessary before going to the next feature. This section will discuss how, next to feature-to-feature movements, these so-called relative feature movements can be incorporated in the feature-based control design approach.

From an operators point of view it would be preferable to design 1) a reference $\underline{r}_i^f(t)$ for performing feature-to-feature movements expressed in the feature domain and 2) a reference \underline{r}_i^p for performing additional movements of the tool with respect to the feature expressed in the metric domain. Feature-based position measurements were introduced in order to handle feature-to-feature movements. Previously, the point of interest was taken static as the image center, $\underline{p}_i^p = \left(\frac{I_h}{2} \quad \frac{I_w}{2}\right)^T$. By prescribing the metric position of the point of interest in time however, i.e., $\underline{r}_i^p(t) = \underline{p}_i^p(t)$, we can induce relative feature movements. Hence, with the tool still assumed to be in the image center, the relative movement of the feature with respect to the tool is obtained.

An example of such a setpoint is given in Fig. 3.8. During the first 0.2 seconds a diagonal feature-based movement is performed from feature zero to feature one. After arriving at this feature, the relative feature movement will be carried out from $t = 0.2$ s to $t = 1$ s. When this movement is completed, the next feature-to-feature movement will be carried out. By repeating this sequence, the tool is moved along the features and each feature is processed.

The final control scheme is now given in Fig. 3.9. The feature-based reference \underline{r}_i^f is applied to the closed-loop system. The controller K is connected to the system. The input of the system \underline{u} are the applied forces, whereas the output of the system are the captured images I . These images together with the metric reference \underline{r}_i^p for relative feature movements are processed in the image processing block **IP** which performs the feature detection and the feature interpolation. The output of this block gives the feature-based position \underline{p}_i^f that is used for feedback.

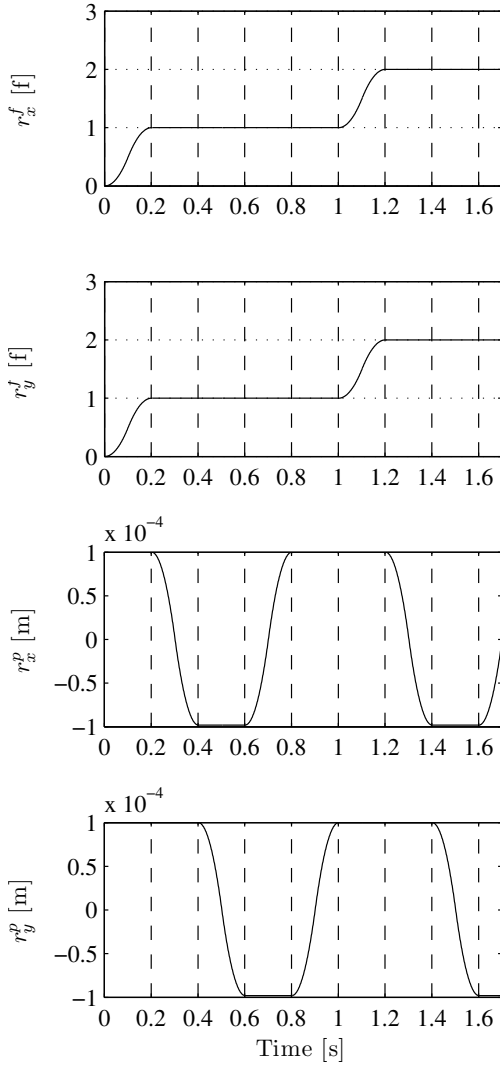


Figure 3.8: Reference \underline{r}_i consisting from $\underline{r}_i^f = (r_x^f \ r_y^f)^T$ for moving from feature to feature and $\underline{r}_i^p = (r_x^p \ r_y^p)^T$ for relative feature movements.

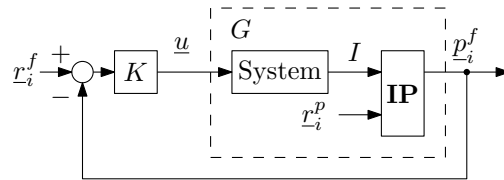


Figure 3.9: Control scheme.

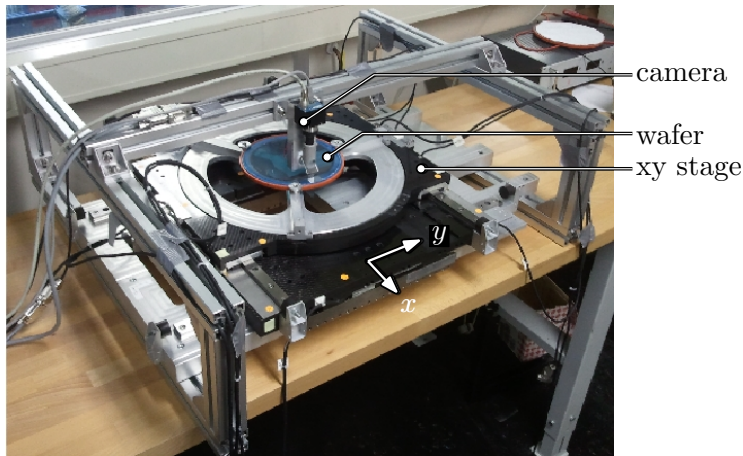


Figure 3.10: Industrial application: an xy-wafer stage.

3.5 Experimental setup

The considered industrial application is an xy-wafer stage and is depicted in Fig. 3.10. On the stage a wafer is clamped which contains the small ($250 \times 250 \mu\text{m}$) discrete semiconductor products, the so-called dies. A frame is mounted above the stage which supports a Prosilica GC640M high-performance machine vision camera (Prosilica, 2009) with Gigabit Ethernet interface (GigE Vision). The camera generates 8 bits monochrome images and is capable of reaching a frame rate of 197 Hz full frame (near VGA, 659×493). To increase the frame rate of the camera to 1 kHz and to reduce the amount of data transport from the camera only a part of the sensor is read out as large as 90×90 pixels. The camera supports jumbo frames of up to 9200 bytes, such that the entire image fits into a single packet. This reduces the CPU load due to less incoming data packets. The magnification of the MC1.00X lens (Opto Engineering, 2010) is one. The pixel size of the camera

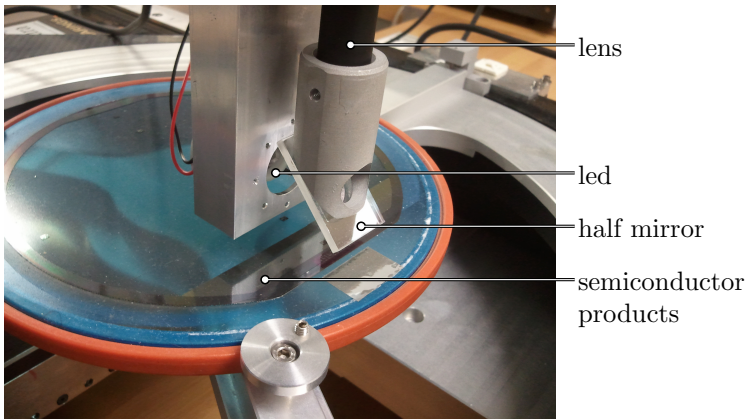


Figure 3.11: Close-up of the xy-wafer stage.

is $9.9 \mu\text{m}$ such that the field of view is approximately $0.9 \times 0.9 \text{ mm}$ which is large enough to view nine dies in a three by three formation. To further reduce the delay and to minimize image blur the exposure time is set as small as $60 \mu\text{s}$. With a single power LED in combination with a half mirror placed under an angle of 45° with respect to the lens coaxial illumination is realized, see Fig. 3.11. Light from the power LED is deflected towards the wafer by the half mirror. The light is then reflected by the semiconductor products, and travels back through the half mirror again and the lens forming the image on the image sensor of the camera. A typical image of a part of the wafer is given in Fig. 3.1(a).

A particular production process to obtain the final product is the picking and placement and wire bonding of each individual semiconductor. Therefore, the tool is to be positioned accurately with respect to each semiconductor. As mentioned, we assume the tool is positioned at the center of the image sensor, such that the problem at hand is transformed into controlling the xy-wafer stage such that the semiconductor is accurately positioned with respect to the center of the image, where the camera is used as position sensor instead of the on board motor encoders. The feature-based position as explained previously is used for feedback. The camera is connected to a PC running a 2.6.28.3 low-latency Linux kernel on which the necessary image processing is done and the control law K is calculated. The real-time executable is built using the real-time workshop (RTW) of Matlab/Simulink. Furthermore, the data-acquisition is realized using an EtherCAT (Jansen and Buttner, 2004) data-acquisition system, where DAC, I/O, and encoder modules are installed to drive the current amplifiers of the motors, to enable the amplifiers and to measure the position of the xy-wafer stage at the motor

side via the on board motor encoders. Hence, this motor encoder position is only used for evaluation purpose and is *not* used in the final control algorithm as such.

3.6 Control design and stability analysis

In this section first the design of the controller K will be presented. Due to the introduction of feature-based positions, the gain of the plant varies dependent on the momentary pitch between successive features. Therefore, a stability analysis will be given for investigating the closed-loop stability while robustness against pitch imperfections is guaranteed.

The transfer from input \underline{u} to the feature-based position output \underline{p}_i^f is denoted by G , see also Fig. 3.9. For this transfer a frequency response function (FRF) has been measured with a repetitive structure with a pitch of $P = \bar{P}$ from which the diagonal terms are given in gray in Fig. 3.12. During the FRF measurement, the rotation of the repetitive structure with respect to the camera was assumed to be zero. Furthermore, the principle directions of the actuation system are assumed to coincide with the principle axes of the camera such that the off-diagonal terms are zero, i.e., no coupling is present between the two principle directions. For low frequencies the diagonal terms can be approximated by double integrators or mass systems. However, for high frequencies the diagonal terms show several resonances and anti-resonances which are due to the flexibilities of the system. These flexibilities are caused for example by cable slab, the finite stiffness of the frame that is supporting the camera, and the finite stiffness of the xy-wafer stage itself. For the nominal plant described as above a nominal diagonal controller $K = \text{diag}(K_x, K_y)$ is designed. The controllers K_x and K_y are designed in continuous time and then discretized. The controllers are built by the series connection of a gain, an integral action, a lead filter, a notch and a second order low-pass filter, see Fig 3.13. The values of the controller tuning are given in Table 3.2. The bandwidth of both control loops is 20 Hz, as can be seen from the open-loop FRFs shown in Fig. 3.14.

In the remainder of this section the stability of the closed-loop system will be investigated. More specifically, it is investigated if the closed-loop system with the diagonal controller, designed on the basis of the nominal plant, is robustly stable against pitch imperfections, i.e., in cases that $P \neq \bar{P}$.

If the repetitive structure moves with a pixel velocity \underline{v}^p then the pixel velocity of each feature is the same, i.e.,

$$\dot{\underline{p}}_{tl}^p = \dot{\underline{p}}_{tr}^p = \dot{\underline{p}}_{bl}^p = \dot{\underline{p}}_{br}^p = \underline{v}^p. \quad (3.15)$$

As a result of this movement a feature-based velocity $\dot{\underline{p}}_i^f$ will be induced. The

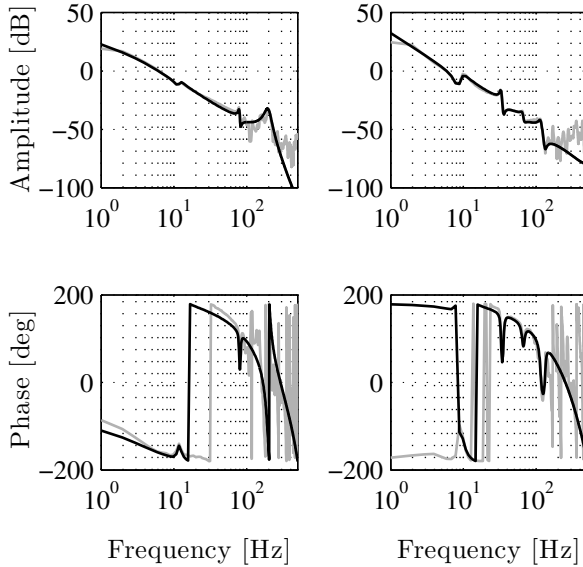


Figure 3.12: The measured diagonal term of the FRF from input \underline{u} to the feature-based position output \underline{p}_i^f with $P = \bar{P}$ is given in gray. Each frequency response function is fitted with a model which is given in black. The left figures show the transfer function from u_x to x_i^f , whereas the right figures show the transfer function from u_y to y_i^f .

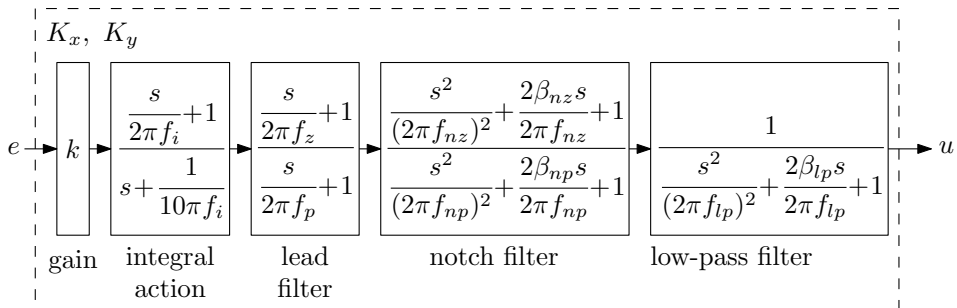
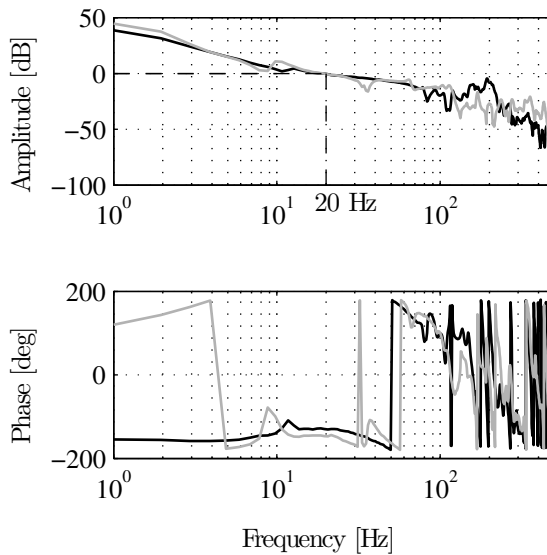


Figure 3.13: Controller structure for K_x and K_y .

Table 3.2: Controller tuning parameters.

	K_x	K_y	Unit
k	57.7	62.1	-
f_i	5	5	Hz
f_z	5	5	Hz
f_p	160	160	Hz
f_{nz}	78	33	Hz
β_{nz}	0.06	0.02	-
f_{np}	78	37	Hz
β_{np}	0.14	0.12	-
f_{tp}	250	250	Hz
β_{tp}	0.6	0.6	-

Figure 3.14: Open-loop in x (black) and y (gray) direction.

same feature-based velocity would occur if we virtually move the pixel position of the point of interest in the opposite direction

$$\underline{\dot{p}}_i^p = -\underline{v}^p, \quad (3.16)$$

while the pixel positions of the features are kept static. The relation between the pixel velocity \underline{v}^p and the feature-based velocity $\underline{\dot{p}}_i^f$ can be written as

$$\underline{\dot{p}}_i^f = J \underline{\dot{p}}_i^p = -J \underline{v}^p, \quad (3.17)$$

with $J \in \mathbb{R}^{2 \times 2}$ being the Jacobian defined as

$$J = \frac{d\underline{\dot{p}}_i^f}{d\underline{\dot{p}}_i^p} = \begin{pmatrix} \frac{dx_i^f}{dx_i^p} & \frac{dx_i^f}{dy_i^p} \\ \frac{dy_i^f}{dx_i^p} & \frac{dy_i^f}{dy_i^p} \end{pmatrix} = \begin{pmatrix} \frac{dg}{dA} \frac{dA}{dx_i^p} & \frac{dg}{dA} \frac{dA}{dy_i^p} \\ \frac{dh}{dB} \frac{dB}{dx_i^p} & \frac{dh}{dB} \frac{dB}{dy_i^p} \end{pmatrix}, \quad (3.18)$$

where the product rule is used. When the repetitive structure has a nominal pitch $P = \bar{P}$ in both directions, and the repetitive structure is perfectly aligned with the camera coordinate system, the Jacobian J simply reduces to

$$J = \begin{pmatrix} \frac{1}{\bar{P}} & 0 \\ 0 & \frac{1}{\bar{P}} \end{pmatrix}. \quad (3.19)$$

However, in the presence of pitch imperfections, the value of J depends on the pixel coordinates of the four features enclosing the point of interest. Hence, they depend on the momentary pitch P between successive features. Therefore, the Jacobian J becomes a non-linear mapping from \underline{v}^p to $\underline{\dot{p}}_i^f$. At this point it is investigated how the Jacobian varies as a function of the pitch imperfection. Therefore, we write the pitch imperfection as a fraction of the nominal pitch, so $\Delta P = \alpha \bar{P}$, with $0 \leq \alpha < 0.25$. The upper bound for α indicates the validity of the bilinear feature-based interpolation. This is shown in Fig. 3.15. In this figure the pitch imperfection $\Delta P = 0.25 \bar{P}$. It can be seen that three features are forming a straight line. If the imperfections are allowed to be larger, than the four features do not span a convex set any more.

