Gaussian Process Position-Dependent Feedforward

Citation for published version (APA):

Document license:
CC BY

DOI:
10.1109/AMC51637.2022.9729327

Document status and date:
Published: 11/03/2022

Document Version:
Accepted manuscript including changes made at the peer-review stage

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.

Download date: 14. Sep. 2023
Gaussian Process Position-Dependent Feedforward: With Application to a Wire Bonder

Max van Haren*, Maurice Poot†, Dragan Kostić‡, Robin van Es§, Jim Portegies† and Tom Oomen*§

*CASA, Dept. of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands, email: m.j.v.haren@tue.nl
†ASM Pacific Technology, Centre of Competency, Beuningen, The Netherlands
‡CASA, Dept. of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
§Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands

Abstract—Mechatronic systems have increasingly stringent performance requirements for motion control, leading to a situation where many factors, such as position-dependency, cannot be neglected in feedforward control. The aim of this paper is to compensate for position-dependent effects by modeling feedforward parameters as a function of position. A framework to model and identify feedforward parameters as a continuous function of position is developed by combining Gaussian processes and feedforward parameter learning techniques. The framework results in a fully data-driven approach, which can be readily implemented for industrial control applications. The framework is experimentally validated and shows a significant performance increase on a commercial wire bonder.

I. INTRODUCTION

Increasing performance requirements for motion control leads to situations where position-dependency of mechatronic systems cannot be neglected anymore. An example is the wafer stage, where a flexible mode of a wafer is observed differently for each position [1]. Furthermore, H-drive machines, such as large-format printing systems, suffer from position-dependent dynamics due to a changing configuration [2]. Traditionally, position-dependence is neglected and the feedforward controller is kept constant over the machine operating range, resulting in suboptimal performance. Recently, due to the developments in computational power, data-driven techniques, such as Gaussian Processes (GPs), are getting more relevant for motion control.

Traditional Linear Time-Invariant (LTI) feedforward design attempts to compensate for a known reference signal of a system. In this paper, the feedforward parameters of a system are not yet developed for position-dependent dynamics. Position-dependent feedforward design can compensate for position-dependent behavior of systems. For this purpose, feedforward parameters can be determined in a grid and estimated with parametric regression techniques such as linear interpolation. However, linear interpolation has approximation errors since the dependency between position and feedforward parameters is generally unknown.

High motion control performance for systems with position-dependent dynamics can furthermore be achieved through the use of Linear Parameter Varying (LPV) or non-linear inversion-based control of the system. First, LPV dynamics can be scheduled according to the current configuration of the LPV system, resulting in high control performance for e.g. wafer stages [5] or xy-positioning tables [6]. Second, data-driven learning techniques, such as Iterative Learning Control (ILC) or adaptive control, can be extended to LPV systems, which results in high performance through learning [7], [8]. Third, a non-linear inversion-based output tracking solution can be applied, resulting in asymptotically exact output tracking [9]. LPV model-free approaches are investigated in [10], directly learning LPV controllers from data, but are at present not competitive with model-based designs. The high performance achieved by using LPV or non-linear control typically requires accurate and extensive modeling, that is often very challenging and the high cost and complexity are usually not justified for industrial control applications.

Although feedforward design has improved significantly compared with traditional acceleration feedforward, a feedforward with systematic tuning for position-dependent effects, capable of estimation at any arbitrary position, is currently lacking. This paper models feedforward parameters as a continuous function of position through a GP [11], [12], which allows for the compensation of position-dependent effects without a full LPV or non-linear model. In addition, a GP is non-parametric and therefore does not require an assumption on the parametric form between the position and the feedforward parameters. Gaussian processes have been applied previously in system identification [13], but are not broadly applied to feedforward and are not yet developed for position-dependent systems. In this paper, the feedforward parameters of a system
are learned in a trial-to-trial fashion using ILC with Basis Functions (ILCBF) [14]. The contributions include:

C1 a generic framework to model feedforward parameters as a function of position using GPs, which can be readily implemented for industrial applications,

C2 ILCBF to automatically learn feedforward parameters for multiple fixed positions, suitable for industrial machines, that are directly used in the GPs,

C3 application and validation of the framework to a state-of-the-art industrial experimental setup, showing the capabilities of the framework.

