Design and Hardware Implementation of a Controller for Active Damping of a Smart Structure

Oliver Janda[1]   Björn Liebig[2]   Holger Lange[2]   Ulrich Konigorski[1]   Andreas Koch[2]

[1] Laboratory for Control Engineering and Mechatronics
[2] Embedded Systems and Applications Group
Technische Universität Darmstadt, GERMANY
{ojanda,ukonigorski}@iat.tu-darmstadt.de   {liebig,lange,koch}@esa.cs.tu-darmstadt.de

Abstract: Control design for active damping of a smart structure can in principal be tackled using optimal feedback theory which requires a model of the plant that is to be controlled. The main problem when applying modern multichannel feedback control techniques to flexible mechanical structures is the high sensitivity to model uncertainties. This is because of the very low inherent plant damping and the high interactivity of the multiple-input multiple-output (MIMO) systems. Therefore, accurate plant uncertainty descriptions are crucial in order to arrive at robust but not overly conservative control designs. We briefly review a method developed by the authors to derive a non-parametric uncertainty description of the plant model as part of the overall model identification process. Using this method, which is immediately suitable for optimal control design and robust stability analysis, we design a robustly stabilizing $\mathcal{H}_2$ optimal controller running on highly power-efficient FPGA hardware using fixed-point arithmetic. The main focus of this contribution is the wordlength optimization for the fixed-point implementation. For this optimization problem, we employ an optimality metric that reflects the discrepancy between the frequency response of the floating-point and the fixed-point controller implementations. Experimental results show the validity of our approach.

Key–Words: Active damping, Robust control, Smart structure, Wordlength Optimization

1 Introduction

Different control concepts for vibration control have been proposed in the literature. One approach is to use feedback control for active damping. The aim is to devise a control algorithm that effectively mitigates a desired number of structural resonances. This approach is known to be effective especially for broadband noise-like disturbances, but it raises the issue of the stability of the feedback loop, unlike feedforward concepts. Stable feedback control of flexible mechanical structures is not trivial because of two main reasons: First, typical modal damping ratios of less then one percent make the controller design very sensitive to modeling errors. Second, mechanical structures are distributed parameter systems which are naturally described by partial differential equations but most frequently approximated by finite-dimensional models. The modal dynamics that are not incorporated in the finite-dimensional model cause so-called spillover-effects when excited by external disturbances. This deteriorates control performance and can even cause spillover instability, see e.g., [3] and [4]. Despite these difficulties, modern multichannel feedback control concepts have great potential in the control of smart structures because they enable the de-

signer to emulate a structure with better inherent vibration and noise emission properties. This can for example be done by eigenstructure assignment strategies that do not only reduce vibration amplitudes but also taylor vibration mode shapes, see for example [23].

There are several barriers to overcome attempting to apply MIMO optimal control to smart structures: The first one, as already elaborated on, is the sensitivity to model errors. This was the main issue of a recent publication by the first author ([10]). The second hurdle is to formulate the task of active damping as a mathematically tractable optimization problem resulting in an implementable controller. This controller is usually computed based on the assumption of infinite precision calculations, which are well approximated by double-precision floating-point arithmetic. For volume production, however, it is desirable to run on cheap or low-power computing hardware such as ASICs or FPGAs, both of with often rely on fixed-point arithmetic. When going from floating-point to fixed-point representation, errors introduced by finite wordlength effects have to be considered. In this contribution, we briefly review the first two issues, and then focus on the conversion of the $\mathcal{H}_2$ optimal controller from floating-point to fixed-point arith-

metic while maintaining robust stability.

The outline of this paper is as follows: The next section shows how the nominal and the uncertainty model of a flexible mechanical structure can be identified in a form that is immediately suitable for robust optimal control design. We also formulate the optimal control design problem. The main part in Sec. 3 focuses on the wordlength optimization of the state space model of the controller for efficient fixed-point implementation without jeopardizing stability or performance. The last section provides experimental results for the active damping of a panel structure with piezoelectric actuator patches and acceleration sensors with a controller running on FPGA hardware.

2 Identification Procedure and Controller Design

This section is a condensed version of Secs. 2 and 3 of the publication [10] from the first author. It establishes the necessary background for understanding the design of the robustly stabilizing controller for the active damping of the smart structure.

2.1 Identification Procedure

It is assumed throughout that the smart structure to be controlled is stable and can be well described by a linear model. Non-linear material effects, like piezoelectric hysteresis, can be taken care of by compensation techniques such as the one presented in [12]. Then, without loss of generality, the plant to be controlled can be described by a stable discrete-time state-space model

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k), \end{aligned} \qquad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^q$ is the control input, and $\mathbf{y} \in \mathbb{R}^p$ is the output. The time index k stands short for kT_s with sample time T_s. The disturbance input is assumed to be unknown and is not part of the model.