For each value of α we can find the maximum and minimum values of J over all possible feature position configurations satisfying the bounds as defined in (3.1). In this case the maximum and minimum values of the elements of the Jacobian J

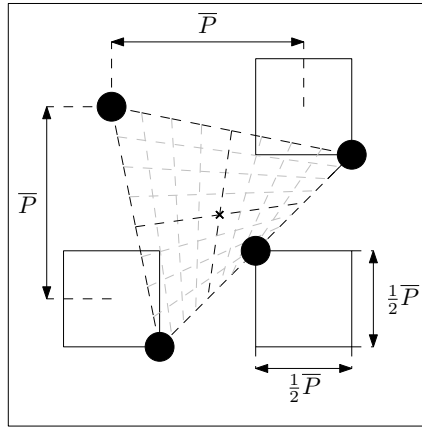


Figure 3.15: Validity of the feature interpolation concept.

are found to satisfy (see Appendix D) analytic functions of α that are given by

$$\bar{j}_d = \frac{1}{\bar{P}} \left(\frac{0.125}{0.25 - \alpha} + 0.5 \right) \left((2\sqrt{10} - 4) \alpha + 1 \right), \quad (3.20)$$

$$\underline{j}_d = \frac{1}{\bar{P}} (1 - 2\alpha) \left((4 - 2\sqrt{10}) \alpha + 1 \right), \quad (3.21)$$

$$\bar{j}_{od} = \frac{1}{\bar{P}} \left(\frac{0.125}{0.25 - \alpha} - 0.5 \right) \left((2\sqrt{10} - 4) \alpha + 1 \right), \quad (3.22)$$

$$\underline{j}_{od} = \frac{1}{\bar{P}} \left(-\frac{0.125}{0.25 - \alpha} + 0.5 \right) \left((4 - 2\sqrt{10}) \alpha + 1 \right). \quad (3.23)$$

This result is graphically shown in Fig. 3.16. In this figure the four elements of the Jacobian J are shown. On the horizontal axis the value of α is given, whereas on the vertical axis the possible values are indicated by the gray shaded areas. As can be seen in this figure an increase of α leads to a larger set of possible values of J . Using this figure, one can determine what possible values of the Jacobian can be present given a specific pitch imperfection. The nominal pitch \bar{P} between consecutive features is 27 pixels. It can also be seen that if $\alpha = 0$, i.e., no pitch imperfections, the Jacobian satisfies (3.19).

For proving stability of the closed-loop system we try to write our system as a linear differential inclusion (LDI) (Boyd et al., 1994). In obtaining such an LDI we first show that at a specific value of α an arbitrary Jacobian can be written as the convex combination of all possible minimum and maximum value combinations of the Jacobian. The number of possible combinations is the number of elements of the Jacobian squared, so $4^2 = 16$. Therefore an arbitrary Jacobian can be written

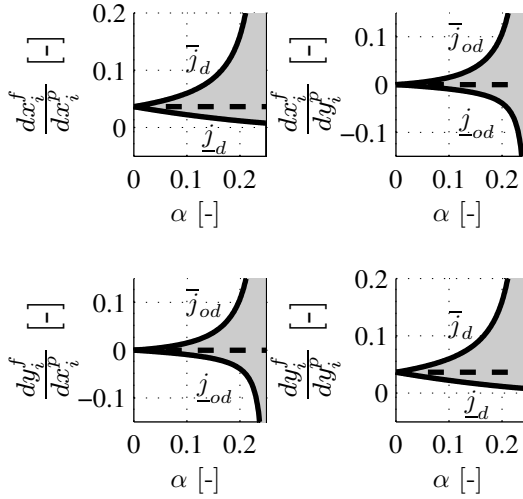


Figure 3.16: The values of the Jacobian J as a function of α .

as

$$J \in \left\{ \sum_{m=1}^{16} \beta_m J_m \mid J_m \in \mathcal{J}, \forall \beta_m \in \mathbb{R}, \beta_m \geq 0, \sum_{m=1}^{16} \beta_m = 1 \right\}, \quad (3.24)$$

with

$$\mathcal{J} = \{J_1, J_2, \dots, J_{16}\} \quad (3.25)$$

$$= \left\{ \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix} \mid j_{11}, j_{22} \in \{\bar{j}_d, \underline{j}_d\}, j_{12}, j_{21} \in \{\bar{j}_{od}, \underline{j}_{od}\} \right\}. \quad (3.26)$$

To prove stability of the closed-loop system the system $G(z)$ is written as the multiplication of $R(z)$ and an integrator $I(z)$. The system $R(z)$ maps the inputs to the system \underline{u} to the pixel velocity of the features \underline{v}^p . In $I(z)$, these pixel velocities \underline{v}^p are mapped to the feature-based velocities $\underline{\dot{p}}_i^f$ by the Jacobian J , which after integration lead to the feature-based position \underline{p}_i^f . The subsystems $R(z)$ and $I(z)$

are given by

$$R : \begin{cases} \underline{x}_R(k+1) = A_R \underline{x}_R(k) + B_R \underline{u}(k), \\ \underline{v}^p(k) = C_R \underline{x}_R(k) + D_R \underline{u}(k), \end{cases} \quad (3.27)$$

$$I : \begin{cases} \underline{x}_I(k+1) = \underline{x}_I(k) - JT \underline{v}^p(k), \\ \quad \quad \quad = \underline{x}_I(k) + T \underline{\dot{p}}_i^f(k), \\ \underline{p}_i^f(k) = \underline{x}_I(k), \end{cases} \quad (3.28)$$

with T the sample time of the system. The total system G is therefore

$$G : \begin{cases} \underline{x}_G(k+1) = A_G \underline{x}_G(k) + B_G \underline{u}k, \\ \underline{p}_i^f(k) = C_G \underline{x}_G(k), \end{cases} \quad (3.29)$$

with $\underline{x}_G = (\underline{x}_R^T \quad \underline{x}_I^T)^T$ and

$$A_G = \begin{pmatrix} A_R & 0 \\ -JT C_R & I \end{pmatrix}, B_G = \begin{pmatrix} B_R \\ -JT D_R \end{pmatrix}, C_G = (0 \quad I).$$

By defining the feature-based error as $\underline{e} = \underline{r}_i^f - \underline{p}_i^f$, with \underline{r}_i^f being the feature-based position reference, the closed-loop system can be calculated as

$$\underline{x}(k+1) = A_{cl} \underline{x}(k) + B_{cl} \underline{r}_i^f(k), \quad (3.30)$$

$$\underline{p}_i^f(k) = C_{cl} \underline{x}(k), \quad (3.31)$$

with $\underline{x} = (\underline{x}_G^T \quad \underline{x}_K^T)^T$ and

$$A_{cl} = \begin{pmatrix} A_G - B_G D_K C_G & B_G C_K \\ -B_K C_G & A_K \end{pmatrix}, B_{cl} = \begin{pmatrix} B_G D_K \\ B_K \end{pmatrix}, C_{cl} = (C_G \quad 0).$$

To assess the stability of this closed-loop system we note that the system can now be written as an LDI (Boyd et al., 1994). That is,

$$A_{cl} \in \left\{ \sum_{m=1}^{16} \beta_m A_{cl,m} \mid \beta_m \in \mathbb{R}, A_{cl,m} \in \mathcal{A}_{cl}, \sum_{m=1}^{16} \beta_m = 1 \right\}, \quad (3.32)$$

where the set \mathcal{A}_{cl} is the set of closed-loop matrices evaluated for every Jacobian $J \in \mathcal{J}$. Using this definition we can search for a common quadratic Lyapunov function $V(\underline{x}) = \underline{x}^T E \underline{x}$ with $E = E^T \succ 0$ such that $V(\underline{x}(k)) - V(\underline{x}(k+1)) > 0, \forall \underline{x}(k+1) = A_{cl,i} \underline{x}(k)$ or similarly by simultaneously checking the following linear matrix inequalities (LMIs)

$$E - A_{cl,i}^T E A_{cl,i} \succ 0, \quad i \in \{1, \dots, 16\} \quad (3.33)$$

$$E \succ 0. \quad (3.34)$$

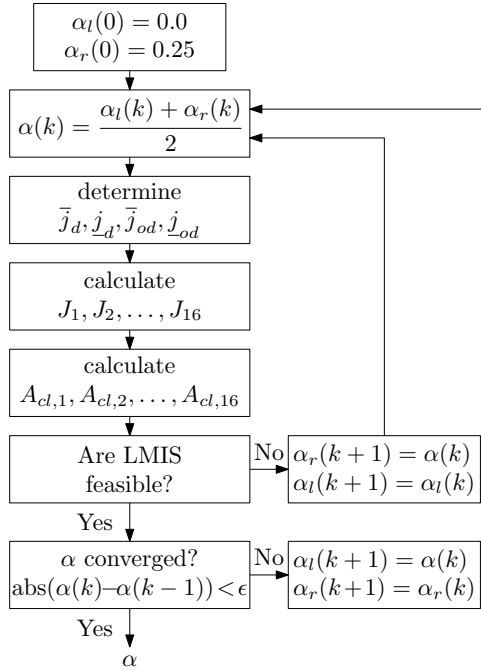


Figure 3.17: Bisection algorithm used to determine maximum value of α .

For a given value of α however the possible values of the Jacobian will not all be on the extreme boundaries as indicated by the black lines in Fig. 3.16. Therefore the conditions for stability given above are sufficient but conservative. Given the closed-loop system it can be investigated up to which value of α the closed-loop system is stable using a bisection algorithm as depicted in Fig. 3.17. The LMIs (3.33) through (3.34) can be solved efficiently using commercially available software (Gahinet et al., 1994). The found value of $\alpha = 0.095$. Therefore, we can conclude that the closed-loop system is guaranteed stable for pitches that satisfy

$$0.905\bar{P} \leq P \leq 1.095\bar{P}. \quad (3.35)$$

Note that this is an a posteriori stability analysis. The considered imperfections in this work are within this boundary. If however the imperfections are outside this boundary, a redesign of the controller K is necessary to be robustly stable against the pitch imperfections.

3.7 Results

The use of the proposed feature-based position measurement is validated in practice on the experimental setup to demonstrate the effectiveness. Therefore, two experiments have been carried out. In the first experiment the improvement of using second order interpolation will be showed. The control task during the experiment is to move the wafer in one direction with a constant feature-based velocity of 36.8 f/s (approximately 0.01 m/s) while pitch imperfections are present. The feature-based velocity obtained from numerical differentiation of the feature-based position is given in Fig. 3.18. The vertical dashed lines indicate when a feature is passed, such that a different pitch is considered. It can be seen that the feature-based velocity switches when using the feature-based position of (3.11), especially around $t = 8.18$ s and $t = 8.21$ s. This instantaneous switching feature-based velocity is not present when the second order interpolation (3.12) is used. The power spectrum of the feature-based velocity is shown in Fig. 3.19. It shows that around 36.8 Hz the power content is increased. This is expected since this frequency corresponds to the applied reference velocity of 36.8 f/s. Furthermore, the power content for frequencies up to approximately 200 Hz is reduced. For frequencies above 200 Hz, there is approximately no difference. The feature-based error is given in Fig. 3.20. Using the second order interpolation the error is reduced by approximately 40 %.

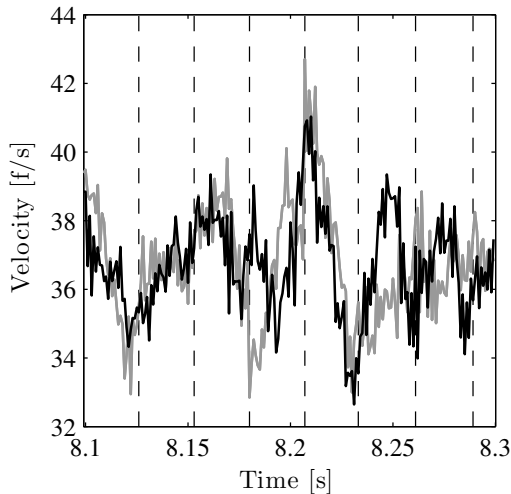


Figure 3.18: Switching feature-based velocity. In gray the feature-based position as defined in (3.11) is used for feedback whereas in black (3.12) was used.

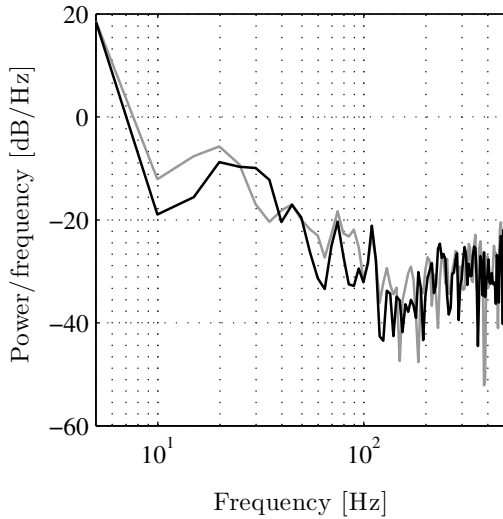


Figure 3.19: Power spectrum of the feature-based velocity. In gray the feature-based position as defined in (3.11) is used for feedback whereas in **black** (3.12) was used.

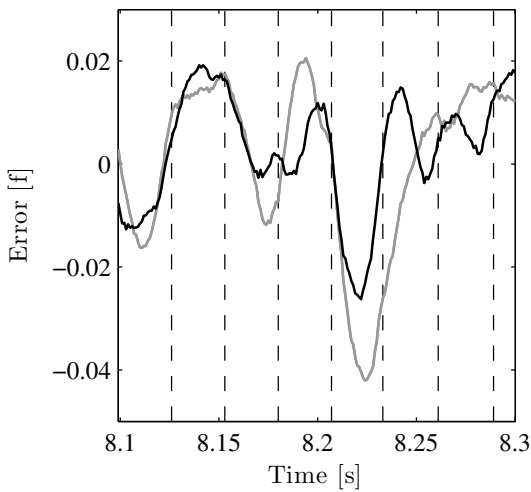


Figure 3.20: Feature-based error. In gray the feature-based position as defined in (3.11) is used for feedback whereas in **black** (3.12) was used.

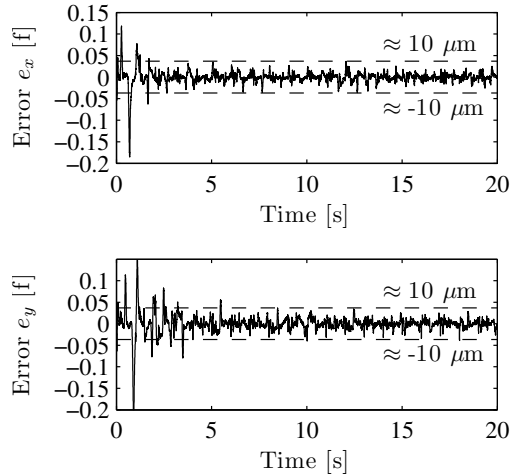


Figure 3.21: Measured feature-based error e .

In the second experiment the reference given in Fig. 3.8 is applied to the closed-loop system, such that the wafer is moved diagonally over the semiconductors, while at each semiconductor the contour is tracked. Since the reference is repetitive in time, the controllers K_x and K_y are expanded with a standard add-on repetitive controller (Hara et al., 1988) to improve the performance. The final positioning error is shown in Fig. 3.21. After a small learning transient the final feature-based error is less than 0.05 f. The nominal pitch between the semiconductors is 27 pixels, which with a pixel size of $9.9 \mu\text{m}$ is $267 \mu\text{m}$. In Fig. 3.21 we indicated the metric position error of $\pm 10 \mu\text{m}$.

3.8 Conclusions

In this chapter a novel feature-based motion control approach is presented, which uses two-dimensional feature-based positions for feedback. The advantage of using these feature-based positions is that online trajectory generation has become redundant in case the metric target position is unknown a priori due to the pitch imperfections between successive features. A stability analysis proves robust stabil-

ity of the closed-loop system while pitch imperfections up to approximately 10% are considered. A second order feature interpolation is implemented to reduce transient responses caused by instantaneously switching feature-based velocities. Experimental validation showed that this leads to an error reduction of 40%. Next to feature-to-feature movements, relative feature movements have been incorporated in the feature-based control approach; operators can easily specify how the tool is to be aligned with respect to the feature as a function of time. During experiments full two-dimensional feature-to-feature movements as well as relative feature movements are applied resulting in position errors less than 10 μm .

Chapter 4

Iterative learning control for scale varying setpoints

ITERATIVE Learning Control (ILC) is a control technique for systems subject to repetitive setpoints or disturbances. However in many applications the setpoint is not strictly repetitive, and the learning process should start all over from the beginning if the setpoint changes. In this chapter point-to-point movements with different magnitudes will be considered which are constructed by scaling a nominal setpoint. Second order ILC with an adaptive low-pass filter in the trial domain is used to accurately track these scale varying setpoints under the influence of disturbances that are either 1) repetitive or 2) experience the same scaling as the setpoint. Experiments have been carried out to validate the proposed method.