The outline of this paper is as follows. In Section II, the problem that is considered in this paper is defined. In Section III, the method for modeling feedforward parameters as a GP and an approach to automatically learn feedforward parameters with ILCBF is described, leading to contributions C1 and C2. In Section IV, a case study on an experimental setup is performed, constituting contribution C3. Finally, in Section V, concluding remarks are given.

Notation: Systems can be single-input single-output or multiple-input multiple-output with $n_i$ inputs and $n_o$ outputs. All systems are discrete-time, unless stated otherwise, with discrete-time $k \in \{0, 1, \ldots, N - 1\}$. Continuous time systems are transformed in their discrete-time counterpart using finite difference approximation. The trial number is indicated with the index $j$. Signals are assumed to be of length $N$. The weighted 2-norm of a vector $x \in \mathbb{R}^N$ is denoted as $\|x\|_W := \sqrt{x^TWx}$, where $W \in \mathbb{R}^{N \times N}$ is a weighting matrix. Matrix $A \in \mathbb{R}^{N \times N}$ is positive (semi-)definite if and only if $x^TAx \geq 0$, $\forall x \neq 0 \in \mathbb{R}^N$ and is denoted as $A \succeq 0$.

II. Problem Definition

In this section, the problem for determining a position-dependent feedforward controller is formulated. First, a problem setup is given, including the systems considered and the parametrization of the feedforward signal. Second, the three categories of position-dependent effects in mechatronic systems considered are elaborated upon. Finally, a hypothesis of the largest contribution to the position-dependency is made and the problem addressed in this paper is defined.

A. Problem Setup

The considered class of position-dependent systems are spatially distributed LTI systems [15]

$$y(k) = G(\rho, q^{-1})u(k),$$

with output $y(k) \in \mathbb{R}^{n_o \times n_i}$, input $u(k) \in \mathbb{R}^{n_i \times n_i}$, system $G(\rho, q^{-1}) \in \mathbb{R}^{n_o \times n_i}$, the initial position $\rho \in \mathbb{R}^{n_o}$ and $q$ denotes the forward-shift operator, i.e. $q^{-1}a(k) = a(k-\tau)$. Spatially distributed LTI systems are generally applicable to systems with slowly varying position-dependency, i.e., the position-dependency is mostly exerted due to the initial position and not due to the reference signal. The wire bonder in Fig. 1 is a benchmark example, since references are relatively short compared with the machine operating range.

The applied control structure can be seen in Fig. 2. Typically, the design of feedforward controller $F$ tries to reduce the reference-induced error to zero, i.e. $e(k) = 0 = r(k) - G(\rho, q^{-1})u(k)$, resulting in the ideal feedforward controller $F = G^{-1}(\rho, q^{-1})$. Often, the feedforward design is based on the physical model of a system. Consider, e.g., a finite-difference approximation of a mass-damper system

$$G_{md}(q^{-1}) = \frac{1}{m(1-q^{-1})^2 + d \frac{1-q^{-1}}{T_s}},$$

with $T_s$ the sampling time of the system, $m$ the mass of the system and a damping coefficient $d$. Zero reference-induced error is achieved by the ideal feedforward controller $F(q^{-1}) = m(1-q^{-1})^2 + d \frac{1-q^{-1}}{T_s}$. However, due to model uncertainties, the exact system parameters are unknown and therefore the controller takes the form

$$F(q^{-1}) = \left(l \frac{1-q^{-1}}{T_s} \frac{1-q^{-1}}{T_s} \right) \left[\hat{m} \right] = \Psi(q^{-1})\theta,$$

with $\hat{m}$ an estimate or modeled value in the feedforward parameters $\theta \in \mathbb{R}^{n_o \times n_i}$ and the basis function matrix $\Psi \in \mathbb{R}^{n_o \times n_i}$. Due to the position-dependency of system $G(\rho, q^{-1})$, best performance is achieved by modeling the feedforward parameters $\theta$ as a function of position.

B. Considered Position-Dependent Effects

Position-dependency of $G(\rho, q^{-1})$ can be caused by several factors, here separated in actuation, mechanical or sensing. First, actuation can cause position-dependency due to for instance cogging [16] or varying magnetic flux density [17]. Second, mechanical position-dependency can directly affect the feedforward parameters that achieve optimal performance. Examples of such are raster scanning of atomic force microscopy [18] or position-dependent gantry systems, such as the large-format printer [2]. Lastly, since sensors are typically
Fig. 3. Illustration of the developed framework, where feedforward parameters are being modeled as a function of position using ILCBF and GPs.