We now present the identification procedure that will be split into two parts. In the first part, a frequency response function (FRF) $\hat{\mathbf{G}}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right)$ of the system is computed in combination with its corresponding confidence intervals at each frequency point. These confidence intervals will be transformed into a bound $\Delta\hat{\mathbf{G}}$ on the difference between the true plant $\mathbf{G}_0$ and the non-parametric model $\hat{\mathbf{G}}$. In the second part, a state-space model will be computed from $\hat{\mathbf{G}}$ and the additive uncertainty bound $\Delta\hat{\mathbf{G}}$ will be modified to incorporate deviations of the parametric from the non-parametric model.

2.1.1 Non-parametric Step

M sets of time domain data each of length $2N$ are collected by applying suitable input signals $\mathbf{u}_m(k) \in \mathbb{R}^q$ and measuring the corresponding output data $\mathbf{y}_m(k) \in \mathbb{R}^p$ with $k = 0, ..., 2N - 1$ and $m = 1, ..., M$. General considerations on suitable input signal and experiment design can for example be found in [17]. In a second step, frequency response functions are computed from the collected data. The DFT spectra of the input and output signals are given by

$$\mathbf{U}_m(\mathrm{j}\omega_n) = T_\mathrm{s} \sum_{k=0}^{2N-1} \mathbf{u}_m(k)e^{-\mathrm{j}\omega_n kT_\mathrm{s}}, \qquad (2\mathrm{a})$$

$$\mathbf{Y}_m(\mathrm{j}\omega_n) = T_\mathrm{s} \sum_{k=0}^{2N-1} \mathbf{y}_m(k)e^{-\mathrm{j}\omega_n kT_\mathrm{s}}, \qquad (2\mathrm{b})$$

with $\omega_n = n\frac{2\pi}{NT_\mathrm{s}}, n = 0, ..., N-1$, and $m = 1, ..., M$. Then, the following input-output relationship holds if $U(\mathrm{j}\omega_n)$ and $Y(\mathrm{j}\omega_n)$ are free of leakage and aliasing effects,

$$\mathbf{Y}(\mathrm{j}\omega_n) = \mathbf{G}_0\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right)\mathbf{U}(\mathrm{j}\omega_n) + \mathbf{V}(\mathrm{j}\omega_n) \qquad (3)$$

with $\mathbf{V}(\mathrm{j}\omega_n)$ being the DFT sequence of the output measurement noise $\mathbf{v}(k) \in \mathbb{R}^p$. Estimates of the cross and auto power spectra of the input and output signals can be derived from the M experiments by

$$\mathbf{S}_{UU}(\mathrm{j}\omega_n) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{U}_m(\mathrm{j}\omega_n)\mathbf{U}_m^\mathrm{H}(\mathrm{j}\omega_n) \qquad (4\mathrm{a})$$

$$\mathbf{S}_{YU}(\mathrm{j}\omega_n) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{Y}_m(\mathrm{j}\omega_n)\mathbf{U}_m^\mathrm{H}(\mathrm{j}\omega_n). \qquad (4\mathrm{b})$$

The so-called H_1 estimate ([21]) of the transfer function is then given as

$$\hat{\mathbf{G}}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right) = \mathbf{S}_{YU}(\mathrm{j}\omega_n)\mathbf{S}_{UU}^{-1}(\mathrm{j}\omega_n). \qquad (5)$$

With (5), we have a non-parametric nominal model. We now compute its confidence interval at each frequency point ω_n. We make the following assumptions for the noise:

Assumption 1 $\mathbf{V}(\mathrm{j}\omega_n)$ *satisfies* $\mathrm{E}\{\mathbf{V}(\mathrm{j}\omega_n)\} = \mathbf{0}$, $\mathrm{E}\{\mathbf{V}(\mathrm{j}\omega_n)\mathbf{V}^\mathrm{T}(\mathrm{j}\omega_n)\} = \mathbf{0}$, $\mathrm{E}\{\mathbf{V}(\mathrm{j}\omega_n)\mathbf{V}^\mathrm{H}(\mathrm{j}\omega_n)\} = \sigma_\mathbf{v}^2(\mathrm{j}\omega_n)$. *Higher order moments are zero. This amounts to* $\mathbf{V}(\mathrm{j}\omega_n)$ *being zero-mean, circular symmetric complex normally distributed. In addition, the Fourier coefficients of the noise sequence are asymptotically independent from each other.*

It is stated in [22] and proven by Brillinger [5] that these requirements are asymptotically satisfied for a wide class of time-domain probability density functions of the noise sequence $\mathbf{v}(k)$, see also Theorem 14.25 in [17]. Considerations on a finite number of samples can be found in [19]. Furthermore, we make the common assumption that the noise is independent and identically distributed (i.i.d.) over the M different experiments and independent of the input.