4.1 Introduction

In many manufacturing processes, production steps are carried out on repetitive structures consisting of identical features placed in a repetitive pattern. Examples of repetitive structures are given in Fig. 4.1(a) and Fig. 4.1(b). In many

This chapter is based on: J.J.T.H. de Best, M.J.G. van de Molengraft and M. Steinbuch. Second Order Iterative Learning Control for Scale Varying Setpoints. *Submitted for journal publication.*

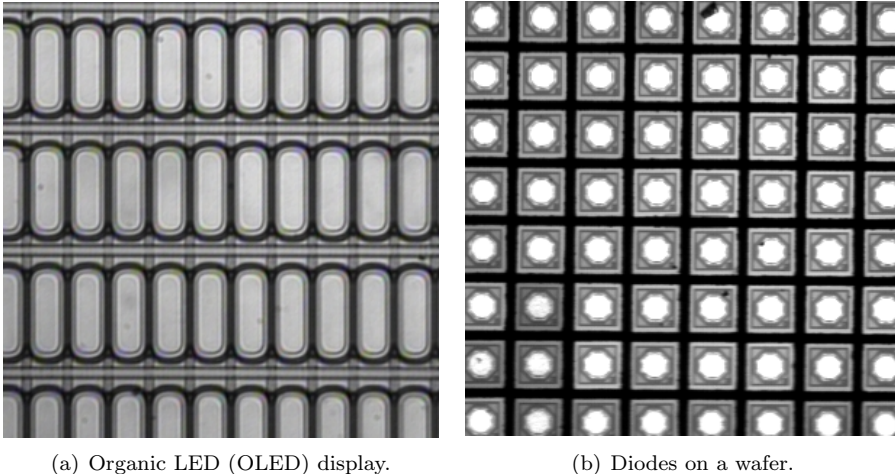


Figure 4.1: Examples of repetitive structures.

of these production steps the tool is to be aligned with respect to a feature of the repetitive structure, perform its task and move towards the next feature via a point-to-point setpoint. Most conventional control approaches use a feedback controller for plant stabilization and disturbance rejection in combination with a feedforward controller according to a predefined structure (e.g. mass, damping and coulomb friction) to increase the overall performance. However, there are limitations using feedback and fixed-structured feedforward control. For example, the closed-loop bandwidth can be limited by system dynamics resulting in a limited disturbance rejection, while the fixed-structure feedforward control may not be sufficiently rich to capture the disturbances at hand (Moore, 1993; Reichardt, 2010; Van der Meulen et al., 2008). For systems that have to track a predefined setpoint over and over again a well known control technique for achieving a convergent feedforward signal called iterative learning control (Moore, 1993) (ILC) can be applied. ILC is a control technique that reduces the tracking error along a trajectory that is traced repeatedly by the iterative refinement of a feedforward signal. Good surveys of recent ILC research can be found in (Moore, 2008; Bristow et al., 2006; Ahn et al., 2007; Moore et al., 2006).

One constraint within ILC is that the setpoint needs to be strictly repetitive for every trial. In practice, however, the distance between consecutive features of the considered repetitive structures may vary due to temperature changes, stretch and manufacturing tolerances of previous processing steps. Hence, the setpoint to move from one feature to the next is not strictly repetitive but varies due to these pitch variations. Applying ILC for varying setpoints is one of the challenges

in current ILC research. The control problem that will be treated here is how to design ILC for point-to-point movements with varying travel distances in practice. Several methods have been developed to use the knowledge from previous ILC trials to construct the feedforward signals for new, different setpoints.

In (Rotariu et al., 2003) time-frequency adaptive ILC is proposed, in which a feedforward signal is learned for one specific setpoint and zeros are inserted in the feedforward signal for setpoints with longer zero acceleration length. In (Rotariu et al., 2008) a piecewise ILC is presented, in which a time-varying robustness Q filter with adaptive cutoff frequency is used. In both of these works different setpoints are generated using a constant velocity phase with variable length. More general trajectories are considered in (Heertjes and Van de Molengraft, 2009) in which a finite impulse response (FIR) mapping strategy is proposed based on converged learning forces obtained with learning control at a specific acceleration set-point profile. Besides FIR, a multi-table approach is also considered. In (Hoelzle et al., 2010) basis tasks are learned a priori and allows the reference trajectory to be arbitrarily chosen, provided it is comprised of the defined basis tasks. Direct Learning Control (DLC) (Xu, 1997) and Recursive Direct Learning Control (RDLC) (Xu et al., 2002) are developed to generate the desired control signal for a new setpoint using several pre-stored setpoints and control signals. The methods presented in (Heertjes and Van de Molengraft, 2009; Hoelzle et al., 2010; Xu, 1997; Xu et al., 2002) all need the converged feedforward signals learned from one or several specific setpoints or basis tasks to construct the new control input for a different setpoint or task. In this work, we will present a *second order* ILC algorithm in which *scale varying* setpoints are applied *during* the learning process and for which the tracking error will be reduced iteratively, i.e., the process of learning signals a priori is not necessary in our approach.

High order ILC was studied in (Bien and Huh, 1989; Chen et al., 1997; Huh, 1997; Chen et al., 1992, 1998; Kim et al., 2003; Moore and Chen, 2002; Norrlöf, 2000). It is shown that high order ILC is useful to increase the convergence speed (Bien and Huh, 1989; Chen et al., 1992; Kim et al., 2003), reject disturbances that satisfy an a priori relation from one trial to the next (Moore and Chen, 2002) and has robustness in the presence of external disturbances (Kim et al., 2003). In this work second order ILC will be used to iteratively identify different classes of disturbances which are used for updating the feedforward signal.

This work focuses on accurate tracking of point-to-point setpoints with different displacement magnitudes which are constructed by scaling a nominal setpoint to the desired travel distance. Iterative learning control is used to update the feedforward signal while *during* iterations different scale varying setpoints are applied. We analyze the convergence of the tracking error for situations with disturbances and without disturbances. In this work it is assumed that these disturbances are 1) repetitive every trial and/or 2) experience the same scaling as the setpoint. A second order ILC strategy will be used to identify these two types of disturbances

and compensate for them during iterations. The contributions of this work are 1) the design of a second order ILC strategy to accurately track scale varying setpoints in which *during* the learning process these scale varying setpoints are applied and 2) which will be implemented in practice on an industrial setup to show the effectiveness.

The rest of the work is organized as follows. In Section 4.2 the standard ILC method is extended by implementing scaling only, leading to normalized ILC. To further improve the performance, second order ILC will be derived in Section 4.4. In Section 4.5 results of experiments will be given where the different methods will be compared. Section 4.6 will present the conclusions.

4.2 Standard ILC and normalized ILC

In this section we will briefly discuss the ILC working principle, see also (Merry et al., 2008; Steinbuch and van de Molengraft, 2000). To explain ILC, the block scheme of the control loop given in Fig. 4.2 is used, where the controller is denoted by $K(z)$ and the plant is denoted by $G(z)$, which are both assumed to be discrete and linear time invariant (LTI). A schematic representation of the plant that is considered in this work is given in Fig. 4.3. For now we take the gain $T_k = 1$ and assume the pitch between successive feature is perfectly repetitive. The

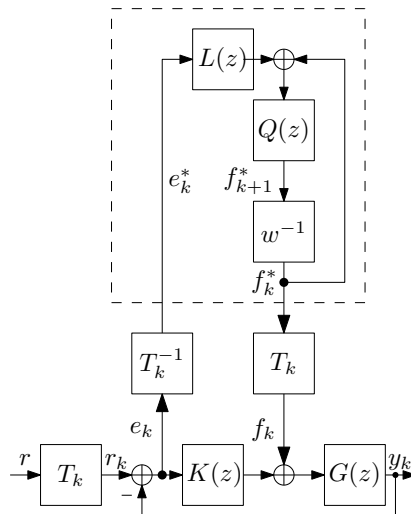


Figure 4.2: (Normalized) iterative learning control scheme.

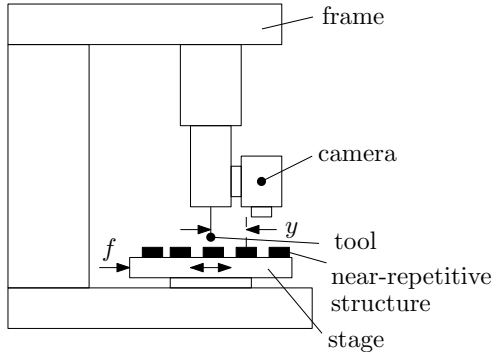


Figure 4.3: Schematic representation of the plant. A near-repetitive structure, e.g., a wafer with discrete semiconductors as given in Fig. 4.1(b), is mounted on an actuated stage with an input force denoted by f . The goal is to position the tool with respect to the features of the near-repetitive structure. In this work the relative position measurement between the tool and the features y is measured by means of a camera and is used for feedback.

time shift operator in Fig. 4.2 is denoted by z , i.e., $z^{-1}x(t) = x(t - 1)$, here t represents the sample number. The trial shift operator is denoted by w , i.e., $w^{-1}x_k(t) = x_{k-1}(t)$, where k represents the trial number, (Moore and Chen, 2002; Norrlöf, 2000). The repetitive setpoint is given by $r(t)$, with $t \in [0, N]$, whereas the output at trial k is denoted by $y_k(t)$. During trial k the feedforward signal $f_k(t)$ is applied and the error $e_k(t)$ is measured. In an off-line operation, the error signal is filtered with the filter $L(z)$, called the learning filter and added to the feedforward $f_k(t)$. This learning filter is chosen as an approximation of the inverse of the process sensitivity $S_p(z)$ and can be designed, for instance, using the zero-phase error tracking controller (ZPETC) algorithm (Tomizuka, 1987). This algorithm cancels all the closed-loop poles and cancelable closed-loop zeros. For uncancelable zeros, which include the non-minimum phase zeros, it cancels the phase shift and compensates the gain introduced by them. After filtering the error through $L(z)$, the robustness filter $Q(z)$ is applied to the sum of the filtered error and the feedforward signal $f_k(t)$, which results in the feedforward signal $f_{k+1}(t)$ that is applied in the next trial $k+1$. The offline updating of the feedforward signal is graphically depicted in the dashed box in Fig. 4.2, whereas mathematically it is written as

$$f_{k+1}(t) = Q(z)(f_k(t) + L(z)e_k(t)). \quad (4.1)$$

The tracking error e in trial $k+1$ can be written as

$$e_{k+1}(t) = S(z)r(t) - S_p(z)f_{k+1}(t), \quad (4.2)$$

where $S(z) = 1/(1+G(z)K(z))$ is the sensitivity and $S_p(z) = G(z)/(1+G(z)K(z))$ is the process sensitivity. Substitution of the update law (4.1) into (4.2) leads to

$$e_{k+1}(t) = S(z)r(t) - Q(z)S_p(z)(f_k(t) + L(z)e_k(t)). \quad (4.3)$$

Similar to (4.2) we use the fact that $S_p(z)f_k(t) = S(z)r(t) - e_k(t)$ and substitute this into (4.3) such that the error at trial $k + 1$ becomes a function of the error in the previous trial k :

$$e_{k+1}(t) = Q(z)(1 - L(z)S_p(z))e_k(t) + (1 - Q(z))S(z)r(t). \quad (4.4)$$

The above system is called an *linear iterative system* for which convergence is obtained when (see (Norrlöf, 2000))

$$\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_\infty < 1, \quad \forall \omega \in [-\pi, \pi], \quad (4.5)$$

is satisfied.

As opposed to the repetitive setpoint considered in standard ILC, point-to-point setpoints with varying travel distances will be considered in this work since small variations are present in the distance between successive features. An example of a nominal second order setpoint is given in gray in Fig. 4.4. In general, two ways of setpoint generation for different magnitudes are

1. keep the periods and magnitudes for the acceleration and deceleration phase fixed while enlarging the constant velocity time (Rotariu et al., 2003). An example of such a stretched second order setpoint is given in Fig. 4.4 by the dashed line.
2. keep all periods fixed, but scale the magnitude of the acceleration, velocity and position (Xu, 1997; Xu et al., 2002). The black line in Fig. 4.4 presents the scaled setpoint.

In this work we handle setpoint variation using the second type and scale a nominal setpoint r by a gain T_k , which results in a setpoint r_k that is used in the trial k , i.e., $r_k(t) = T_k r(t)$. The value of T_k is assumed to be bounded by $T_k \in [\underline{T}, \bar{T}]$, where $\underline{T} \in \mathbb{R}^+$ is the lower bound and $\bar{T} \in \mathbb{R}^+$ is the upper bound of the gain, which are directly related to the pitch variation present in the repetitive structure. The scaling factor T_k with which the nominal setpoint is scaled, can be determined a priori by taking a snapshot of the scene. The center of the camera is initially located above the center of a feature, while the next neighboring target feature is already in the field of view. From this, the distance between the features can be determined and the scaling factor can be determined. We use this second type such that the time to reach each target is the same. Moreover, the switching times

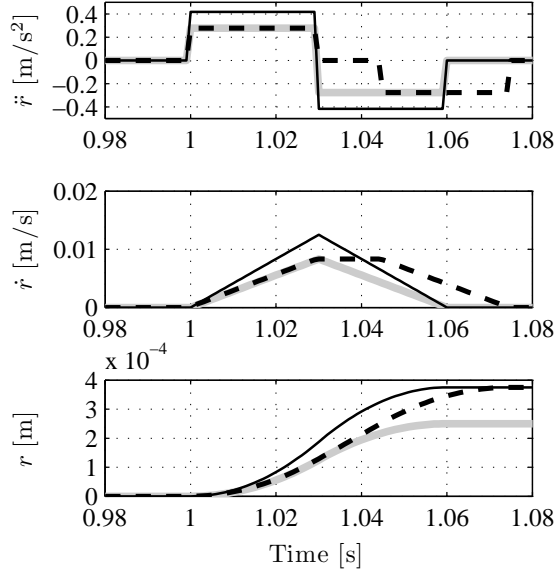


Figure 4.4: Setpoint generation for different magnitudes: nominal setpoint (**gray**), stretched setpoint (**dashed**), scaled setpoint (**black**).

for the acceleration in this case remain the same, such that we can exploit the use of scaling.

As standard ILC can only cope with a *repetitive* setpoint, the error will not converge if during iterations these *scale varying* setpoints r_k are applied. To handle scale varying setpoints, we proceed as follows. Standard ILC can be extended by incorporating the gain T_k before and after the ILC update block as depicted in Fig. 4.2, which will be referred to as normalized ILC (NILC) in the remainder of this work. The learning update uses the normalized error $e_k^*(t) = T_k^{-1}e_k(t)$ to construct a normalized feedforward signal $f_{k+1}^*(t)$ as shown in the dashed area in Fig. 4.2, whereas the applied feedforward signal is given by $f_k(t) = T_k f_k^*(t)$. The same analysis to prove convergence can be carried out as above and shows that the error is convergent if (4.5) is satisfied. However, in case disturbances are present in the system, normalized ILC is likely to fail due to the fact that disturbances will not scale in general.

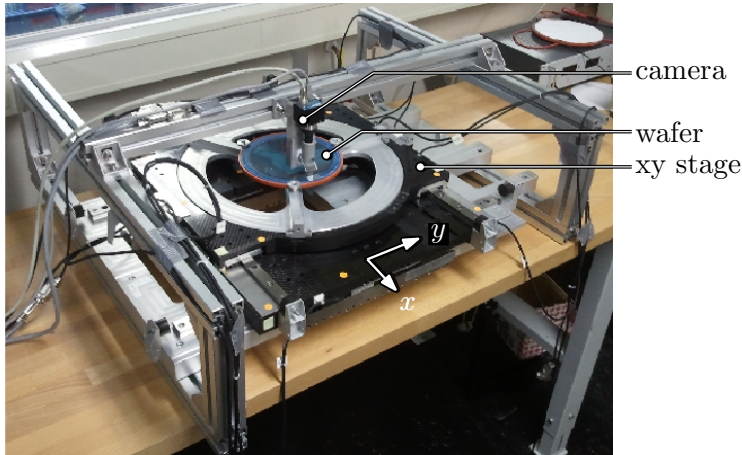


Figure 4.5: Industrial application: an xy-wafer stage.

4.3 Existence of disturbances

The underlying assumption of normalized ILC is that the error scales with the same gain as the setpoint in every trial. In practice, however the error is also influenced by the presence of disturbances, such that this assumption does not hold, see also (Merry et al., 2005). In this work, three different types of disturbances will be considered:

- Ⓘ disturbances that *experience the same scaling* as the setpoints, i.e., $d_k(t) = T_k d(t)$. Viscous damping in a mechanical motion system is an example of such a disturbance. Moreover, the reference itself can be seen as one.
- Ⓜ disturbances that are *repetitive every trial*, i.e., $d_{k+1}(t) = d_k(t) = \dots = d(t)$. These kind of disturbances are for example caused by an amplifier offset, gravity forces or dry friction.
- Ⓝ disturbances that are of a *random* nature such as sensor noise.

The existence of disturbances is investigated in practice by using the industrial application depicted in Fig. 4.5. The setup consists of an xy-stage on which a wafer with discrete semiconductors (see Fig. 4.1(b)) is mounted. The stage is actuated by current controlled linear motors. For feedback the metric position of the discrete semiconductors is measured at a rate of 1 kHz by means of a camera

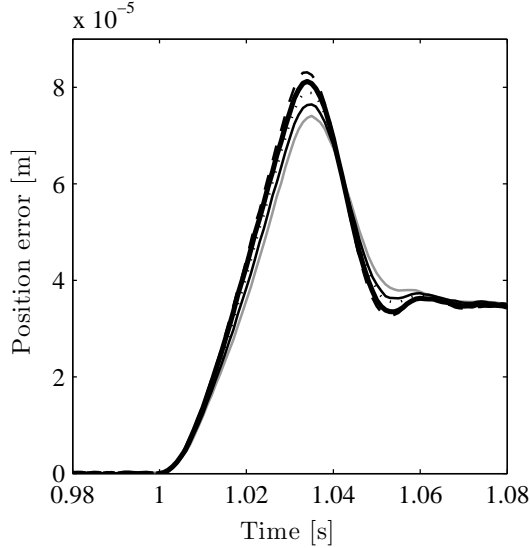


Figure 4.6: Measured errors for different gains, gray: $T_1 = 0.9$, black: $T_2 = 0.95$, dotted black: $T_3 = 1$, **bold**: $T_4 = 1.05$, dashed black: $T_5 = 1.1$.

which is mounted above the stage, leading to a non-collocated direct visual servoing control problem. A single-input single-output (SISO) controller is tuned using classical loopshaping techniques (Franklin et al., 1994) such that the closed-loop system has a bandwidth of 20 Hz, see also Section 4.5.