The mean function \( m \) can be interpreted as the mean at any input point and the covariance function \( k \) as the similarity between values of \( f(\rho) \) on different inputs \( \rho \) and \( \rho' \).

B. Gaussian Processes for Position-Dependent Feedforward Parameters

In this section, feedforward parameters are estimated on test positions \( \rho \in P_* \), given feedforward parameters learned on training positions \( \rho \in P \) using GPs, leading to contribution C1. The test and training positions are defined as

\[
P_* = [\rho_1, \rho_2, \ldots, \rho_l]^T \in \mathbb{R}^{l \times n_o},
\]

\[
P = [\rho_1, \rho_2, \ldots, \rho_l]^T \in \mathbb{R}^{l \times n_o},
\]

with \( l, l \in \mathbb{N} \) the amount of test and training positions, respectively. Observations of the feedforward parameters, i.e. \( \theta(P) \), are contaminated with noise,

\[
\theta(P) = f(P) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbb{I}),
\]

with the noise variance \( \sigma^2 \). The continuous function of position \( \theta(P) \) and the observations \( \theta(P) \) are assumed to be random variables and have a joint multivariate Gaussian distribution

\[
\begin{bmatrix}
\theta(P) \\
\theta(P_*)
\end{bmatrix} \sim \mathcal{N} \left( 0, \begin{bmatrix}
K(P, P) + \sigma^2 \mathbb{I} & K(P, P_*) \\
K(P_*, P) & K(P_*, P_*)
\end{bmatrix} \right),
\]

with covariance matrices \( K_y := K(P, P) + \sigma^2 \mathbb{I} \in \mathbb{R}^{l \times l} \), \( K_\theta := K(P, P_*) = K(P_*, P)^\top \in \mathbb{R}^{l \times l} \), and \( K_{\theta\theta} := K(P_*, P_*) \in \mathbb{R}^{l \times l} \), which specify similarity between outputs on different positions. The parameter \( \sigma^2 \) is an approximation of the noise variance \( \sigma^2 \) on the observations in (5). The parameter \( \sigma^2 \), in addition to the hyperparameters of the covariance function \( K \), are approximated based on data by using marginal likelihood optimization [11, Chapter 5]. Note that the mean function is assumed to be zero, which is not strictly necessary, see e.g. [11, Section 2.7]. The joint distribution in (6) can be conditioned on function observations using Bayes’ rule, resulting in the posterior distribution

\[
\theta(P_*) \mid [P, P_*, \theta(P)] \sim \mathcal{N}(\tilde{\theta}(P_*), \text{cov}(\theta(P_*))),
\]

with posterior mean and posterior covariance

\[
\tilde{\theta}(P_*) := \mathbb{E}(\theta(P_*)) = K \tilde{T} K_y^{-1} \theta(P),
\]

\[
\text{cov}(\theta(P_*)) = K_{\theta\theta} - K \tilde{T} K_y^{-1} K_{\theta\theta}.
\]

In combination with a suitable covariance function, (8) estimates feedforward parameters, given parameter observations.

C. Learning Feedforward Parameters using Basis Functions for a Fixed Position

In this section, feedforward parameters are learned for multiple fixed positions to serve as training data for the GP, leading to contribution C2. Here, ILCBF is used to learn the parameters, but the framework can directly be extended to other feedforward parameter tuning approaches.

The optimization criterion in ILCBF is specified as [21]

\[
V(\theta_{j+1}) = ||e_{j+1}(k)||_W^2 + ||f_{j+1}(k)||_{W_\Delta}^2 + ||f_{j+1}(k) - f_j(k)||_{W_\Delta}^2.
\]