If this holds, it can be shown that the H_1 estimator is unbiased, $\mathrm{E}\{\hat{\mathbf{G}}\} = \mathbf{G}_0$, and its covariance $\sigma_{\hat{\mathbf{G}}}^2 \in \mathbb{R}^{pq \times pq}$ is given by

$$
\begin{aligned}
&\sigma_{\hat{\mathbf{G}}}^2(\mathrm{j}\omega_n) \\
&= \mathrm{E}\Big\{ \mathrm{vec}\Big\{ \hat{\mathbf{G}}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right) - \mathrm{E}\Big\{ \hat{\mathbf{G}}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right) \Big\} \Big\} \\
&\quad \Big(\mathrm{vec}\Big\{ \hat{\mathbf{G}}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right) - \mathrm{E}\Big\{ \hat{\mathbf{G}}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right) \Big\} \Big\} \Big)^{\mathrm{H}} \Big\} \\
&= \frac{1}{M} \mathbf{S}_{UU}^{-\mathrm{T}}(\mathrm{j}\omega_n) \otimes \sigma_{\mathbf{v}}^2(\omega_n),
\end{aligned}
\tag{6}
$$

where $\otimes$ denotes Kronecker product. In (6), an unbiased estimate of the noise covariance matrix can be used which is given by

$$
\begin{aligned}
\hat{\sigma}_{\mathbf{v}}^2(\mathrm{j}\omega_n) &= \frac{M}{M-q} \\
&\times \left(\mathbf{S}_{YY}(\mathrm{j}\omega_n) - \mathbf{S}_{YU}(\mathrm{j}\omega_n)\mathbf{S}_{UU}^{-1}(\mathrm{j}\omega_n)\mathbf{S}_{UY}(\mathrm{j}\omega_n) \right)
\end{aligned}
\tag{7}
$$

where $\mathbf{S}_{YY}$ and $\mathbf{S}_{UY}$ can be calculated in analogy to (4a) and (4b). The last two results can for example be found in [21].

The variances of the individual SISO transfer functions $\hat{G}_{ij}$, $i = 1, ..., p$, $j = 1, ..., q$ of $\hat{\mathbf{G}}$ can be found on the diagonal of $\sigma_{\hat{\mathbf{G}}}^2$ and will be termed $\sigma_{\hat{G}, ij}^2$ with

$$
\sigma_{\hat{G}, ij}^2(\mathrm{j}\omega_n) = \mathrm{E}\{ |\hat{G}_{ij}(\mathrm{j}\omega_n) - \mathrm{E}\{\hat{G}_{ij}(\mathrm{j}\omega_n)\}|^2 \}.
\tag{8}
$$

By considering (3), it can by concluded that the estimate $\hat{\mathbf{G}}$ is also asymptotically circular symmetric normally distributed. Due to the properties of the circular symmetric distribution, the variances of the respective real and imaginary parts are

$$
\sigma_{\Re, ij}^2(\mathrm{j}\omega_n) = \sigma_{\Im, ij}^2(\mathrm{j}\omega_n) = \frac{1}{2}\sigma_{\hat{G}_{ij}}^2(\mathrm{j}\omega_n).
\tag{9}
$$

From these variances, additive uncertainty descriptions can be constructed. An additive uncertainty description for some transfer function $\hat{G}_{ij}$ states that

$$
G_{0,ij}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right) \in \hat{G}_{ij}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right) + \Delta\hat{G}_{ij}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right)
\tag{10}
$$

where the possible values of $\Delta\hat{G}_{ij}$ are bounded in some way. We choose the bounds to be

$$
\Delta\hat{G}_{ij,\max}\left(e^{\mathrm{j}\omega_n T_\mathrm{s}}\right) := \frac{1}{\sqrt{2}}n(\alpha)\sigma_{\hat{G}_{ij}}(\mathrm{j}\omega_n)(1 + \mathrm{j})
\tag{11}
$$

where $n(\alpha)$ is a real number that is determined by the desired confidence level α. The uncertain quantity $\Delta\hat{G}_{ij}$ can be any complex number where the absolute value, taken separately for the real and imaginary part, takes at most the value of the real or imaginary part of $\Delta\hat{G}_{ij,\max}$. The rationale behind equations (9) to (11) is illustrated in Fig. 1. The variance $\sigma_{\hat{G}_{ij}}^2$ defines a circle of radius $n(\alpha)\sigma_{\hat{G}_{ij}} = \left|\Delta\hat{G}_{ij}\right|$ where the true plant $\hat{G}_{0,ij}$ can be expected to be located with confidence level α,

$$
\Re\{G_{0,ij} - \hat{G}_{ij}\}^2 + \Im\{G_{0,ij} - \hat{G}_{ij}\}^2 \leq |\Delta\hat{G}_{ij}|^2 \ \text{w.p. } \alpha.
\tag{12}
$$