Next, five experiments are conducted, in which each time the nominal setpoint of Fig. 4.4 is scaled by a different gain T_k and applied to the closed-loop control system without feedforward, while measuring the resulting position error. The gains T_k that are used in the experiments are $T_1 = 0.9$, $T_2 = 0.95$, $T_3 = 1$, $T_4 = 1.05$ and $T_5 = 1.1$. The corresponding errors are given in Fig. 4.6. A first observation is that the five measured errors have a similar shape, but each with different amplitude. From scaling, the errors are expected to satisfy

$$\frac{e_i(t)}{T_i} = \frac{e_j(t)}{T_j}, \quad i, j \in \{1, \dots, 5\}. \quad (4.6)$$

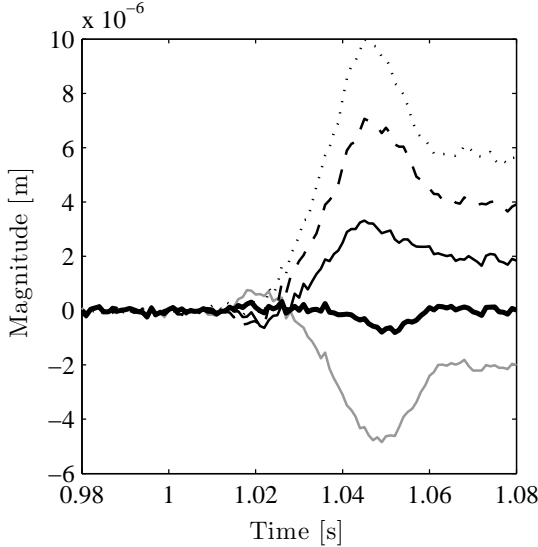


Figure 4.7: Differences between measured error $e_2(t)$ and approximations by scaling and by combination of $e_1(t)$ and $e_3(t)$, gray: $e_2(t) - \frac{T_2}{T_1}e_1(t)$, black: $e_2(t) - \frac{T_2}{T_3}e_3(t)$, dashed black: $e_2(t) - \frac{T_2}{T_4}e_4(t)$, dotted black: $e_2(t) - \frac{T_2}{T_5}e_5(t)$, **bold**: difference between measured error $e_2(t)$ and approximation by combination of $e_1(t)$ and $e_3(t)$ given in (4.11).

Using this, the error $e_2(t)$ for example can be estimated from $e_1(t)$, $e_3(t)$, $e_4(t)$ and $e_5(t)$ as

$$\frac{T_2}{T_i}e_i(t), \quad i \in \{1, 3, 4, 5\}. \tag{4.7}$$

The differences of the measured error $e_2(t)$ and these estimated errors are given in Fig. 4.7. It can be seen that the differences are not exactly zero, since disturbances of type (II) and (III) are present. Considering these types of disturbances, a much more accurate estimate of $e_2(t)$ can be computed by taking combinations of errors. This is explained using Fig. 4.8 where, without feedforward $f_k(t) = 0$ and without noise $n_k(t) = 0$, the errors are given by

$$e_i(t) = T_i g(t) + h(t), \quad i \in \{1, \dots, 5\}, \tag{4.8}$$

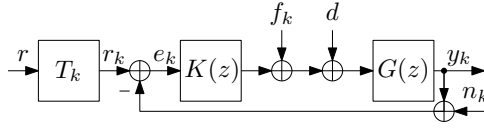


Figure 4.8: Feedback control structure with a *repetitive* input disturbance d .

with

$$g(t) = S(z)r(t), \quad h(t) = -S_p d(t). \quad (4.9)$$

Note that $g(t)$ originates from a type $\textcircled{\text{I}}$ disturbance which is the reference $r(t)$ in this case. The signal $h(t)$ originates from a type $\textcircled{\text{II}}$ disturbance, being the repetitive disturbance $d(t)$. From two measurements, for example $e_1(t)$ and $e_3(t)$, we can estimate $g(t)$ and $h(t)$ by

$$\tilde{g}(t) = \frac{e_3(t) - e_1(t)}{T_3 - T_1}, \quad \tilde{h}(t) = \frac{T_3 e_1(t) - T_1 e_3(t)}{T_3 - T_1}. \quad (4.10)$$

Using this, $e_2(t)$ can now be estimated more accurately as

$$T_2 \tilde{g}(t) + \tilde{h}(t) = \frac{T_3 - T_2}{T_3 - T_1} e_1(t) + \frac{T_2 - T_1}{T_3 - T_1} e_3(t). \quad (4.11)$$

The difference of the measured error $e_2(t)$ and this estimate is also given in Fig. 4.7 in bold. It can be seen that it is much more accurate than the scaled errors. The accuracy of this estimate is less than 1 μm , whereas others are as large as 10 μm . Similar results are obtained when this estimate is constructed with other combinations of errors. In the remainder of this work this idea will be extended to ILC by learning the signals $g(t)$ and $h(t)$, which will lead to second order ILC.

4.4 Second order ILC

In this section first the principle of second order ILC (SOILC) will be explained. Then it will be analyzed under which conditions the proposed SOILC approach is convergent and what the influence of sensor noise and applying similar setpoints is. Finally, improvements will be presented by adding an adaptive low-pass filter in the trial domain for SOILC.

4.4.1 Principle of SOILC

Consider the control scheme in Fig. 4.8 at this moment without sensor noise $n_k = 0$, where the goal is to design a feedforward signal $f_{k+1}(t)$ in such a way that the error $e_{k+1}(t)$ in trial $k + 1$ is zero. Assume we have two trials $k - 1$ and k , for which the errors of these two trials can be written as

$$e_{k-1}(t) = T_{k-1}g(t) + h(t) - S_p(z)f_{k-1}(t), \quad (4.12)$$

$$e_k(t) = T_k g(t) + h(t) - S_p(z)f_k(t). \quad (4.13)$$

After these two trials, the repetitive terms $g(t)$ and $h(t)$ can be estimated similar to (4.10):

$$\tilde{g}(t) = \frac{e_{k-1}(t) - e_k(t)}{T_{k-1} - T_k} + \frac{S_p(z)(f_{k-1}(t) - f_k(t))}{T_{k-1} - T_k}, \quad (4.14)$$

$$\begin{aligned} \tilde{h}(t) = & \frac{T_{k-1}e_k(t) - T_k e_{k-1}(t)}{T_{k-1} - T_k} + \\ & \frac{S_p(z)(T_{k-1}f_k(t) - T_k f_{k-1}(t))}{T_{k-1} - T_k}. \end{aligned} \quad (4.15)$$

Assume the feedforward signal for trial $k + 1$ is $f_{k+1}(t)$, then the error for trial $k + 1$ can be estimated as

$$\begin{aligned} e_{k+1}(t) = & T_{k+1}\tilde{g}(t) + \tilde{h}(t) - S_p f_{k+1}(t) \\ = & (1 - \alpha)e_{k-1}(t) + \alpha e_k(t) + S_p(z)((1 - \alpha)f_{k-1}(t) + \alpha f_k(t)) - S_p f_{k+1}(t), \end{aligned} \quad (4.16)$$

with α defined as

$$\alpha = \frac{T_{k-1} - T_{k+1}}{T_{k-1} - T_k}, \quad T_{k-1} \neq T_k. \quad (4.17)$$

Since we want to design a feedforward signal $f_{k+1}(t)$ in such a way that $e_{k+1}(t) = 0$, from (4.16) we solve f_{k+1} :

$$f_{k+1}(t) = (1 - \alpha)f_{k-1}(t) + \alpha f_k(t) + \frac{1}{S_p(z)}((1 - \alpha)e_{k-1}(t) + \alpha e_k(t)). \quad (4.18)$$

As in standard ILC, the inverse of the process sensitivity $S_p(z)$ is approximated by $L(z)$ and a robustness filter $Q(z)$ can be added (see Section 4.2), leading to the second order update law

$$f_{k+1}(t) = Q(z) \left((1 - \alpha)f_{k-1}(t) + \alpha f_k(t) + L(z)((1 - \alpha)e_{k-1}(t) + \alpha e_k(t)) \right). \quad (4.19)$$

Update law (4.19) can be used for the situation where type ① and ② disturbances are present in the system. However at this point, three questions remain to be answered: 1) is the new linear iterative system convergent, 2) what happens in case $T_k = T_{k-1}$ and 3) how does SOILC deal with type ③ disturbances?

4.4.2 Analysis of SOILC

In this section the SOILC approach is analyzed with respect to the three previous mentioned questions.

1. Convergence

From equation (4.12) and (4.13), we obtain

$$S_p(z)f_{k-1}(t) = T_{k-1}g(t) + h(t) - e_{k-1}(t), \quad (4.20)$$

$$S_p(z)f_k(t) = T_k g(t) + h(t) - e_k(t). \quad (4.21)$$

In (4.16) substitute $f_{k+1}(t)$ by (4.19)

$$\begin{aligned} e_{k+1}(t) = & (1 - Q(z)L(z)S_p(z))((1 - \alpha)e_{k-1}(t) + \alpha e_k(t)) \\ & + (1 - Q(z))S_p(z)((1 - \alpha)f_{k-1}(t) + \alpha f_k(t)). \end{aligned}$$

and together with (4.20) and (4.21), we obtain

$$\begin{aligned} e_{k+1}(t) = & Q(z)(1 - L(z)S_p(z))(\alpha e_k(t) + (1 - \alpha)e_{k-1}(t)) \\ & + (1 - Q(z))(T_{k+1}g(t) + h(t)). \end{aligned} \quad (4.22)$$

To analyze the convergence of error, the system is constructed as a linear iterative system. From (4.22) it can be seen that $e_{k+1}(t)$ is related to $e_k(t)$ and $e_{k-1}(t)$. We define $\underline{x}_k(t) = (e_k(t) \quad e_{k-1}(t))^T$ and $\underline{u}_k(t) = T_{k+1}g(t) + h(t)$ such that

$$\underline{x}_{k+1}(t) = A(z)\underline{x}_k(t) + B(z)\underline{u}_k(t). \quad (4.23)$$

with

$$A(z) = \begin{pmatrix} a_1(z) & a_2(z) \\ 1 & 0 \end{pmatrix}, \quad B(z) = \begin{pmatrix} 1 - Q(z) \\ 0 \end{pmatrix}, \quad (4.24)$$

where $a_1(z) = Q(z)(1 - L(z)S_p(z))\alpha$ and $a_2(z) = Q(z)(1 - L(z)S_p(z))(1 - \alpha)$. Convergence of this linear iterative system is assessed in the frequency domain using the work of (Norrlöf, 2000). Transforming the linear iterative system to the frequency domain leads to

$$\underline{X}_{k+1}(\omega) = A(e^{j\omega})\underline{X}_k(\omega) + B(e^{j\omega})\underline{U}_k(\omega), \quad (4.25)$$

where the signals $\underline{x}_k(t)$ and $u_k(t)$ are transformed to the frequency domain using

$$X(\omega) = \sum_{l=0}^{\infty} x(l)e^{-j\omega l}. \quad (4.26)$$

Convergence is now obtained if

$$\bar{\rho} = \sup_{\omega \in [0, \pi]} \rho(A(e^{j\omega})) < 1. \quad (4.27)$$

with $\rho(A(e^{j\omega}))$ denoting the spectral radius of $A(e^{j\omega})$ defined as

$$\rho(A(e^{j\omega})) = \max_{i=\{1,2\}} |\lambda_i(A(e^{j\omega}))|. \quad (4.28)$$

The eigenvalues of the matrix $A(e^{j\omega})$ are given by

$$\lambda_{1,2}(e^{j\omega}) = \frac{a_1(e^{j\omega}) \pm \sqrt{a_1(e^{j\omega})^2 + 4a_2(e^{j\omega})}}{2}. \quad (4.29)$$

Therefore, convergence of the linear iterative system (4.23) is guaranteed if the condition

$$\left\| \frac{a_1(e^{j\omega}) \pm \sqrt{a_1(e^{j\omega})^2 + 4a_2(e^{j\omega})}}{2} \right\|_{\infty} < 1, \quad \forall \omega \quad (4.30)$$

is satisfied. Since the phase of $L(e^{j\omega})S_p(e^{j\omega})$ is zero for all frequencies and $Q(e^{j\omega})$ is a zero-phase low-pass filter, $a_1(e^{j\omega})$ and $a_2(e^{j\omega})$ are real-valued, $\forall \omega$. It will be shown that there is a trade-off between the designed $Q(z)$ and $L(z)$ filter and the maximum allowable values of α which guarantee convergence. In Fig. 4.9 the gray area indicates the allowable values of α for different values of $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_{\infty}$. From this figure we have the following observations:

- for convergence, $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_{\infty} \leq 1$ is necessary,
- if $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_{\infty} = 1$, then $0 \leq \alpha \leq 2$. However, from the definition of α , it is possible that α is negative depending on the values of the gains, and
- the smaller the value of $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_{\infty}$, the larger the range of possible values of α .

Note that the presented convergence analysis is conservative, since it is expected that the error in the next trial is always smaller than the current error, *irrespective* of the applied gain with which the setpoint is scaled. If the setpoint in the next trial is obtained by scaling with a larger gain than the current setpoint, the error in the next trial is also expected to be larger than the current error. As a result it is harder to always obtain a smaller error in the next trial than the error in the current trial.

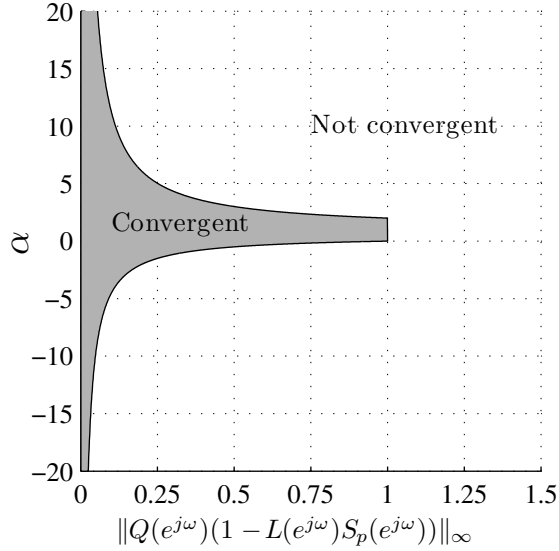


Figure 4.9: Computed α values based on $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_{\infty}$ that make the system convergent.

2. Perfect pitch

If in the previous two trials, the gains T_k and T_{k-1} are the same, by definition, the value of α becomes infinity. Hence, if $\|Q(e^{j\omega})(1 - L(e^{j\omega})S_p(e^{j\omega}))\|_{\infty} > 0$, then convergence can not be guaranteed. This is caused by the fact that we can not estimate $g(t)$ and $h(t)$ after two trials with the same setpoint. Improvements for this case will be discussed in the next Section.

3. Type $\textcircled{\text{III}}$ disturbances

If T_{k-1} is close to T_k then type $\textcircled{\text{III}}$ disturbances highly affect the estimations of $g(t)$ and $h(t)$. This is explained as follows. By considering sensor noise n_k in Fig. 4.8 the errors $e_{k-1}(t)$ and $e_k(t)$ can be written as

$$e_{k-1}(t) = T_{k-1}g(t) + h(t) - S_p(z)f_{k-1}(t) - S(z)n_{k-1}(t), \quad (4.31)$$

$$e_k(t) = T_k g(t) + h(t) - S_p(z)f_k(t) - S(z)n_k(t). \quad (4.32)$$

Substitution of (4.31) and (4.32) into (4.14) and (4.15) leads to

$$\tilde{g}(t) = g(t) - S(z) \underbrace{\frac{n_{k-1}(t) - n_k(t)}{T_{k-1} - T_k}}_{\text{estimation error}}, \quad (4.33)$$

$$\tilde{h}(t) = h(t) - S(z) \underbrace{\frac{T_{k-1}n_k(t) - T_k n_{k-1}(t)}{T_{k-1} - T_k}}_{\text{estimation error}}. \quad (4.34)$$

Therefore, if T_k is close to T_{k-1} , there will be large estimation errors, since the noise terms are amplified. In this work, we will deal with this sensor noise by iteratively estimating $g(t)$ and $h(t)$ such that these noise terms will not be amplified. This is done by introducing an adaptive low-pass filter in the trial domain on the estimates of $g(t)$ and $h(t)$.

4.4.3 Improving SOILC

In this section, we will improve the principle of SOILC with respect to 1) sensor noise and 2) for cases in which $T_k = T_{k-1}$. By introducing an adaptive low-pass filter in the trial domain, SOILC is first made less sensitive to sensor noise. Incorporating sensor noise $n_k(t)$ in the previous analysis leads to the update law,

$$\begin{aligned} f_{k+1}(t) = & Q(z) \left((1 - \alpha)f_{k-1}(t) + \alpha f_k(t) + \right. \\ & L(z) \left((1 - \alpha)e_{k-1}(t) + \alpha e_k(t) + \right. \\ & \left. \left. S(z)(1 - \alpha)n_{k-1}(t) + S(z)\alpha n_k(t) \right) \right). \end{aligned} \quad (4.35)$$

If the previous two gains, T_k and T_{k-1} are close to each other, the absolute value of α can be large. Hence, the sensor noise is amplified by a large gain and becomes part of the next feedforward signal $f_{k+1}(t)$, which is not desired and may cause a large error.