The optimization criterion in ILCBF is specified as [21]

\[
V(\theta_{j+1}) = ||e_{j+1}(k)||_W^2 + ||f_{j+1}(k)||_{W_\Delta}^2 + ||f_{j+1}(k) - f_j(k)||_{W_\Delta}^2,
\]

\[
\theta_{j+1} = \text{argmin}_\theta V(\theta_{j+1}).
\]
with weighting matrices $W_e > 0$ and $W_f$, $W_{\Delta f} \succeq 0$ and $\theta_j$ the feedforward parameters in trial $j$. The error $e_{j+1}$ is

$$e_{j+1}(k) = S(q^{-1})r(k) - S(q^{-1})G_0(q^{-1})f_{j+1}(k) = e_j(k) - S(q^{-1})G_0(q^{-1})(f_{j+1}(k) - f_j(k)), \quad (10)$$

where now, $G_0$ is for instance a nominal model of a position-dependent system. The feedforward signal is parameterized in terms of the feedforward parameters $\theta_j$, i.e., $f_j(k) = \Psi(q^{-1})r(k)\theta_j$. The feedforward parameters are updated as

$$\theta_{j+1}^* = \arg\min_{\theta_{j+1}} V(\theta_{j+1}). \quad (11)$$

Since the feedforward force is chosen linearly in the feedforward parameters, the optimization criterion in (9) becomes quadratic in $\theta_{j+1}$. Hence, an analytic solution to (12) is [22]

$$\theta_{j+1} = Le_j + Q\theta_j,$$

$$L = R^{-1}(\Psi^T G_0^T S^T W_e),$$

$$Q = R^{-1}\Psi^T (G_0^T S^T W_e G_0 S + W_{\Delta f}) \Psi,$$

$$R = (\Psi^T (G_0^T S^T W_e G_0 S + W_f + W_{\Delta f}) \Psi)^{+},$$

where $(q^{-1})$ and $(k)$ have been left out for brevity. The parameter update in (12) leads to monotonic convergence of $\|f_j(k)\|$, provided matrices $W_e$, $W_f$ and $W_{\Delta f}$ are selected properly [21]. Robustness, with respect to model mismatch due to the position-dependent dynamics, can be enforced by increasing $W_f$. Now, (12) can be used in combination with the error in trial $j$ to compute a new set of feedforward parameters $\theta_{j+1}$ for a fixed training position.

### D. Developed Framework and Example

The combination of GPs and ILCBF is used to model feedforward parameters as a function of position. The framework is schematically shown in Fig. 3. First, ILCBF is performed $l$ times on the positions $\rho \in P$, where $P$ is chosen by the user. Second, the feedforward parameters learned in ILCBF, i.e. the training data $\theta(P)$, can be used in a GP regression. Finally, the GP results in the feedforward parameters modeled as a function of the test inputs $P_*$, i.e. $\theta(P_*)$.

**Example 1.** Consider $G_m$ with spatially-distributed mass,

$$G_m(\rho, q^{-1}) = \frac{1}{\bar{m}} \left(1 - 2 \left(\frac{1}{2} - \rho\right)^2 \left(1 - q^{-1}\right)^2\right) \forall \rho \in [0, 1],$$

with $\bar{m} = 1$ kg a nominal mass. The feedforward controller is designed using acceleration feedforward, i.e.,

$$f_j(k) = \Psi(q^{-1})r(k)\theta_j = \ddot{r}(k)\theta_j.$$

The feedforward parameters are learned on 4 positions $\rho \in P$, with $P = [0.05 \quad 0.35 \quad 0.65 \quad 0.95]$, using ILCBF. The feedforward parameters have bias with respect to the true system values due to measurement noise, which is typically observed in ILCBF [23]. The framework is applied using a squared exponential covariance function, see e.g. [11, Section 4.2]. Fig. 4 shows how the framework models a position-dependent feedforward parameter, resulting in an accurate model.

### IV. Experimental Case Study

In this section, position-dependent feedforward with GPs is applied to a benchmark experimental setup, hence constituting contribution C3. First, the example setup is presented, followed by the application and results of the framework. Finally, a discussion is presented regarding the outcome of the experiments.