This circle is approximated by the additive uncertainty description as the square region that touches the circle, where the side length of the square is given by $2\Re\left\{\Delta\hat{G}_{ij,\max}\right\} = 2\Im\left\{\Delta\hat{G}_{ij,\max}\right\}$. Since the real and imaginary part of some transfer function are jointly normally distributed and uncorrelated, they are also statistically independent. Therefore, the statement in (10) is true with probability α^2.

The elements $\Delta\hat{G}_{ij,\max}$ taken together define an *uncertain* complex matrix $\Delta\hat{\mathbf{G}}$ where, again, the absolute value of each entry, taken separately for the real and imaginary parts, is allowed to take at most the value defined in (11).

To ensure that *all* elements of the transfer matrix $\mathbf{G}_0$ are captured by the nominal FRF and its additive uncertainty description, at least at a desired probability level α^*, one can make use of Bonferroni's inequality [14]. It states for this case that

$$
\mathrm{Pr}\left(\mathbf{G}_0 \in \hat{\mathbf{G}} + \Delta\hat{\mathbf{G}}\right) \overset{!}{=} \alpha^* \geq 1 - pq(1 - \alpha^2).
\tag{13}
$$

Assume, for example, a system with four inputs and four outputs using $n(\alpha) = 3$ standard deviations ($\alpha = 99.7\,\%$). One can calculate that (13) is satisfied with a joint probability of more than $90.4\,\%$.

2.1.2 Parametric Step

In the previous paragraph, we derived a non-parametric nominal model (5) and a frequency-wise bound on the difference of true and nominal plant (11). We now compute a parametric nominal model that can be utilized for model based controller design. In principal, any suitable identification algorithm can
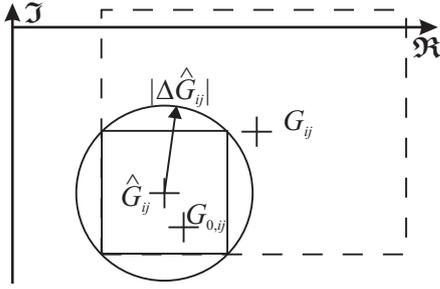
Figure 1: Extension of the uncertainty description to account for the differences of non-parametric model $\hat{\mathbf{G}}$ and parametric model $\mathbf{G}$

be used. In our numerical example in Sec. 4, we make use of the frequency-domain ERA-OKID algorithm presented in [11]. It takes the FRFs (5) as input data and fits a model of the form (1). The resulting model $\mathbf{G}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ is a minimal realization, i.e. the model is controllable and observable.

We now assume that such a nominal state-space model of desired order is available. In a last step, we have to account for the differences between the FRFs of the non-parametric model (5) and the parametric model $\mathbf{G}(z)$. These differences are inevitable in practice, because extremely high model orders would have to be used to perfectly fit the FRF data. More importantly, moderate model orders are desired to alleviate numerical difficulties in the subsequent computation of the controller. Usually, the limitation of the model order leads to a neglect of high frequency dynamics, because a limited set of model parameters is used to fit the FRF data as well as possible in the desired control loop bandwidth. The model will therefore be fairly inaccurate above the control loop bandwidth which increases the danger of spillover effects.

We propose to incorporate the differences of the non-parametric and the parametric model by extending the uncertainty region by the amount of the difference of the real and imaginary part, respectively,

$$\Delta G_{ij,\max} := \left(\left| \Re\left\{ G_{ij} - \hat{G}_{ij} \right\} \right| + \frac{1}{\sqrt{2}} n(\alpha)\sigma_{\hat{G}_{ij}} \right)$$
$$+ \mathrm{j}\left(\left| \Im\left\{ G_{ij} - \hat{G}_{ij} \right\} \right| + \frac{1}{\sqrt{2}} n(\alpha)\sigma_{\hat{G}_{ij}} \right) \tag{14}$$

for all transfer functions $i = 1, ..., p$, $j = 1, ..., q$ and all frequencies $\omega_n = 0, ..., \frac{N-1}{N}\frac{2\pi}{T_s}$. This is also illustrated in Fig. 1.