In the trial domain, the random type $\textcircled{\text{III}}$ disturbances, like sensor noise, are changing from trial to trial, while the repetitive type $\textcircled{\text{II}}$ disturbances remain the same. Therefore, the sensor noise can be seen as a high-frequency signal in the trial domain, while the repetitive disturbances can be seen as a low-frequency signal in the trial domain. This implies that we can use a low-pass filter in the trial domain to reject the sensor noise. We use SOILC with an adaptive low-pass filter in the

trial domain to smoothen out the sensor noise. In this way, the estimations of $g(t)$ and $h(t)$ are obtained iteratively and filtered, and are then used in the generation of the new feedforward signal $f_{k+1}(t)$. Define the terms $g(t)$ and $h(t)$ as the true values and $\tilde{g}_k(t)$ and $\tilde{h}_k(t)$ to represent the corresponding estimations after trial k :

$$\tilde{g}_k(t) = \frac{e_{k-1}(t) - e_k(t)}{T_{k-1} - T_k} + S_p(z) \frac{f_{k-1}(t) - f_k(t)}{T_{k-1} - T_k}, \quad (4.36)$$

$$\tilde{h}_k(t) = \frac{T_{k-1}e_k(t) - T_k e_{k-1}(t)}{T_{k-1} - T_k} + S_p(z) \frac{T_{k-1}f_k(t) - T_k f_{k-1}(t)}{T_{k-1} - T_k}. \quad (4.37)$$

The first order adaptive low-pass filters in the trial domain are chosen as

$$\hat{g}_{k+1}(t) = (1 - \gamma_k)\hat{g}_k(t) + \gamma_k\tilde{g}_k(t), \quad (4.38)$$

$$\hat{h}_{k+1}(t) = (1 - \gamma_k)\hat{h}_k(t) + \gamma_k\tilde{h}_k(t), \quad (4.39)$$

where $\hat{g}_{k+1}(t)$ and $\hat{h}_{k+1}(t)$ are the low-pass filtered outputs of the estimations, which are going to be used in the update law. The value of γ_k can be tuned to give a weighting on how much the current estimates of $g(t)$ and $h(t)$ are used for the construction of the new feedforward signal. Using the trial shift operator w , these low-pass filters can also be written as

$$\hat{g}_k(t) = \frac{\gamma_k}{w - (1 - \gamma_k)} \tilde{g}_k(t), \quad (4.40)$$

$$\hat{h}_k(t) = \frac{\gamma_k}{w - (1 - \gamma_k)} \tilde{h}_k(t). \quad (4.41)$$

For stability of these low-pass filters, it is required that $0 \leq \gamma_k \leq 1$. To prevent sensor noise amplification when $T_k \approx T_{k-1}$, we choose γ_k as

$$\gamma_k = \beta |T_{k-1} - T_k|, \quad (4.42)$$

where now the scalar β should satisfy

$$0 \leq \beta \leq \frac{1}{\bar{T} - \underline{T}}. \quad (4.43)$$

By choosing $\gamma_k = \beta |T_{k-1} - T_k|$ we cancel out the denominator $(T_{k-1} - T_k)$ in (4.33) and (4.34), so only $\beta S(z)(n_{k-1} - n_k)$ and $\beta S(z)(T_{k-1}n_k - T_k n_{k-1})$ are considered in the estimations of g and h . A trade-off is present in this case between convergence speed and sensitivity to random disturbances, as is also discussed in (Bristow, 2008). A larger value of β results in faster convergence but results in a system that is more sensitive to noise, whereas a lower value of β leads to a slower convergence but results in a system that is less sensitive to noise.

In case $T_{k-1} = T_k$, the value of γ_k becomes zero. As a result the estimates $\hat{g}_{k+1}(t)$ and $\hat{h}_{k+1}(t)$ are not updated and equal the previous ones $\hat{g}_k(t)$ and $\hat{h}_k(t)$ such that no learning is performed. For the cases where $T_{k-1} = T_k$ and learning is to be performed in trial $k+1$ with $T_{k+1} = T_k$ two options are considered.

- First, in case $T_{k-1} = T_k$ a standard ILC update could be applied for trial $k+1$, which updates the feedforward signal and decreases the next tracking error. The update law is in that case given by $f_{k+1}(t) = Q(z)(f_k(t) + L(z)e_k(t))$. A disadvantage however is that the estimates \hat{g} and \hat{h} are not updated while standard ILC is applied. Whenever future gain values differ from T_k such that SOILC can be applied again, the error might increase significantly, since old and possibly non-converged values of \hat{g}_{k+1} and \hat{h}_{k+1} are used.
- Second, in order to keep learning *and* update \hat{g}_{k+1} and \hat{h}_{k+1} while $T_{k+1} = T_k$ we propose the following. Instead of using information of the previous two trials k and $k-1$ to update \hat{g}_{k+1} and \hat{h}_{k+1} in SOILC, we use the information of the previous trial k and trial $k-p$, where $p \geq 1$ is the smallest number for which $T_{k-p} \neq T_k$ to update \hat{g} and \hat{h} . Therefore the update can be written as

$$\hat{g}_{k+1}(t) = (1 - \gamma_k)\hat{g}_k(t) + \gamma_k\tilde{g}_k(t), \quad (4.44)$$

$$\hat{h}_{k+1}(t) = (1 - \gamma_k)\hat{h}_k(t) + \gamma_k\tilde{h}_k(t), \quad (4.45)$$

with now $\gamma_k = \beta|T_{k-p} - T_k|$ and

$$\tilde{g}_k(t) = \frac{e_{k-p}(t) - e_k(t)}{T_{k-p} - T_k} + S_p(z) \frac{f_{k-p}(t) - f_k(t)}{T_{k-p} - T_k}, \quad (4.46)$$

$$\tilde{h}_k(t) = \frac{T_{k-p}e_k(t) - T_k e_{k-p}(t)}{T_{k-p} - T_k} + S_p(z) \frac{T_{k-p}f_k(t) - T_k f_{k-p}(t)}{T_{k-p} - T_k}. \quad (4.47)$$

As an example consider the gains depicted in Fig. 4.10. The gains are different for each iteration except for iteration five to eight where the gains are the same. Hence for $k = 6, 7$ and 8 the value $T_{k-1} - T_k = 0$. Therefore for iteration $k = 6, 7$ and 8 , the values of p in (4.46) and (4.47) are $2, 3$ and 4 , respectively. For iteration $k = 2, 3, 4, 5, 9$ and 10 the value $T_{k-1} - T_k \neq 0$. Therefore the value of p in (4.46) and (4.47) is taken as one, such that it results in its original form of (4.36) and (4.37).

The error in trial $k+1$ can now be estimated as

$$e_{k+1}(t) = T_{k+1}\hat{g}_{k+1}(t) + \hat{h}_{k+1}(t) - S_p f_{k+1}(t). \quad (4.48)$$

Since the goal is to design a feedforward signal $f_{k+1}(t)$ such that $e_{k+1}(t) = 0$, we derive the new update law:

$$f_{k+1}(t) = L(z)(T_{k+1}\hat{g}_{k+1}(t) + \hat{h}_{k+1}(t)). \quad (4.49)$$

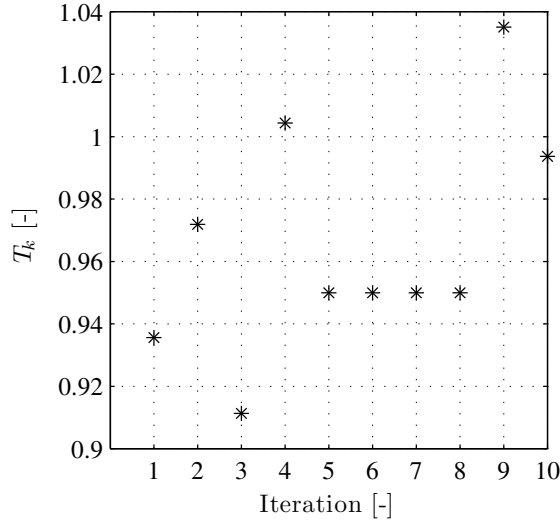


Figure 4.10: Example of possible gains. The gains in iteration 5 to 8 are the same, while others are different each trial.

By successive substitution we can write \hat{g}_{k+1} as

$$\hat{g}_{k+1}(t) = (1 - \gamma_k) \hat{g}_k(t) + \gamma_k \frac{e_{k-1}(t) - e_k(t)}{T_{k-1} - T_k} + \gamma_k \frac{T_{k-1} \hat{g}_{k-1}(t) + \hat{h}_{k-1}(t) - T_k \hat{g}_k(t) - \hat{h}_k(t)}{T_{k-1} - T_k}.$$

With the definition of γ_k this leads to

$$\hat{g}_{k+1}(t) = (1 - \beta |T_{k-1} - T_k|) \hat{g}_k(t) + \beta \cdot \text{sign}(T_{k-1} - T_k) \times (e_{k-1}(t) - e_k(t) + T_{k-1} \hat{g}_{k-1}(t) + \hat{h}_{k-1}(t) - T_k \hat{g}_k(t) - \hat{h}_k(t)). \quad (4.50)$$

Similarly,

$$\hat{h}_{k+1}(t) = (1 - \beta |T_{k-1} - T_k|) \hat{h}_k(t) + \beta \cdot \text{sign}(T_{k-1} - T_k) \times (T_{k-1} e_k(t) - T_k e_{k-1}(t) + T_{k-1} (T_k \hat{g}_k(t) + \hat{h}_k(t)) - T_k (T_{k-1} \hat{g}_{k-1}(t) + \hat{h}_{k-1}(t))). \quad (4.51)$$

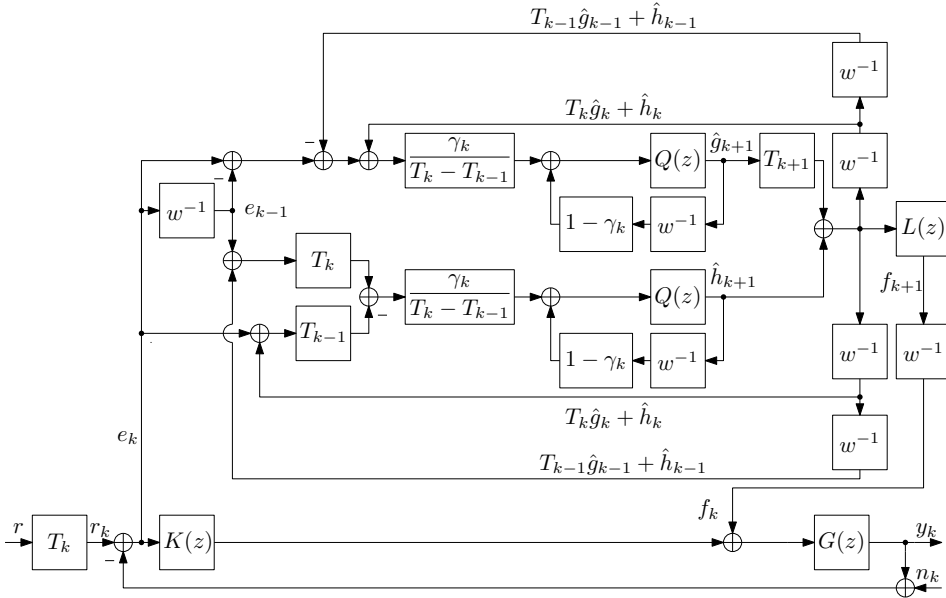


Figure 4.11: Second order ILC with low-pass filters in the trial domain.

To filter out the high frequency components in the measured error, we still use a zero-phase low-pass filter $Q(z)$ as the robustness filter after the updated estimations $\hat{g}_{k+1}(t)$ and $\hat{h}_{k+1}(t)$. Therefore, the first order low-pass filters in the trial domain (4.38) and (4.39) now also include the low-pass filtering in frequency domain, i.e., the new filters now are

$$\hat{g}_{k+1}(t) = Q(z) \left((1 - \gamma_k) \hat{g}_k(t) + \gamma_k \tilde{g}_k(t) \right), \quad (4.52)$$

$$\hat{h}_{k+1}(t) = Q(z) \left((1 - \gamma_k) \hat{h}_k(t) + \gamma_k \tilde{h}_k(t) \right). \quad (4.53)$$

Fig. 4.11 shows the block diagram of the second order ILC with low-pass filters.

4.5 Results

In this section the performance of 1) standard ILC, 2) normalized ILC (NILC), 3) second order ILC (SOILC) and 4) SOILC with an adaptive low-pass filter in the trial domain for scale varying setpoints will be compared.

The proposed methods are validated on an industrial application, being a xy-wafer stage, where the task is to move from one discrete semiconductor to the next, see Fig. 4.5. The frequency response function (FRF) from the input of the motor to the metric position output measured by the camera is given in Fig. 4.12. The plant is modeled by a mass-damper system with delay. The obtained model, with as input the applied voltage to the current amplifier and with output the metric position obtained from the camera, given by

$$G(z) = 1 \times 10^{-7} \cdot \frac{1.34z^2 + 5.14z + 1.23}{z^4 - 1.85z^3 + 0.85z^2}, \quad (4.54)$$

is also shown in Fig. 4.12 by its FRF and shows a good match until approximately 60 Hz. As a consequence a mismatch between the measured process sensitivity and its model is expected after 60 Hz. Therefore the Q filter gets a cutoff frequency of 50 Hz. A feedback controller $K(z)$ is tuned which consists of a lead filter with a zero at 6 Hz and a pole at 100 Hz and a second order low-pass filter with a cutoff frequency of 250 Hz and a damping of 0.6. Finally, a notch is added at 80 Hz. The discrete controller is given by

$$K(z) = 1 \times 10^4 \cdot \frac{3.3z^5 - 2.3z^4 - 7.8z^3 + 11z^2 - 3.5z - 0.82}{z^5 - 2.4z^4 + 2.4z^3 - 1.2z^2 + 0.38z - 0.070}. \quad (4.55)$$

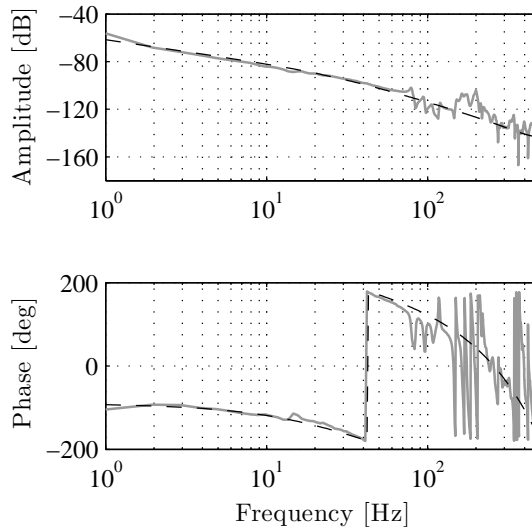


Figure 4.12: Measurement frequency response function and the corresponding fit of the plant: measurement data (gray), fitted model (dashed black).

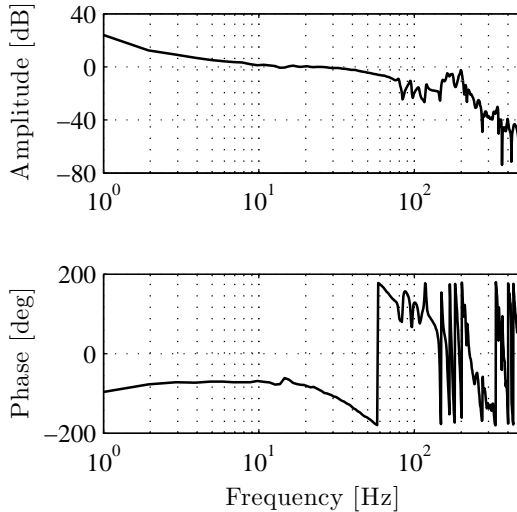


Figure 4.13: Open-loop response of xy-wafer stage.

This leads to an open-loop frequency response as shown in Fig. 4.13. From this figure it can be seen that the obtained cross-over frequency, is 20 Hz.

The number of iterations that will be performed is 30. For the sake of comparison, the arbitrary gains are chosen the same for the four methods. During experiments, the lower bound for T_k will be $\underline{T} = 0.75$, whereas the upper bound for T_k will be $\overline{T} = 1.25$ such that $T_k \in [0.75, 1.25]$. Fig. 4.14 shows the applied gains. The proposed methods are applied on the xy-wafer stage with the value of β chosen as 1 in this case, such that $0 \leq \beta \leq \frac{1}{\overline{T}-\underline{T}} = 2$ is satisfied. The maximum errors for each iteration are given in Fig. 4.15. Since standard ILC does not incorporate the scaling of the setpoint it is expected that the final error oscillates depending on the applied gains, which also shows in Fig. 4.15. Normalized ILC does incorporate the scaling of the setpoint, however it is based on the absence of type $\textcircled{\text{II}}$ disturbances. Dry friction is one of the major disturbances present in the experimental setup. It can be seen that if two successive gains are quite different the error of normalized ILC increases, which is caused by the inappropriate scaling of the error; the error due to the dry friction is also scaled, which in practice remains approximately the same. Second order ILC *without* low-pass filters in the trial domain suffers from

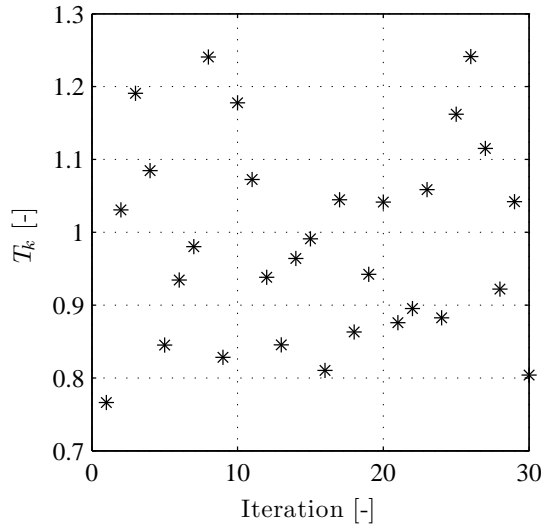


Figure 4.14: Gains applied during iterations on the xy-wafer stage.

the fact that α will become large if $T_{k-1} - T_k$ is small, resulting in an increase of the errors due to sensor noise amplification. This is the case for iteration 16, where the feedforward update (4.19) is dependent on iterations 14 and 15, which are close to each other, see Fig. 4.14. The same reasoning holds for iteration 23. The most satisfactory results are obtained using second order ILC with an adaptive low-pass filter in the trial domain. After eleven iterations the error is converged to maximum errors of less than $5 \mu\text{m}$, while different setpoints are applied during the learning process.