#### A. Experimental Setup

Position-dependent effects hamper the performance of semiconductor back end equipment, such as the commercial wire bonder seen in Fig. 1. Position-dependent dynamics can be caused due to several reasons, e.g. position-dependent actuators, flexible dynamics and changing configuration. The wire bonder consists of two inputs $u_1$ and $u_2$ and two outputs $y_1$ and $y_2$. The feedforward parameters are modeled as a function of the initial position of the machine for $y_1$ and $y_2$. The feedforward force $f$ is parameterized using

$$\Psi(q^{-1})r(k) = \begin{bmatrix} \bar{r}_1(k) & \dddot{r}_1(k) & 0 & 0 & 0 \\ 0 & 0 & \bar{r}_2(k) & \dddot{r}_2(k) & \psi_2(k) \end{bmatrix}, \quad (13)$$

with $\psi_2(k)$ a non-linear basis function. The references $r_1$ and $r_2$ are polynomial trajectories, see [24]. The feedforward parameters are defined as

$$\theta = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5]^T, \quad (14)$$

which are used, in combination with the basis function matrix in (13), to calculate the feedforward force $f$.

#### B. Experimental Results

The framework is applied to the wire bonder seen in Fig. 1. First, feedforward parameters are learned on several positions with ILCBF using 20 trials, as described in Section III-C. The error 2-norm and parameters of the first 8 trials in the center position can be seen in Fig. 5, which shows that ILCBF has converged. Second, each feedforward parameter in (14) is modeled as a separate GP, as seen in Section III-B, by using the feedforward parameters learned in trials 13 up to 20 on the training positions in Fig. 6. As many converged feedforward parameters should be used in the algorithm, such that the result is unbiased and the variance of the GP is as low as possible. A squared exponential covariance function is used. For visualization purposes, the positions $P_*$ are chosen as a
fine grid covering the operating range of the machine. In Fig. 7 and Fig. 8 the GP regressions of the acceleration feedforward parameters can be seen and show a significant difference in the parameters for the operating range of the machine.

To evaluate the performance of the framework, several test positions other than the training positions have been chosen, seen in Fig. 6. The GP regressions made with the feedforward parameters learned on the training positions are used to estimate feedforward parameters on the test positions using (8). On the test positions, three methods of choosing feedforward parameters have been performed:

M1 center, where the feedforward parameters determined in the center position of the machine are used,

M2 GP, where the feedforward parameters are estimated using the GP at the given test position,

M3 local ILCBF, where the feedforward parameters are determined at the test position using ILCBF.

Here, local ILCBF is performed to serve as a reference frame to compare the first two methods with. The error 2-norm for the test positions for all methods can be seen in Fig. 9. The error 2-norm of \( y_1 \) in Fig. 9 shows that a significant performance increase can be achieved when using feedforward parameters modeled as a function of position by a GP. This is additionally supported by looking at the time domain error of \( y_1 \) for test position 2 in Fig. 10. Furthermore, the GP feedforward achieves an error 2-norm similar to that of ILCBF, indicating optimal performance in terms of (9) with the specified basis function and requiring no learning at the test position, in contrast to local ILCBF. The error 2-norm of \( y_2 \) shows the performance difference for the two methods and ILCBF is only marginal, showing that the \( y_2 \) direction does not have much position-dependency.

C. Discussion of Experimental Case Study

The experimental setup has position-dependent effects, which can be concluded by looking at the feedforward pa-
parameters in Fig. 7 and Fig. 8. For the acceleration feedforward parameter $\theta_2$ in Fig. 7, significant position-dependent effect is seen. The unusual behavior of the parameter might be caused due to the periodic magnetic flux density in linear actuators [17]. Indeed, when measuring the distance between the peaks in Fig. 7, a distance roughly equal to the magnet pitch of the linear actuator is observed. The position-dependent effect is directly seen in the performance difference for $y_1$ in Fig. 9.

For test positions 1 and 3, the performance for center and GP feedforward are equal for the $y_1$ direction. This can directly be explained by looking at Fig. 6 and Fig. 7, that shows test positions 1 and 3 are located roughly one magnet pitch away from the center position, resulting in optimal feedforward parameters equal to the center position.

Compared with the acceleration feedforward parameter $\theta_2$, the acceleration feedforward parameter $\theta_1$ in Fig. 8 has considerably less position-dependency. The unusual behavior is also not observed for $\theta_1$, indicating that the magnetic flux density is not affecting the feedforward parameter as much. When taking a look at the performance for $y_2$, the difference is only marginal between the feedforward methods. Therefore, it can be concluded that for the considered experimental setup, position-dependent effects in the actuator heavily outweigh position-dependent effects in the mechanics or sensing.