As a result of the identification procedure, we derived a nominal parametric model (1) and a non-parametric additive uncertainty description given by the uncertain complex matrix $\Delta\mathbf{G}$ where the absolute value of each entry is bounded by (14). We can now
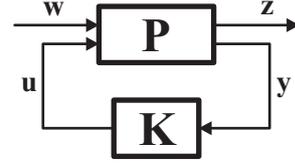
Figure 2: Closed loop with generalized plant and controller
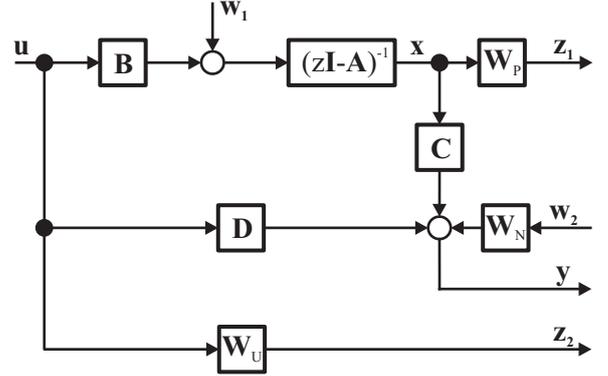
Figure 3: Generalized plant $\mathbf{P}$

define the set of perturbed plants

$$\Pi := \left\{ \mathbf{G}^*(z) \ \middle| \ \mathbf{G}^*\left(e^{\mathrm{j}\omega_n T_{\mathrm{s}}} \right) \in \mathbf{G}\left(e^{\mathrm{j}\omega_n T_{\mathrm{s}}} \right) + \Delta\mathbf{G}\left(e^{\mathrm{j}\omega_n T_{\mathrm{s}}} \right) \right\}. \tag{15}$$

This set can be utilized in standard robust control theory as demonstrated in the next section.

2.2 Controller Design

The results of Sec. 2.1 are immediately applicable for the robust optimal control design to achieve active damping of a flexible structure. The general goal of the $\mathcal{H}_2$ optimal design is to compute a controller $\mathbf{K}$ for some generalized plant $\mathbf{P}$, see Fig. 2, that minimizes the $\mathcal{H}_2$ norm of the transfer function from some exogenous input $\mathbf{w}$ to the performance variables $\mathbf{z}$. We propose the generalized plant $\mathbf{P}$ displayed in Fig. 3.

Since the input matrix of the disturbance forces acting on the structure is generally unknown, the disturbance $\mathbf{w}_1$ is modeled as acting directly on the states. This is equivalent to setting its input matrix to the identity matrix. The second exogenous input $\mathbf{w}_2$ represents measurement noise which is weighted relatively to $\mathbf{w}_1$ by $\mathbf{W}_N$. The first performance variable $\mathbf{z}_1$ is some linear combination of the system states introduced by the performance weight $\mathbf{W}_P$. The limitation of the controller gain is realized by passing the control signal $\mathbf{u}$ through a suitable weighting function $\mathbf{W}_U$ and defining it as a second performance variable $\mathbf{z}_2$. Instructions for the design of the weighting functions $\mathbf{W}_P$, $\mathbf{W}_U$ and $\mathbf{W}_N$ can be found in [10].

Once the generalized plant and the weighting functions are defined, one needs to solve the following $\mathcal{H}_2$ optimization problem:

Find an internally stabilizing controller $\mathbf{K}(z)$ for the control-loop configuration in Fig. 2 that minimizes the performance index $\|\mathbf{P}_{zw}\|_2 = \left(\frac{1}{2\pi} \mathrm{tr} \left[\int_{-\pi}^{\pi} \mathbf{P}_{zw} \left(e^{j\omega_n T_s} \right) \mathbf{P}_{zw}^{H} \left(e^{j\omega_n T_s} \right) \mathrm{d}\omega \right] \right)^{\frac{1}{2}}$.

Instructions for the solution to this problem can be found in standard literature on optimal control theory, e.g., [18]. In our design, we made the additional restriction that the controller $\mathbf{K}$ should be strictly proper. This has the advantage that a complete sample period is available for calculation of the controller output. The controller is represented by a discrete-time state space model with matrices $\mathbf{A}_c$, $\mathbf{B}_c$ and $\mathbf{C}_c$ running with a sampling period of $T_s = 0.6$ milliseconds.

In [10], it is also elaborated on how the identified additive uncertainty $\Delta \mathbf{G}$ can be utilized to check for robust stability via standard tests like Small Gain Theorem or μ-Analysis.