The proposed solution in case $T_{k-1} = T_k$ in second order ILC with adaptive low-pass filtering in the trial domain is investigated next. During the iterations the applied gains in this case are the same as in Fig. 4.14, except that the gains of iterations eleven through twenty are kept the same in this case and equal to the gain of iteration eleven. Furthermore, the value of β is taken as 0.5, such that learning is slower. If $\beta = 1$ we saw in the previous results that the error and therefore also the estimates \hat{g} and \hat{h} already converged within eleven iterations. The effect that we want to visualize here is that learning is still present even when the gain values of two successive iterations are the same. The results are given in Fig. 4.16. The gray hatched area indicates that the gains are the same

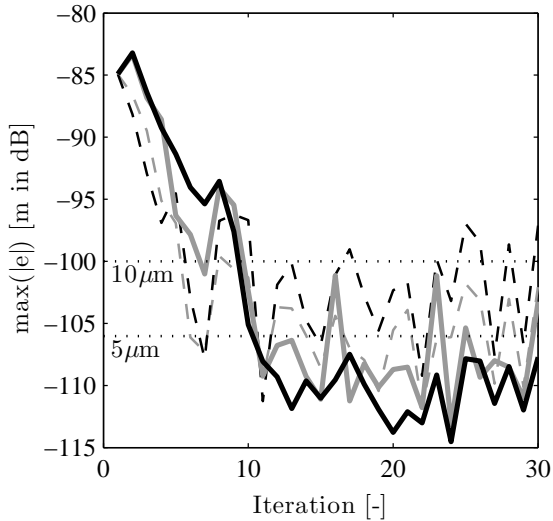


Figure 4.15: Maximum absolute error in meters for different methods on xy-wafer stage, dashed gray: standard ILC, dashed black: NILC, gray: SOILC, **bold**: SOILC with an adaptive low-pass filter in the trial domain.

for these iterations. A first observation is that the error converges slower. This was expected due to the lower value of β . Second, the error converges even when the gains of iterations eleven through twenty are the same. Furthermore, after iteration twenty, when the gains deviate again, learning is still present as can be seen by the further reduction of the error. This result shows that the proposed solution in the case where two successive gains are the same is effective.

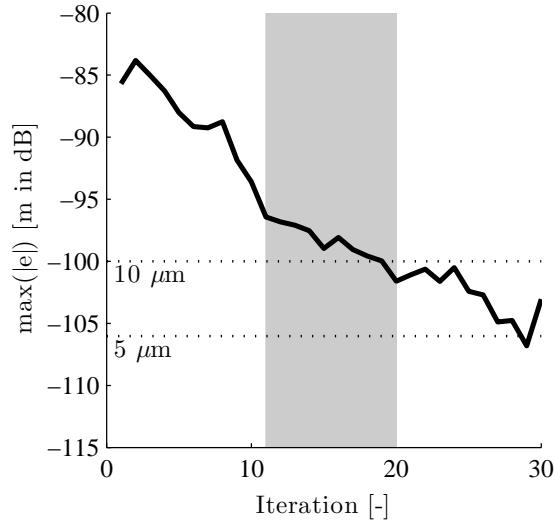


Figure 4.16: Maximum absolute error in meters for different methods on xy-wafer stage, **bold**: SOILC with an adaptive low-pass filter in the trial domain with $\beta = 0.5$. The gain values of iteration eleven through twenty (depicted by the hatched area) are the same but learning is still present.

4.6 Conclusions

Three methods, NILC, SOILC and SOILC with an adaptive low-pass filter in the trial domain have been investigated in this work to handle scale varying setpoints in iterative learning control. Experiments with an industrial setup are carried out to validate these methods. NILC achieves a good performance when there is no disturbance at all. SOILC is sensitive to non-repetitive noise when the previous applied setpoints are almost the same. SOILC with an adaptive low-pass filter in the trial domain can handle the situation when both repetitive disturbances and non-repetitive noise exist and achieves a good performance. After convergence the error is reduced to less than $5 \mu\text{m}$ for an industrial xy-wafer stage application. The investigated methods consider disturbances that experience the same scaling as the setpoint, and trial independent repetitive disturbances. Another class of disturbances known in practice are position dependent disturbances, e.g., cogging.

This kind of disturbances cannot be handled in the presented work, since scaling cannot be applied. Incorporating these kinds of disturbances in the proposed method will therefore be subject for future research.

Chapter 5

Conclusions and recommendations

I*N this chapter, the main conclusions of this work are given. Furthermore, recommendations for future work on motion control for near-repetitive structures are given.*

5.1 Conclusions

The problem statement of this work was to investigate control design approaches for the relative positioning of a tool with respect to a feature of a near-repetitive structure. Regarding this problem statement the conclusions of this work are as follows.

Visual servoing enables a direct relative position measurement of the tool with respect to the features of the repetitive structure, in contrast with an indirect relative position measurement with on board motor encoders which depends on machine properties such as geometric accuracy, stiffness of the machine frame and thermal expansion. Therefore, using this direct relative position measurement, the dynamics of interest can be accounted for in the control design. Moreover, disturbances that are unobservable to on board motor encoders are measured and accounted for in a closed-loop direct visual servoing control approach.

In case of near-repetitive structures the exact metric position of a feature with respect to the tool cannot be predicted. As a consequence, it is impossible to

generate a metric setpoint trajectory for a feature-to-feature motion offline. The introduction of feature-based control has solved this problem. As a result of the proposed feature-based measurement the gain of the controlled system will vary with the momentary pitch. The proposed a posteriori stability analysis can assess stability of the closed-loop system for bounded pitch variations. The use of piecewise second order interpolation in the feature domain offers a higher positioning accuracy as compared to the bilinear interpolation, due to the elimination of transient responses at the feature frame transitions.

Common tasks in the production of repetitive structures consist of a global feature-to-feature movement followed by a local relative feature movements. The feature-based motion control approach is designed such that both feature-to-feature movements and relative feature movements can be carried out by combining the feature domain with the metric domain. Operators can intuitively program to which feature to go to and what metric movements have to be carried out with respect to that specific feature. The combination of feature-to-feature movements and relative feature movements has been validated on an industrial applications in which discrete semiconductors on a wafer are positioned with respect to the camera with accuracies of less than 10 μm .

For feature-to-feature movements ILC techniques are investigated in this work. Standard ILC can only be used in cases where the applied setpoint is strictly repetitive for every trial. In this work however the distance between successive features is prone to pitch variations, such that standard ILC cannot be used. The introduction of second order ILC handles this constraint and shows that the tracking error for scale varying setpoints is reduced iteratively, while different setpoints are applied during the learning process. The error in this work is assumed to be consisting of 1) scale varying disturbances, 2) repetitive disturbances and 3) measurement noise. The presented method is experimentally validated on the industrial wafer stage application and shows that it successfully compensates for the first two classes of disturbances resulting in errors of less than 5 μm .

An experimental visual servoing setup was created using a commercially available off-the-shelf camera on top of an industrial xy-wafer stage. The setup is capable of running at a frame rate of 1 kHz with a delay of 2.5 ms. The high frame rate is obtained by reading out only a part of the sensor. The delay is reduced, by 1) adapting the image size to match the possible interface transfer rate, 2) using a priori knowledge about the repetitive structure such as the nominal pitch between features and the size and shape of the features, 3) online predicting the pixel positions of the features and iv) illuminating the substrate with power LEDs to reduce the exposure time of the camera.

5.2 Recommendations

The introduction of feature-based positions directly results in non-linear system behavior, since pitch variations cause the system gain to vary. In this work a stability analysis is carried out to prove robustness with respect to this matter. To even further increase the stability margins of the closed-loop system, it is recommended to investigate the use of feedback linearization or gain scheduling techniques, to compensate for the gain variation in the controller. At the moment the pixel positions of the features are known, the momentary gain of the system is known and can be incorporated by its inverse in the controller to obtain a nominal open-loop.

Next to bilinear feature interpolation, in this work we applied second order feature interpolation to successfully reduce transient responses at feature frame transitions. The proposed method is however not restricted to second order interpolation. High order interpolation can be used to generate continuous feature-based accelerations or even feature-based jerks at the feature frame transitions. The design choices for these higher order interpolations directly affect the gain of the system. On the one hand the feature-based position can be made more smooth across feature frame transitions, however at the same time due to the varying gain of the system the stability should be considered. Therefore, it is recommended to investigate the trade-off between smoothen the feature-based position and the robust stability of the closed-loop system.

The presented feature-based control is able to deal with pitch imperfections between successive features, i.e., variations in the positions of the features. Rotation imperfections of each feature are not considered. By detecting the orientation of the feature next to its position, and by assuming all features are identical, the complete contour of each feature can be calculated and used for relative feature movements. Therefore, it is recommended to investigate the incorporation of the orientation of the individual features next to the position in the feature-based control approach.

The proposed second order ILC approach for scale varying setpoints compensates for disturbances that experience the same scaling as the setpoint, and trial independent repetitive disturbances. Another class of disturbances known in practice are position dependent disturbances, e.g., cogging. These kind of disturbances are not incorporated in the presented ILC techniques, since scaling cannot be applied. Incorporating these kinds of disturbances in the proposed methods is therefore recommended, where the problem is to identify position dependent disturbances and non-position dependent disturbances during iterations.

By using second order ILC we are able to estimate disturbances 1) which experience the same scaling as the setpoint and 2) which are repetitive every trial. Higher order ILC can be introduced to even detect other disturbance classes which are directly related to the gain of the applied reference. An example of such a disturbance can be air resistance. This disturbance is typically dependent on the quadratic velocity, i.e., $d_k(t) = (T_k d(t))^2$. By using third order ILC, next to the repetitive and scale varying disturbances even these disturbances can be identified and compensated.

For systems with multiple inputs and outputs it is recommended to investigate if the presented second order ILC approach is also applicable. In that case not only does the reference scale but also has a varying input direction, which should be taken into account.

One assumption that remains in the second order ILC approach is that the initial state of the system should be equal for every iteration. Repetitive control is not restricted by this constraint. Therefore, a natural research question for future work is if the proposed second order ILC approach can be extended towards a high order repetitive control architecture, in which scale varying setpoints can be applied in a more continuous manner, without having to reinitialize the system at the beginning of every iteration.

In this work visual servoing is used for positioning the features of the repetitive structure with respect to the tool. The focus of this work was on the measurement concept and control design where the attainable bandwidth was limited by the flexibilities of the system in this case. For systems with higher eigenfrequencies, the attainable bandwidth might eventually be hampered by the delay of the system, which in this case is 2.5 ms. In those cases, the delay should be reduced. It is then recommended to investigate the availability of high speed vision sensors. Moreover, the interfacing between the camera and image processing unit for the image transfer is a bottleneck. A processing unit directly next to the image sensor such as field-programmable gate arrays (FPGA) can significantly reduce the delay of the image transfer. Moreover, the parallel processing capabilities enables possibilities for faster image processing.

Bibliography

- Ahn, H. S., Chen, Y. Q., and Moore, K. L. (2007). Iterative learning control: Brief survey and categorization. *IEEE Trans. Syst., Man, Cybern. C*, 37(6):1099–1121.
- van Assen, H. C., Egmont-Petersen, M., and Reiber, J. H. C. (2002). Accurate object localization in gray level images using the center of gravity measure; accuracy versus precision. *IEEE Trans. Image Process.*, 11(12):1379–1384.
- Bazaz, S. A. and Tondu, B. (1999). Minimum time on-line joint trajectory generator based on low order spline method for industrial manipulators. *Robotics and Autonomous Systems*, 29(4):257–268.
- de Best, J., van de Molengraft, M. J. G., and Steinbuch, M. (2009). Direct dynamic visual servoing at 1 kHz by using the product as encoder. In *IEEE Proc. Conf. on Contr. and Autom.*, pages 361–366.
- de Best, J., van de Molengraft, M. J. G., and Steinbuch, M. (2011a). High speed visual motion control applied to products with repetitive structures. *Accepted for publication in IEEE Trans. Control Syst. Technol.*
- de Best, J., van de Molengraft, M. J. G., and Steinbuch, M. (2011b). Planar feature-based motion control for near-repetitive structures. *Submitted for publication.*
- de Best, J., van de Molengraft, M. J. G., and Steinbuch, M. (2011c). Second order iterative learning control for scale varying setpoints. *Submitted for publication.*
- Bien, Z. and Huh, K. M. (1989). Higher-order iterative learning control algorithm. *IEE Proc. Part D Contr. Theory & Appl.*, 136(3):105–112.
- Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA.

- Brier, P., Steinbuch, M., and Jonker, P. (2007). Low latency 2d position estimation with a line scan camera for visual servoing. In *Proc. Conf. Adv. Concepts for Intell. Vis. Syst.*, volume 1, pages 37–47.
- Bristow, D. (2008). Frequency domain analysis and design of iterative learning control for systems with stochastic disturbances. In *IEEE Proc. Amer. Contr. Conf.*, pages 3901–3907.
- Bristow, D. A., Tharayil, M., and Alleyne, A. G. (2006). A survey of iterative learning control. *IEEE Contr. Syst. Mag.*, 26(3):96–114.
- Broquère, X., Sidobre, D., and Herrera-Aguilar, I. (2008). Soft motion trajectory planner for service manipulator robot. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, pages 2808–2813.
- Chaumette, F. and Hutchinson, S. (2006). Visual servo control part 1: Basic approaches. *IEEE Robot. Autom. Mag.*, 13(4):82–90.
- Chaumette, F. and Hutchinson, S. (2007). Visual servo control part 2: Advanced approaches. *IEEE Robot. Autom. Mag.*, 14(1):109–118.
- Chaumette, F., Rives, P., and Espiau, B. (2002). Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, pages 2248–2253.
- Chen, Y., Gong, Z., and Wen, C. (1998). Analysis of a high-order iterative learning control algorithm for uncertain nonlinear systems with state delays. *Automatica*, 34(3):345–353.
- Chen, Y., Sun, M., Huang, B., and Dou, H. (1992). Robust higher order repetitive learning control algorithm for tracking control of delayed repetitive systems. In *IEEE Proc. Conf. on Dec. & Contr.*, volume 3, pages 2504–2510.
- Chen, Y., Wen, C., and Sun, M. (1997). A robust high-order p-type iterative learning controller using current iteration tracking error. *Int. J. of Contr.*, 68(2):331–342.
- Chesi, G., Hashimoto, K., Prattichizzo, D., and Vicino, A. (2004). Keeping features in the field of view in eye-in-hand visual servoing: A switching approach. *IEEE Trans. Robot.*, 20(5):908–914.
- Corke, P. I. (1995). Dynamic issues in robot visual-servo systems. In *Proc. Int. Symp. on Robot. Research*, pages 488–498.
- Corke, P. I. (2001). *Visual Control of Robot Manipulators, A Review*, volume 7, pages 1–32. World Scientific, Singapore.

- Corke, P. I. and Good, M. C. (1992). Dynamic effects in high-performance visual servoing. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, volume 2, pages 1838–1843.
- Corke, P. I. and Good, M. C. (1996). Dynamic effects in visual closed-loop systems. *IEEE Trans. Robot. Autom.*, 12(5):671–683.
- Corke, P. I. and Hutchinson, S. A. (2001). A new partitioned approach to image-based visual servo control. *IEEE Trans. Robot. Autom.*, 17(4):507–515.
- Crétual, A. and Chaumette, F. (1997). Positioning a camera parallel to a plane using dynamic visual servoing. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, volume 1, pages 43–48.
- Espiau, B., Chaumette, F., and Rives, P. (1992). A new approach to visual servoing in robotics. *IEEE Trans. Robot. Autom.*, 8(3):313–326.
- Feddema, J. T. and Mitchell, O. R. (1989). Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. Robot. Autom.*, 5(5):691–700.
- Franklin, G. F., Powell, J. D., Emami-Naeini, A., and Powell, J. D. (1994). *Feedback control of dynamic systems*, volume 3. Addison-Wesley Reading (Ma) etc.
- Frost, A. R., Street, M. J., and C., H. R. (1993). The development of a pneumatic robot for attaching a milking machine to a cow. *Mechatronics*, 3(4):409–418.
- Gahinet, P., Nemirovski, A., Laub, A., and Chilali, M. (1994). The LMI control toolbox. In *IEEE Proc. Conf. on Dec. & Contr.*, volume 3, pages 2038–2041.
- Gangloff, J. and de Mathelin, M. (2002). High speed visual servoing of a 6 dof manipulator using mimo predictive control. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, volume 4, pages 3751–3756.
- Gangloff, J. and de Mathelin, M. (2003). High-speed visual servoing of a 6-dof manipulator using multivariable predictive control. *Advanced Robotics*, 17(10):993–1021.
- Gonzalez, R. C. and Woods, R. E. (2008). *Digital Image Processing*. Pearson Prentice hall, 3rd edition.
- Hara, S., Yamamoto, Y., Omata, T., and Nakano, M. (1988). Repetitive control system: A new type servo system for periodic exogenous signals. *IEEE Trans. Autom. Control*, 33(7):659–668.
- Haschke, R., Weitnauer, E., and Ritter, H. (2008). On-line planning of time-optimal, jerk-limited trajectories. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, pages 3248–3253.