V. CONCLUSION

The developed framework models feedforward parameters as a function of position with GPs, resulting in constant performance for any arbitrary position. The training data for the GPs are determined using ILCBF, but can be extended to any feedforward parameter tuning approach. Since the position-dependent effects of motion systems are typically unknown or hard to model, non-parametric regression techniques, such as GPs, work especially well. GPs, in combination with a data-driven feedforward parameter tuning approach, results in a framework where data outweighs assumptions on the position-dependency of a motion system. Lastly, experiments on a complex motion system show that the framework can improve control performance significantly by compensating for, e.g., an unknown position-dependent magnetic flux density. Ongoing work on this topic is directed at optimal and automatic computation of the training positions using sensor placement techniques such as mutual information optimization.

ACKNOWLEDGMENT

The authors wish to thank Kelvin Kai Wa Yan for his support during the experimental study.

REFERENCES


Sample Index [-]

[10x49]1

[13x73][9x46]e

[9x53](

[9x59])

[10x45]1

[16x38]∥

[16x49]t

[16x52]∥

[16x58]1

[18x45]0

[28x25]−

[34x25]0

[34x67]0.

[41x25]5

[41x46]0

[41x67]5

[41x88]1

[49x76]techniques such as mutual information optimization.

computation of the training positions using sensor placement work on this topic is directed at optimal and automatic dependency of a motion system. Lastly, experiments on a framework where data outweighs assumptions on the position-driven feedforward parameter tuning approach, results in a GPs, work especially well. GPs, in combination with a data-driven feedforward approach for LTI/LTV systems is through LQ tracking and stable inversion. Enabling large feedforward tasks on a position-dependent printer, Mechatronics, vol. 38, pp. 76–90, 2016.


Sample Index [-]

[10x49]1

[13x73][9x46]e

[9x53](

[9x59])

[10x45]1

[16x38]∥

[16x49]t

[16x52]∥

[16x58]1

[18x45]0

[28x25]−

[34x25]0

[34x67]0.

[41x25]5

[41x46]0

[41x67]5

[41x88]1

[49x76]techniques such as mutual information optimization.

computation of the training positions using sensor placement work on this topic is directed at optimal and automatic dependency of a motion system. Lastly, experiments on a framework where data outweighs assumptions on the position-driven feedforward parameter tuning approach, results in a GPs, work especially well. GPs, in combination with a data-driven feedforward approach for LTI/LTV systems is through LQ tracking and stable inversion. Enabling large feedforward tasks on a position-dependent printer, Mechatronics, vol. 38, pp. 76–90, 2016.


Sample Index [-]

[10x49]1

[13x73][9x46]e

[9x53](

[9x59])

[10x45]1

[16x38]∥

[16x49]t

[16x52]∥

[16x58]1

[18x45]0

[28x25]−

[34x25]0

[34x67]0.

[41x25]5

[41x46]0

[41x67]5

[41x88]1

[49x76]techniques such as mutual information optimization.

computation of the training positions using sensor placement work on this topic is directed at optimal and automatic dependency of a motion system. Lastly, experiments on a framework where data outweighs assumptions on the position-driven feedforward parameter tuning approach, results in a GPs, work especially well. GPs, in combination with a data-driven feedforward approach for LTI/LTV systems is through LQ tracking and stable inversion. Enabling large feedforward tasks on a position-dependent printer, Mechatronics, vol. 38, pp. 76–90, 2016.


Sample Index [-]

[10x49]1

[13x73][9x46]e

[9x53](

[9x59])

[10x45]1

[16x38]∥

[16x49]t

[16x52]∥

[16x58]1

[18x45]0

[28x25]−

[34x25]0

[34x67]0.

[41x25]5

[41x46]0

[41x67]5

[41x88]1

[49x76]techniques such as mutual information optimization.

computation of the training positions using sensor placement work on this topic is directed at optimal and automatic dependency of a motion system. Lastly, experiments on a framework where data outweighs assumptions on the position-driven feedforward parameter tuning approach, results in a GPs, work especially well. GPs, in combination with a data-driven feedforward approach for LTI/LTV systems is through LQ tracking and stable inversion. Enabling large feedforward tasks on a position-dependent printer, Mechatronics, vol. 38, pp. 76–90, 2016.