3 Digital Implementation

With the theory of the controller now established, we continue to examine how it can be efficiently realized using hardware suitable for volume production. Low-cost and/or low-power embedded computing generally replaces floating-point by fixed-point computation. However, the previously discussed robustness must still be maintained. Thus, a trade-off must be found between computation accuracy and device costs as well as power consumption. Fixed point arithmetic is often used since its operators (especially addition) require less transistors than floating point operators of the same bit width. While in floating point arithmetic, numbers are represented by a fixed number of digits scaled at run-time using a variable exponent, fixed point representation uses a fixed exponent chosen at design time which remains constant at run-time. The latter is considered more efficient in applications where the high dynamic range of floating point arithmetic is not required. Hence, fixed point arithmetic is especially beneficial for systems implemented as application specific integrated circuit (ASIC) or in a Field-programmable Gate Array (FPGA). An FPGA is an integrated circuit designed to be (re)configured with sequential digital logic functions *after* the silicon device itself has been manufactured . The logic circuits are implemented by the FPGA fabric using lookup tables (LUTs) with associated flip-flops as well as programmable interconnects. These basic logic structures are complemented with special function blocks such as multipliers, DSP units, I/O transceivers, or on-chip memories.

To convert an infinite word length (or floating point) model to fixed point arithmetic, all operands must be quantized while limiting the quantization error to bounds acceptable for the specific application. The quantization error in the data path of the controller, which basically consists of input (A/D converter), output (D/A converter), and state vector, can be modeled with white noise injection as long as the quantization error is small compared to the signal. The variance of the injected noise σ^2 is a function of the fractional bit length n_F, [16]

$$\sigma^2 = \frac{1}{12} \cdot 2^{-2n_F}. \tag{16}$$

On this basis, several analytic and simulation-based approaches have been developed to identify the optimal word length of data path variables [7] [13] [9]. As long as the quantization error remains small enough, their quantization effect on the FRF can be neglected.

Beyond the data path, the coefficients of the controller matrices must be quantized as well. They are known at design time, but even small changes due to quantization have direct influence on the frequency response. Hence, the FRF change needs to be observed during quantization of the coefficients. To find the optimal quantization, we propose the min+b search algorithm [6] using the mean square frequency response discrepancy as quality metric.

The min+b algorithm is an heuristic method for finding (near) optimal fractional word lengths of a set of fixed point variables. The optimum is defined as the word length configuration with the smallest sum of word lengths that still satisfies the quality metric. Common quality metrics aim for a limited mean square error of system output (compared to infinite word length system) or a limited quantization noise. The min+b algorithm uses two passes to determine the word length. The first pass starts with a near-infinite world length and estimates a lower bound:

1. Set all variables to the maximum word length supported by the architecture.

2. Pick one variable not yet optimized and decrease its word length until the quality measure is violated.

3. Store the smallest working bit size for this variable and continue with step 1 until all variables have been optimized.

The second pass subsequently uses the given lower bound estimates and increases the individual word lengths to find a working configuration:

1. Set $b = 1$.

2. Check all combinations of additional b bits distributed over all variables.

3. If a working configuration was found, terminate.

4. If $b < b_{max}$ increase b and go back to 2.

5. If $b = b_{max}$ use configuration with best quality score as new base and go back to step 1.

Smaller values of b_{max} will reduce the run-time but result in a higher probability to miss the optimal configuration. Cantin compares the min+b algorithm to several other methods in [6]. In some cases, these methods computed small widths than the min+b lower bound estimates. But min+b remains attractive by providing a good trade-off between quantization quality and run-time.

For coefficient quantization, we choose the mean square frequency response discrepancy as quality metric J:

$$J = \sum_{\omega_n=0}^{\frac{N-1}{N}\frac{2\pi}{T_s}} |G\left(e^{j\omega_n T_s}\right) - \hat{G}\left(e^{j\omega_n T_s}\right)|^2. \quad (17)$$

Due to the MIMO structure of the controller described in this paper, multiple frequency responses must be calculated. Two effects, which will be described in the following paragraphs, need to be considered when adapting the min+b algorithm for coefficient quantization using the frequency response quality metric.

First, when relying on injected noise as quality metric, only the fractional part is optimized since reducing the word length starting from the most-significant bits will not allow the use of white noise error model (the most-significant bits cannot be considered white noise). In contrast, when examining the frequency response, the coefficients *can* be optimized at both most- and the least-significant sides. Here, both quantization effects (saturation and rounding) will affect the frequency response. Quantization of the integer part is simplified by the knowledge of coefficients during optimization. Thus, the required integer word length can be calculated analytically. If a small saturation error is acceptable, the most-significant bit remains for further optimization.

However, in some cases the integer word length can depend on the fractional word length, i.e., removing bits from the fractional part can cause a larger error in the integer part. As an example, assume the optimization of the decimal unsigned value 1.99 with a maximum allowed error of decimal 0.01. In its original binary 16-bit representation of 0001.111111010111 only one bit is required for the integer part. However, restricting the integer part's word length to one bit *before* optimization the fractional word length would require six bits for the fractional part, since rounding up to a value of two (which would take two bits in the integer part) is no longer possible. To avoid this potential source of suboptimal word length results for the integer part, Phase 1 of the min+b algorithm was slightly modified to always optimize the fractional part first, and reuse the fractional result during the optimization of the integer part.