- Hashimoto, K. (2003). A review on vision-based control of robot manipulators. *Advanced Robotics*, 17(10):969–991.
- Hashimoto, K., Kimoto, T., Ebine, T., and Kimura, H. (1991). Manipulator control with image-based visual servo. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, volume 3, pages 2267–2271.
- Heertjes, M. F. and Van de Molengraft, M. J. G. (2009). Set-point variation in learning schemes with applications to wafer scanners. *Control Engineering Practice*, 17(3):345–356.
- Heikkila, J. and Silven, O. (2002). A four-step camera calibration procedure with implicit image correction. In *IEEE proc. Conf. on Comp. Vision and Pattern Recog.*, pages 1106–1112.
- Hill, J. and Park, W. (1979). Real time control of a robot with a mobile camera. In *Proc. Int. Symp. on Indus. Rob.*, pages 233–246.
- Hoelzle, D., Alleyne, A., Johnson, W., et al. (2010). Basis task approach to iterative learning control with applications to micro-robotic deposition. *IEEE Trans. Control Syst. Technol.*, (99):1–11.
- Honderd, G., Jongkind, W., Klomp, C., Dessing, J., and Paliwoda, R. (1991). Strategy and control of an autonomous cow-milking robot system. *Robotics and Autonomous Systems*, 7(2–3):165–179.
- Hosoda, K. and Asada, M. (1994). Versatile visual servoing without knowledge of true jacobian. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, volume 1, pages 186–193.
- Huh, K. M. (1997). A study on the robustness of the higher-order iterative learning control algorithm. In *Proc. Asian Contr. Conf.*
- Hutchinson, S., Hager, G. D., and Corke, P. I. (1996). A tutorial on visual servo control. *IEEE Trans. Robot. Autom.*, 12(5):651–670.
- Hwang, J. H., Arkin, R. C., and Kwon, D. S. (2003). Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, volume 2, pages 1444–1449.
- Ishii, I., Nakabo, Y., and Ishikawa, M. (1996). Target tracking algorithm for 1ms visual feedback system using massively parallel processing. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, pages 2309–2314.
- Ishikawa, M., Morita, A., and Takayanagi, N. (1992). High speed vision system using massively parallel processing. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, pages 373–377.

- Jain, R., Kasturi, R., and Schunck, B. G. (1995). *Machine Vision*. McGraw-Hill, Inc.
- Jang, W. and Bien, Z. (2002). Feature-based visual servoing of an eye-in-hand robot with improved tracking performance. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, pages 2254–2260.
- Jansen, D. and Buttner, H. (2004). Real-time ethernet the EtherCAT solution. *IET Comput. & Contr. Eng.*, 15(1):16–21.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 82(Series D):35–45.
- Kalmár-Nagy, T., D’Andrea, R., and Ganguly, P. (2004). Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. *Robotics and Autonomous Systems*, 16(1):47–64.
- Kalmár-Nagy, T., Ganguly, P., and D’Andrea, R. (2002). Real-time trajectory generation for omnidirectional vehicles. In *IEEE Proc. Amer. Contr. Conf.*, pages 286–291.
- Kelly, R., Carello, R., Nasisi, O., Kuchen, B., and Reyes, F. (2000). Stable visual servoing of camera-in-hand robotic systems. *IEEE/ASME Trans. Mechatronics*, 5(1):39–48.
- Kim, Y. T., Lee, H., Noh, H. S., and Bien, Z. (2003). Robust higher-order iterative learning control for a class of nonlinear discrete-time systems. *IEEE Proc. Conf. on Syst., Man, Cybern.*, 3:2219–2224.
- Kitano, H., Asada, M., Noda, I., and Matsubara, H. (2002). RoboCup: Robot world cup. *IEEE Robot. Autom. Mag.*, 5(3):30–36.
- Kolloor, S. and lalmurugan, S. (1965). Machine vision for alignment and inspection in die bonder. In *SPIE Proc. Automatic Inspection Systems I*, volume 3185, pages 2–10.
- Komuro, T., Iwashita, A., and Ishikawa, M. (2009). A qvga-size pixel-parallel image processor for 1,000-fps vision. *IEEE Micro*, 29(6):58–67.
- Kragic, D. and Christensen, H. I. (2002). Survey on visual servoing for manipulators. Technical Report ISRN KTH/NA/P-02/01-SE, Computational Vision and Active Perception Laboratory, Royal Institute of Technology, Stockholm, Sweden.
- Kröger, T., Tomiczek, A., and Wahl, F. M. (2006). Towards on-line trajectory computation. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, pages 736–741.

- Kröger, T. and Wahl, F. M. (2010). Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events. *IEEE Trans. Robot.*, 26(1):94–111.
- Kuglin, C. D. and Hines, D. C. (1975). The phase correlation image alignment method. In *IEEE proc. Conf. on Cybern. & Society*, pages 163–165.
- Kuo, W. M., Chuanga, S. F., Nian, C. Y., and S., T. Y. (2008). Precision nano-alignment system using machine vision with motion controlled by piezoelectric moto. *Mechatronics*, 18(1):21–34.
- Malis, E. (2002). Survey of vision-based robot control. *ENSIETA European Naval Ship Design Short Course, Brest, France*.
- Malis, E. (2004). Improving vision-based control using efficient second-order minimization techniques. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, pages 507–515.
- Martinet, P. and Gallice, J. (1999). Position based visual servoing using a non-linear approach. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, volume 1, pages 531–536.
- Merry, R., van de Molengraft, R., and Steinbuch, M. (2005). The influence of disturbances in iterative learning control. In *IEEE Proc. Conf. on Contr. Appl.*, pages 974–979.
- Merry, R., van de Molengraft, R., and Steinbuch, M. (2008). Iterative learning control with wavelet filtering. *International Journal of Robust and Nonlinear Control*, 18(10):1052–1071.
- Van der Meulen, S., Tousain, R. L., and Bosgra, O. H. (2008). Fixed structure feedforward controller design exploiting iterative trials: Applied to a wafer stage and a desktop printer. *ASME Journal of Dynamic Systems, Measurements, & Control*, 130(5). 051006.
- Mezouar, Y. and Chaumette, F. (2000). Path planning in image space for robust visual servoing. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, pages 2759–2764.
- Mezouar, Y. and Chaumette, F. (2002). Path planning for robust image-based control. *IEEE Trans. Robot. Autom.*, 18(4):534–549.
- Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117.
- Moore, K. L. (1993). *Iterative learning control for deterministic systems*. Springer-Verlag, New York.

- Moore, K. L. (2008). Iterative learning control: An expository overview. *Appl. & Comp. Contr., Sign. Proc., & Circ.*, 1(1):151–214.
- Moore, K. L. and Chen, Y. Q. (2002). On monotonic convergence of high order iterative learning update laws. In *15th IFAC Congr. Invited Session High-order Iterative Learning Control*, pages 1–6.
- Moore, K. L., Chen, Y. Q., and Ahn, H. S. (2006). Iterative learning control: a tutorial and big picture view. In *IEEE Proc. Conf. on Dec. & Contr.*, pages 2352–2357.
- Nakabo, Y., Ishikawa, M., Toyoda, H., and Mizuno, S. (2000). 1 ms column parallel vision system and its application of high speed target tracking. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, volume 1, pages 650–655.
- Namiki, A., Imai, Y., Ishikawa, M., and Kaneko, M. (2004). Development of a high-speed multifingered hand system and its application to catching. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, volume 3, pages 2666–2671.
- Namiki, A. and Ishikawa, M. (2003a). Robotic catching using a direct mapping from visual information to motor command. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, volume 2, pages 2400–2405.
- Namiki, A. and Ishikawa, M. (2003b). Vision-based online trajectory generation and its application to catching. *Control Problems in Robotics*, 4:249–264.
- Nian, C. Y. and Tarng, Y. S. (2005). An auto-alignment vision system with three-axis motion control mechanism. *Int. J. Adv. Manuf. Technol.*, 26(9–10):1121–1131.
- Norrlöf, M. (2000). *Iterative Learning Control: Analysis, Design, and Experiments*. PhD thesis, Linköpings University.
- Ogawa, N., Oku, H., Hashimoto, K., and Ishikawa, M. (2005a). Microrobotic visual control of motile cells using high-speed tracking system. *IEEE Trans. Robot.*, 21(4):704–712.
- Ogawa, N., Oku, H., Hashimoto, K., and Ishikawa, M. (2005b). Single-cell level continuous observation of microorganism galvanotaxis using high-speed vision. In *IEEE Proc. Int. Symp. on Biomed. Imag.*, pages 1331–1334.
- Opto Engineering (2010). MC Series: Zero Distortion Macro Lenses. <http://www.opto-engineering.com/mc-zero-distortion.html>.
- Papanikolopoulos, N., Khosla, P., and Kanade, T. (1993). Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. Robot. Autom.*, 9(1):14–35.

- Prosilica (2009). Ultra-compact gige vision cameras - gc640 fast cmos vga camera - 200 fps.
- Purwin, O. and D'Andrea, R. (2006). Trajectory generation and control for four wheeled omnidirectional vehicles. *Robotics and Autonomous Systems*, 54(1):13–22.
- Reichardt, E. (2010). High accuracy positioning using direct visual servoing. Technical Report CST 2010.142, University of Technology Eindhoven.
- RoboCup (2010). The official RoboCup site. <http://www.robocup.org>.
- Rotariu, I., Ellenbroek, R., van Baars, G., and Steinbuch, M. (2003). Iterative learning control for variable setpoints, applied to a motion system. In *IEEE Proc. Eur. Contr. Conf.*
- Rotariu, I., Steinbuch, M., and Ellenbroek, R. (2008). Adaptive iterative learning control for high precision motion systems. *IEEE Trans. Control Syst. Technol.*, 16(5):1075–1082.
- Sakou, H., Miyatake, T., S., K., and Ejiri, M. (1989). A position recognition algorithm for semiconductor alignment based on structural pattern matching. *IEEE Trans. Acoust., Speech, Signal Process.*, 37(12):2148–2157.
- Sanderson, A. and Weiss, L. (1980). Image-based visual servo control using relational graph error signals. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, pages 1074–1077.
- Schramm, F., Micaelli, A., and Morel, G. (2005). Calibration free path planning for visual servoing yielding straight line behaviour both in image and work space. In *IEEE Proc. Int. Conf. on Intell. Robot. & Syst.*, pages 2216–2221.
- Schramm, F. and Morel, G. (2006). Ensuring visibility in calibration-free path planning for image-based visual servoing. *IEEE Trans. Robot.*, 22(4):848–854.
- Sequeira Goncalves, P. J. (2001). Kinematic and dynamic 2d visual servoing. In *IEEE Proc. Eur. Contr. Conf.*
- Sequeira Goncalves, P. J. and Caldas Pinto, J. R. (2003). Dynamic visual servoing of robotic manipulators. In *IEEE Proc. Emerging Techn. and Fact. Autom.*, volume 2, pages 560–565.
- Sherback, M., Purwin, O., and D'Andrea, R. (2006). Real-time motion planning and control in the 2005 cornell robocup system. *Robot Motion and Control*, 335(1):245–263.

- Shimizu, K. and Hirai, S. (2006). Cmos+fpga vision system for visual feedback of mechanical systems. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, pages 2060–2065.
- Shirai, Y. and Inoue, H. (1973). Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5:99–108.
- Sonka, M., Hlavac, V., and Boyle, R. (1999). *Image Processing, Analysis and Machine Vision*. PWS Publishing.
- Stegger, C., Ulrich, M., and Wiedermann, C. (2008). *Machine Vision*. Wiley-VCH.
- Steinbuch, M. and van de Molengraft, M. J. G. (2000). Iterative learning control of industrial motion systems. In *1st IFAC Conference on Mechatronic Systems, Darmstadt, Germany*, pages 967–972.
- Sturm, J., Wu, C. C., Marcy, D., and Hebner, T. H. (2000). Fabrication of organic semiconductor devices using ink jet printing. United States Patent 6,087,196.
- Thuilot, B., Martinet, P., Cordesses, L., and Gallice, J. (2002). Position based visual servoing: Keeping the object in the field of view. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, volume 2, pages 1624–1629.
- Tomizuka, M. (1987). Zero phase error tracking algorithm for digital control. *ASME Journal of Dynamic Systems, Measurements, & Control*, 109(1):65–68.
- Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. Robot. Autom.*, 3(5):966–979.
- Tsourveloudis, N. C., Valavanis, K. P., and Hebert, T. (2002). Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic. *IEEE Trans. Robot. Autom.*, 17(4):490–497.
- Verstegen, P. P. H., Van Gastel, J. M. M., and Spronck, J. W. (2006). Design of a novel single camera vision system for both component and board alignment in pick and place machines. In *IFAC Symp. Mech. Syst.*
- Vincze, M. (2000). Dynamics and system performance of visual servoing. In *IEEE Proc. Int. Conf. on Robot. & Autom.*, pages 644–649.
- W3schools.com (2011). Browser display statistics. http://www.w3schools.com/browsers/browsers_display.asp.
- Weiss, L., Sanderson, A., and Neuman, C. (1987). Dynamic sensor-based control of robots with visual feedback. *IEEE Trans. Robot. Autom.*, 3(5):404–417.

- Wilson, W. J., Williams Hulls, C. C., and Bell, G. S. (1996). Relative end-effector control using cartesian position based visual servoing. *IEEE Trans. Robot. Autom.*, 12(5):684–696.
- Wittenberg, G. (1993). A robot to milk cows. *Industrial Robot: An International Journal*, 20(5):22–25.
- Xu, J. X. (1997). Direct learning of control efforts for trajectories with different magnitude scales. *Automatica*, 33(12):2191–2195.
- Xu, J. X., Xu, J., and Viswanathan, B. (2002). Recursive direct learning of control efforts for trajectories with different magnitude scales. *Asian J. of Contr.*, 4(1):49–59.
- You, B. J., Oh, Y. S., and Bien, Z. (1990). A vision system for an automatic assembly machine of electronic components. *IEEE Trans. Ind. Electron.*, 37(5):349–357.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334.
- Zheng, C., Su, Y., and Müller, P. C. (2009). Simple online smooth trajectory generations for industrial systems. *Mechatronics*, 19(4):571–576.

Appendix A

Stability proof

In this section we prove that if the inequalities (2.38) and (2.39) are satisfied, the arbitrary switching closed-loop system is stable.

Using the Schur complement, (2.38) can be written as

$$\begin{pmatrix} E & A_{cl,\min}^T E \\ EA_{cl,\min} & E \end{pmatrix} \succ 0. \quad (\text{A.1})$$

Since $\alpha_1 > 0$ we are allowed to write

$$\begin{pmatrix} \alpha_1 E & \alpha_1 A_{cl,\min}^T E \\ \alpha_1 EA_{cl,\min} & \alpha_1 E \end{pmatrix} \succ 0. \quad (\text{A.2})$$

The same transformation can be applied to (2.39)

$$\begin{pmatrix} \alpha_2 E & \alpha_2 A_{cl,\max}^T E \\ \alpha_2 EA_{cl,\max} & \alpha_2 E \end{pmatrix} \succ 0. \quad (\text{A.3})$$

Summing up the two inequalities above results in

$$\begin{pmatrix} E & (\alpha_1 A_{cl,\min}^T + \alpha_2 A_{cl,\max}^T) E \\ E(\alpha_1 A_{cl,\min} + \alpha_2 A_{cl,\max}) & E \end{pmatrix} \succ 0, \quad (\text{A.4})$$

which is the same as (2.37) using the Schur complement.

Appendix B

Bilinear interpolation equivalence

Let A and B be defined as in (3.8) and (3.9). Then the following holds

$$\frac{x_i^p - x_t^p}{x_b^p - x_t^p} = \frac{y_i^p - y_t^p}{y_b^p - y_t^p} = A, \quad (\text{B.1})$$

$$\frac{x_i^p - x_l^p}{x_r^p - x_l^p} = \frac{y_i^p - y_l^p}{y_r^p - y_l^p} = B. \quad (\text{B.2})$$

From (3.8) and (3.9) $x_l^p, y_l^p, x_r^p, y_r^p, x_t^p, y_t^p, x_b^p$ and y_b^p can be written as

$$x_l^p = x_{tl}^p + A(x_{bl}^p - x_{tl}^p), \quad (\text{B.3})$$

$$y_l^p = y_{tl}^p + A(y_{bl}^p - y_{tl}^p), \quad (\text{B.4})$$

$$x_r^p = x_{tr}^p + A(x_{br}^p - x_{tr}^p), \quad (\text{B.5})$$

$$y_r^p = y_{tr}^p + A(y_{br}^p - y_{tr}^p), \quad (\text{B.6})$$

$$x_t^p = x_{tl}^p + B(x_{tr}^p - x_{tl}^p), \quad (\text{B.7})$$

$$y_t^p = y_{tl}^p + B(y_{tr}^p - y_{tl}^p), \quad (\text{B.8})$$

$$x_b^p = x_{bl}^p + B(x_{br}^p - x_{bl}^p), \quad (\text{B.9})$$

$$y_b^p = y_{bl}^p + B(y_{br}^p - y_{bl}^p). \quad (\text{B.10})$$

The horizontal line connecting \underline{p}_l^p with \underline{p}_r^p can be written as

$$y = a_h x + b_h, \quad (\text{B.11})$$

with coefficients a_h and b_h

$$a_h = \frac{y_l^p - y_r^p}{x_l^p - x_r^p}, \quad (\text{B.12})$$

$$b_h = \frac{y_r^p x_l^p - y_l^p x_r^p}{x_l^p - x_r^p}. \quad (\text{B.13})$$

Similarly, the vertical line can be written as

$$y = a_v x + b_v, \quad (\text{B.14})$$

with

$$a_v = \frac{y_t^p - y_b^p}{x_t^p - x_b^p}, \quad (\text{B.15})$$

$$b_v = \frac{y_b^p x_t^p - y_t^p x_b^p}{x_t^p - x_b^p}. \quad (\text{B.16})$$

The intersection of the horizontal line and the vertical line is given by \underline{p}_i^p . Therefore, the coordinates of the point of interest can be obtained by equating (B.11) and (B.14) followed by solving for x . Substituting the solution of x in (B.11) gives the corresponding y solution. They are given by

$$x_i^p = AB(x_{tl}^p + x_{br}^p - x_{tr}^p - x_{bl}^p) + A(x_{bl}^p - x_{tl}^p) + B(x_{tr}^p - x_{tl}^p) + x_{tl}^p. \quad (\text{B.17})$$

$$y_i^p = AB(y_{tl}^p + y_{br}^p - y_{tr}^p - y_{bl}^p) + A(y_{bl}^p - y_{tl}^p) + B(y_{tr}^p - y_{tl}^p) + y_{tl}^p, \quad (\text{B.18})$$

These solutions together with (B.3) through (B.10) can be substituted into (B.1) and (B.2) which lead to equality.