Second, since the less significant bits in the data path are generally not correlated to each other, injected noise can be modeled as zero mean white noise as stated earlier. Equation (16) implies that noise intensity is a monotonically decreasing function of word length of the fractional part . Hence, the first phase of min+b algorithm stops shrinking the word lengths as soon as a length fails to satisfy the quality criterion. No such correlation can be proven for the frequency response error. In practice, however, we have not observed a single case in numerous experiments where shrinking the word length lead to a *decrease* of the frequency response error. Our heuristics thus use the first violation of frequency response quality constraints as a termination criterion for shrinking the word lengths.

The algorithm was implemented in MATLAB-Simulink targeting a high-level hardware synthesis tool flow for FPGAs. However, it can also be used for fixed-point software implementations by limiting the legal word lengths to just 8, 16 or multiples of 32 bit. Due to a limitation in the Simulink gain block quantization setting, only a single word length can be specified for all elements of the matrix. This is a limitation acceptable for both software and hardware realizations: In software all coefficients of a matrix will be stored in a single array, and hence must all have the same data type. In hardware on the FPGA, our compute performance is sufficiently high that, at the required sample rate, we can time-multiplex the computations for *all* matrix elements onto a *single* hardware-operator (conserving hardware area), and then just size the fractional and integer parts for that single operator.

For this application, the tolerable mean square absolute error in frequency response was defined as 10^{-5} to closely match the FRF of the floating point controller. The quantization results for the controller described in this paper can be found in table 1. Compared to the MATLAB default floating point format (64-bit IEEE-754 double), 49 to 47 bits (74–77%) could be saved in addition to gaining the performance benefits of using fixed point arithmetic.

Table 1: Word length (WL) after optimization

matrix	state $\mathbf{A}_C$	input $\mathbf{B}_C$	output $\mathbf{C}_C$
integer WL	2	1	2
fractional WL	15	14	13
matlab std.	64	64	64
reduction	74%	77%	77%
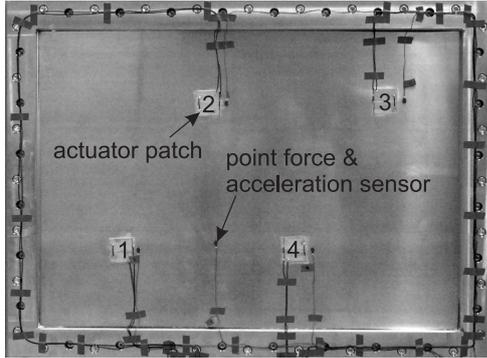
Figure 4: Layout of the smart panel

4 Experimental Results

This section provides experimental results for the identification, control design and fixed-point implementation algorithms presented in the last two sections.

The smart structure to be considered is a flexible panel with four bonded piezo patches which are used as actuators to apply bending moments onto the plate to counteract disturbance forces. The vibratory motion of the plate is measured with four piezoelectric acceleration sensors nearly collocated to the actuators. The layout of the smart panel can be seen in Fig. 4. In addition to the actuators and sensors on the frontside, an electromagnetic shaker with mounted force sensor excites the plate from the backside with a disturbance force. There is a fifth acceleration sensor collocated to the excitation point on the frontside which is used for test purposes.

A plant model $\mathbf{G}(z)$ was identified with the actuator voltages as inputs and the signals of the collocated acceleration sensors as outputs. The spectral density of the logarithmically accelerated swept sine signal which was used to excite the first 15 modes of the structure up to 500 Hz is displayed in Fig. 5. The spectrum of a typical corresponding acceleration signal is shown in Fig. 6.