Appendix C

Second order interpolation

This section explains how the function $h(B)$ is designed. Note that $g(A)$ is designed in the similar way. We choose the function $h(B)$ as a piecewise quadratic function

$$h(B) : \begin{cases} h_1(B) = a_{12}B^2 + a_{11}B + a_{10} & \text{if } B < 0.5, \\ h_2(B) = a_{22}B^2 + a_{21}B + a_{20} & \text{if } B \geq 0.5. \end{cases} \quad (\text{C.1})$$

The constraints as mentioned in Section 3.3.4 are written as

$$h_1(0) = 0, \quad h_2(1) = 1, \quad \frac{dh_1(0)}{dB} = \frac{P}{\bar{P}}, \quad \frac{dh_2(1)}{dB} = \frac{P}{\bar{P}}. \quad (\text{C.2})$$

Two additional constraints are added to have the function $h(B)$ continuous across $B = 0.5$ in both position and velocity

$$h_1(0.5B) = h_2(0.5B), \quad \frac{dh_1(0.5B)}{dB} = \frac{dh_2(0.5B)}{dB}. \quad (\text{C.3})$$

These constraints can be written as

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 & 1 & 0 \\ 0.25 & 0.5 & 1 & -0.25 & -0.5 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 \end{pmatrix} \begin{pmatrix} a_{12} \\ a_{11} \\ a_{10} \\ a_{22} \\ a_{21} \\ a_{20} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{P}{\bar{P}} \\ 1 \\ \frac{P}{\bar{P}} \\ 0 \\ 0 \end{pmatrix}, \quad (\text{C.4})$$

such that the parameters a_{12} , a_{11} , a_{10} , a_{22} , a_{21} and a_{20} can be found through

$$\begin{pmatrix} a_{12} \\ a_{11} \\ a_{10} \\ a_{22} \\ a_{21} \\ a_{20} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 & 1 & 0 \\ 0.25 & 0.5 & 1 & -0.25 & -0.5 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \frac{P}{P} \\ 1 \\ \frac{P}{P} \\ 0 \\ 0 \end{pmatrix}. \quad (\text{C.5})$$

The final formula for $h(B)$ is then given by

$$h(B) = \begin{cases} \left(2 - \frac{2P}{P}\right) B^2 + \frac{P}{P} B & \text{if } B < 0.5, \\ \left(-2 + \frac{2P}{P}\right) B^2 + \left(4 - \frac{3P}{P}\right) B - 1 + \frac{P}{P} & \text{if } B \geq 0.5. \end{cases} \quad (\text{C.6})$$

Appendix D

Minimum and maximum Jacobian values

In this section we show how to calculate the minimum and maximum values of the Jacobian given in (3.18). The feature-based position $\underline{p}_i^f = (x_i^f \ y_i^f)^T$ is given by

$$x_i^f = x_{ti}^f + g(A(x_i^p, y_i^p)), \quad (\text{D.1})$$

$$y_i^f = y_{ti}^f + h(B(x_i^p, y_i^p)). \quad (\text{D.2})$$

Therefore, by using the product rule we can rewrite the Jacobian of (3.18) as

$$J = \begin{pmatrix} \frac{dx_i^f}{dg} \frac{dg}{dA} \frac{dA}{dx_i^p} & \frac{dx_i^f}{dg} \frac{dg}{dA} \frac{dA}{dy_i^p} \\ \frac{dy_i^f}{dh} \frac{dh}{dB} \frac{dB}{dx_i^p} & \frac{dy_i^f}{dh} \frac{dh}{dB} \frac{dB}{dy_i^p} \end{pmatrix} \quad (\text{D.3})$$

$$= \begin{pmatrix} \frac{dg}{dA} & 0 \\ 0 & \frac{dh}{dB} \end{pmatrix} \begin{pmatrix} \frac{dA}{dx_i^p} & \frac{dA}{dy_i^p} \\ \frac{dB}{dx_i^p} & \frac{dB}{dy_i^p} \end{pmatrix}. \quad (\text{D.4})$$

For each part of (D.4) we will determine the minimum and maximum values as a function of α , starting with the first part. The functions $g(A)$ and $h(B)$ are given by

$$g(A) = \begin{cases} \left(2 - \frac{2P}{P}\right) A^2 + \frac{P}{P} A & \text{if } A < 0.5, \\ \left(-2 + \frac{2P}{P}\right) A^2 + \left(4 - \frac{3P}{P}\right) A - 1 + \frac{P}{P} & \text{if } A \geq 0.5. \end{cases} \quad (\text{D.5})$$

$$h(B) = \begin{cases} \left(2 - \frac{2P}{\bar{P}}\right) B^2 + \frac{P}{\bar{P}} B & \text{if } B < 0.5, \\ \left(-2 + \frac{2P}{\bar{P}}\right) B^2 + \left(4 - \frac{3P}{\bar{P}}\right) B - 1 + \frac{P}{\bar{P}} & \text{if } B \geq 0.5. \end{cases} \quad (\text{D.6})$$

The derivatives with respect to A and B can be written as

$$\frac{dg}{dA} = \begin{cases} \left(4 - \frac{4P}{\bar{P}}\right) A + \frac{P}{\bar{P}} & \text{if } A < 0.5, \\ \left(-4 + \frac{4P}{\bar{P}}\right) A + \left(4 - \frac{3P}{\bar{P}}\right) & \text{if } A \geq 0.5. \end{cases} \quad (\text{D.7})$$

$$\frac{dh}{dB} = \begin{cases} \left(4 - \frac{4P}{\bar{P}}\right) B + \frac{P}{\bar{P}} & \text{if } B < 0.5, \\ \left(-4 + \frac{4P}{\bar{P}}\right) B + \left(4 - \frac{3P}{\bar{P}}\right) & \text{if } B \geq 0.5. \end{cases} \quad (\text{D.8})$$

Dependent on the momentary pitch P , the values of $\frac{dg}{dA}$ and $\frac{dh}{dB}$ vary. If ΔP is $\alpha\bar{P}$, then the largest momentary pitch P is given by $\sqrt{(\bar{P} + 2\alpha\bar{P})^2 + (2\alpha\bar{P})^2}$. Similarly, if ΔP is $\alpha\bar{P}$, then the smallest momentary pitch P is given by $\bar{P} - 2\alpha\bar{P}$. Now for different values of α we search for the maximum and minimum values of $\frac{dg}{dA}$ and $\frac{dh}{dB}$ on the interval $0 \leq A \leq 1$ and $0 \leq B \leq 1$. Therefore, we can find the minimum and maximum values of $\frac{dg}{dA}$ and $\frac{dh}{dB}$, which are graphically depicted in Fig. D.1. The function corresponding to the maximum values of both $\frac{dg}{dA}$ and $\frac{dh}{dB}$ is given by $(2\sqrt{10} - 4)\alpha + 1$, whereas the function corresponding to the minimum values of both $\frac{dg}{dA}$ and $\frac{dh}{dB}$ is given by $(4 - 2\sqrt{10})\alpha + 1$.

The minimum and maximum values of the second part of (D.4) are determined as follows. As explained in Section 3.3.3, the values of A and B can be determined analytically using (3.8), (3.9) and (3.10). Note that the values of A and B are dependent on 1) the coordinates of the four enclosing features and 2) the coordinates of the point of interest. For the second part of the Jacobian in (D.4), we have to derive A and B with respect to the coordinates of the point of interest. These derivatives are dependent on the positions of the four enclosing features, which as a result of the pitch imperfections can lie with a predefined range subject to (3.1). For different positions of the four enclosing features, different derivatives will appear. We have investigated all possible combinations of configuration of the four enclosing features. For each value of α and for each of those combinations the minimum and maximum values were determined. These are graphically depicted in Fig. D.2. The multiplication of the Fig. D.1 and Fig. D.2 leads to the result given in Fig. 3.16.

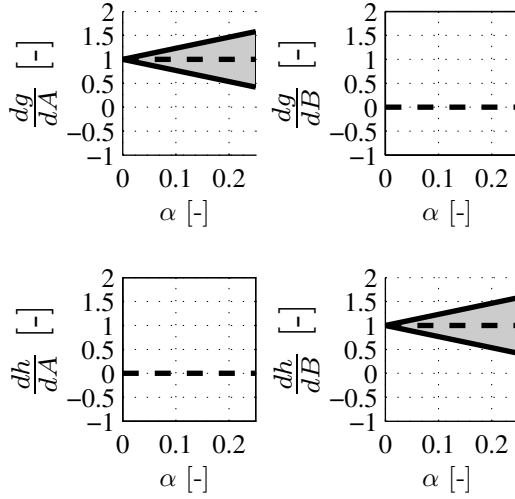


Figure D.1: First part of the Jacobian given in (D.4).

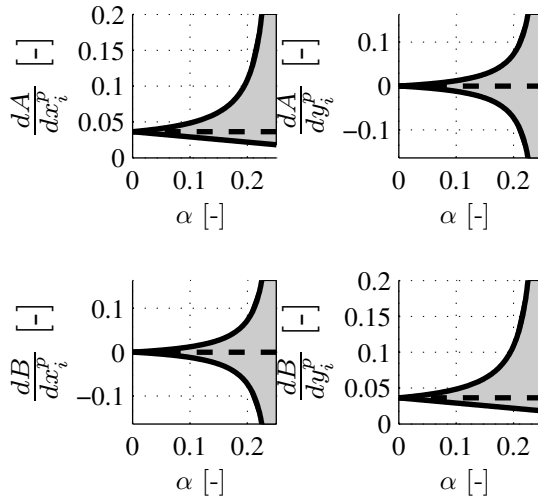


Figure D.2: Second part of the Jacobian given in (D.4).

Samenvatting

In veel fabricageprocessen worden productiestappen uitgevoerd op repeterende structuren. Deze structuren bestaan uit identieke producten die geplaatst zijn in een repeterend patroon. Bij ieder van deze producten zijn één of meerdere productiestappen nodig om het uiteindelijke product te verkrijgen. Voor een hoge productkwaliteit is het noodzakelijk om het bewerkingsgereedschap met een hoge nauwkeurigheid ten opzichte van de individuele producten te positioneren. In de huidige industrie worden vaak lokale positie-sensoren zoals encoders gebruikt om afzonderlijk de positie van het gereedschap en de tafel waarop de repeterende structuur zich bevindt te meten. De haalbare nauwkeurigheid is in dat geval direct afhankelijk van de thermische stabiliteit van de machine, de stijfheid van het machineframe en de variatie in de onderlinge afstand tussen de producten van de repeterende structuur. De ontwerpeisen van deze eigenschappen dreigen onhaalbaar streng te worden door de tendens naar grotere repeterende structuren.

Het doel van dit proefschrift is het ontwerpen van regeltechnische methoden om het gereedschap nauwkeurig te kunnen positioneren ten opzichte van de producten van de repeterende structuur, zonder dat overmatig strenge eisen nodig zijn ten aanzien van thermische stabiliteit, stijfheid van het machineframe en de gelijkmatigheid van de repeterende structuren. In dit proefschrift wordt een camera gebruikt in de regellus om de relatieve positie tussen het gereedschap en de producten te meten en te regelen. Dit wordt ook wel *visual servoing* genoemd. Door een camera te gebruiken als meetinstrument kunnen de relevante dynamica en bijbehorende verstoringen gemeten en gecompenseerd worden in een gesloten-lus regelkring op de plaats waar de positioneer-nauwkeurigheid gewenst is.

Vaak is de onderlinge afstand tussen twee opeenvolgende producten van de repeterende structuur niet perfect constant, bijvoorbeeld door thermische uitzetting van de repeterende structuur of door de eindige nauwkeurigheid van voorgaande productiestappen. Als gevolg hiervan varieert de afstand tussen twee producten,

zodat vooraf geen referentietrajecten ontworpen kunnen worden. In dit proefschrift wordt een nieuwe product-gebaseerde positiemeting gebruikt die de positie uitgeeft in producteenheden. Aangezien de product-gebaseerde posities van iedere product vooraf bekend zijn, kunnen referentietrajecten daarom vooraf ontworpen worden zonder kennis van de exacte metrische positie. Naast bewegingen van product naar product zijn ook kleine metrische bewegingen van het gereedschap ten opzicht van het product mogelijk, zoals bijvoorbeeld bij graveren. De regeltechnische methoden zijn in de praktijk gevalideerd op een industriële machine. Een visueel servoing opstelling is gerealiseerd met een beeldverwerkingssnelheid van 1 kHz. Deze opstelling bestaat uit een xy-tafel waarop een wafer is geklemd die de discrete semiconductors bevat die samen een repeterende structuur vormen. De taak is om het gereedschap relatief te positioneren ten opzichte van de discrete semiconductors van de wafer. Een camera die is gericht op de repeterende structuur van de wafer wordt gebruikt voor de positie-terugkoppeling. De tijdsvertraging van het systeem is 2.5 ms en de variatie van de positiemeting is $0.3 \mu\text{m}$, (3σ). De uiteindelijke behaalde nauwkeurigheden voor de industriële machine zijn $\pm 10 \mu\text{m}$, hetgeen de prestaties van de traditioneel geregelde machine aanzienlijk verbetert.

Voor bewegingen van product naar product met variërende onderlinge afstand tussen de producten is een nieuwe voorwaartssturing ontworpen gebaseerd op iteratief lerende regeltechnieken (ILC). Om met de variërende afstand tussen de producten om te kunnen gaan, worden de referentietrajecten van product naar product ontworpen door een nominaal metrisch referentietraject te schalen. Tijdens het leerproces worden deze geschaalde trajecten toegepast en wordt er tweede-orde ILC gebruikt om de variërende trajecten met een hoge nauwkeurigheid te volgen. De uiteindelijke behaalde nauwkeurigheden op de industriële machine zijn $\pm 5 \mu\text{m}$ voor verschillende geschaalde referentietrajecten, hetgeen een significante verbetering van de prestatie betekent.

Dankwoord

Een proefschrift komt niet alleen tot stand door degene wiens naam je op de kaft vindt. Veel mensen hebben mij op wat voor manier dan ook geholpen om mijn proefschrift tot dit resultaat te brengen. Daarom wil ik graag, onder het genot van een colaatje, van de gelegenheid gebruik maken om tegen iedereen te zeggen: bedankt! In het bijzonder wil ik een aantal mensen noemen.

Maarten, tijdens mijn afstuderen kwam je naar me toe en vroeg me om nog vier jaar te blijven. Je enthousiastmerende karakter en enorme drive hadden al een halve ja bewerkstelligd. Met het bijbehorende leuke onderwerp was daarom de keus al snel helder. Bedankt voor het vertrouwen en de kans die je me hebt gegeven, zodat ik uiteindelijk sta waar ik nu sta. Voor de dagelijkse begeleiding wil ik graag mijn co-promotor René van de Molengraft bedanken. Jouw kritische blik, 'helicopterview', maar ook versimpelingen hebben bijgedragen aan de kwaliteit van dit proefschrift. Naast het werken aan mijn onderzoek zorgden jouw diverse onderwijstaken waaraan ik mee mocht werken voor een zeer welkome afwisseling.

Alle mensen van DCT wil ik bedanken voor de geweldige sfeer waarvan ik heb mogen genieten tijdens de uitjes, koffiepauzes en conferenties. Graag wil ik speciaal mijn kamergenoten die ik heb 'versleten' bedanken: Cesar, Gert, Erik, Tim en in het bijzonder Roel en Rob! Verder wil ik ook mijn afstudeerders Rob, Erik en Lancheng bedanken voor hun bijdrage aan dit proefschrift. 's Ochtends koffie drinken met Harrie, Rob, Pieter, Sjef, Peter en Hennie zorgde elke dag weer voor een leuke start met een enorme diversiteit aan gesprekstof. Bedankt daarvoor!

Het Tech United RoboCup team mag ook zeker niet ontbreken in mijn lijstje. Ups en downs hebben we met elkaar gedeeld. Elk toernooi, elke dinsdagavond, elke keer hebben we veel plezier gehad en hebben we samen mooie prestaties behaald onder soms onmenselijke omstandigheden met bijbehorende emoties, maar met altijd één heel duidelijk doel voor ogen: wij worden de beste. Ik kijk terug op een

schitterende tijd en een geweldige ervaring met het mooiste team waarin ik ooit heb mogen deelnemen. Bedankt!

Pap en mam, ondanks het feit dat het af en toe toch maar moeilijk was om te begrijpen waar ik nou precies mee bezig was, was jullie steun en interesse er niet minder om. Bedankt daarvoor! En dan is er nog mijn grote bruur. Je zorgzaamheid en interesse hebben mij enorm geholpen, evenals de vele telefoontjes waarbij ik alle vragen op je af kon vuren. Bruur, bedankt wah!

Tenslotte wil ik mijn geweldige gezinnetje bedanken. Hein, zo klein als je nu bent, zo veel steun heb je pappa al gegeven. Bij een guitige glimp van jou wist ik meteen dat ik thuis was en kwam het werk later wel weer. Tanja, ik ben in de wolken van jou, maar gelukkig heb je me tegelijkertijd ook met beide benen aan de grond gehouden. Zonder jouw onvoorwaardelijke steun en liefde was dit proefschrift er nooit gekomen. Schatten, ik hou van jullie!

Jeroen de Best
Juli, 2011

Curriculum Vitae

Jeroen de Best was born on June 30, 1982 in Wamel, the Netherlands. After finishing his secondary education in 2000 at Pax Christi College in Druten, he studied Mechanical Engineering at the Eindhoven University of Technology. In 2006 he received the M.Sc. degree (Cum Laude) within the Control Systems Technology group on the topic 'Piecewise Linear Sheet Control in a Printer Paper Path'. Since September 2006, he is a Ph.D. student in the Control Systems Technology group at the department of Mechanical Engineering of Eindhoven University of Technology from which the results are presented in this dissertation. During his Ph.D. project, he completed the educational program of the Graduate School DISC and was a member of the Tech United RoboCup team Eindhoven.