The ERA-OKID algorithm was used to fit a parametric model $\mathbf{G}(z)$ of order $n = 197$ to the FRF $\hat{\mathbf{G}}\left(e^{j\omega_n T_s}\right)$ in the first instance which was subsequently reduced via frequency weighted balanced truncation to a model of order 100. The magnitude
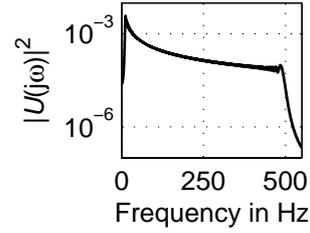
Figure 5: Spectrum of the swept sine input signal
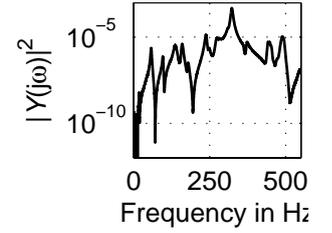
Figure 6: Typical spectrum of the output signal
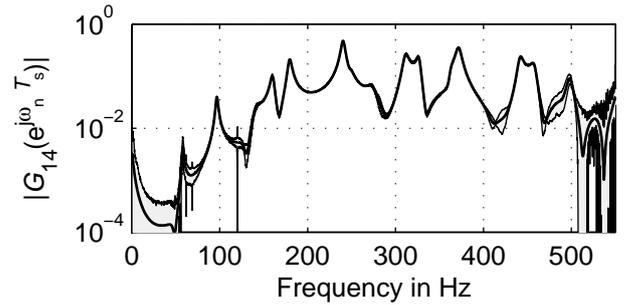
Figure 7: Frequency response of the parametric model from fourth input to first output along with its confidence region for $\alpha = 99.7\%$

of the frequency response of the reduced order model from the fourth input to the first output (compare Fig. 4) is presented in Fig. 7 along with its 99.7 % confidence band. There is a region of high model uncertainty below 50 Hz, which is caused by the low SNR of the sensor signals. Model uncertainty is also significant above 500 Hz which is due to the drop in the spectral density of the input signal.

Based on the nominal model, an $\mathcal{H}_2$ optimal controller of order 60 was designed according to the procedure outlined in Sec. 2.2 in order to damp the first twelve bending modes of the plate. Robust stability was checked via μ-analysis based on the uncertainty description computed in Sec. 2.1, see Fig. 8. The maximum value of the structured singular value μ over the relevant frequency band is 1.05, which is
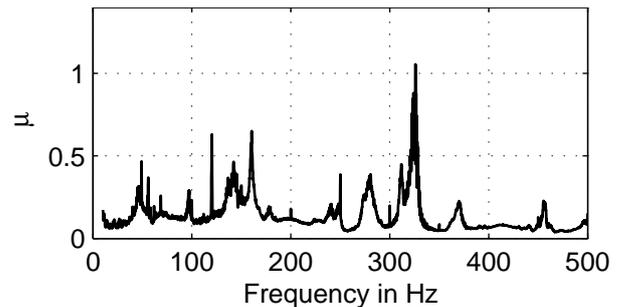
Figure 8: Result of μ-analysis

slightly greater than the theoretically allowed maximum of one. Nevertheless, the controller can be regarded to be robustly stable since the bound on the uncertainty is conservative. Especially, step (14) is conservative which can be seen in Fig. 1.

To evaluate the performance of the controller, the frequency responses from the disturbance force to the collocated acceleration sensor and the four control sensors are depicted in Fig. 9. It can be seen that the resonance peaks of the first twelve modes in the targeted frequency range up to 400 Hz are significantly damped. Furthermore, the frequency response above 400 Hz is unchanged, which shows that there are neglible spillover effects in the control loop.

To evaluate the impact of our word length optimization procedure and the resulting hardware implementation, the controller data path was quantized using the min+b algorithm with injected noise as quality metric, while the coefficients were word-length optimized by focusing on the frequency response. The resulting finite word length controller was then formulated manually as a high-level hardware description using the Synphony HLS Block Set [20] for Simulink and successfully mapped to the Actel IGLOO M1AGL1000 Low-Power FPGA [1] [8]. This would not have been possible without our word length optimizations, since the IGLOO series is a power-optimized family of small FPGAs without integrated multiplier units unable to efficiently perform double precision floating point arithmetic. To compare the efficiency of the FPGA solution with a pure software approach, a software implementation of the controller was generated by exporting the controller as C code using MATLAB Real-Time Workshop [15], which was then compiled for an Texas Instruments TMS320C5515 16-bit Low-Power DSP with all optimizations enabled [2]. The power draw and computation time was measured on both devices to identify the energy consumption per sample. While the FPGA requires only 15.7 μJ per sample, the DSP consumes 140 μJ due to much longer calculation times. This large difference may be due to some word lengths slightly exceeding the 16 bit architecture of the DSP, thus requiring inefficient composite 32 bit operations.

5   Conclusion

Model based optimal control techniques were applied to actively damp a flexible structure. We showed how an uncertainty description of the plant can be derived, from the data measurement to the checking for robust stability of the control loop.

We then introduced a novel world length optimization heuristic to derive an area-efficient fixed-

point computing architecture for the controller. This could then be mapped to a low-power FPGA using high-level descriptions, without the need to resort to low-level digital hardware design languages such as Verilog or VHDL. The resulting compute unit is considerably more power efficient than even a low-power DSP and fulfills both the quality and sample rate requirements.

The entire system was evaluated using a lab experiment consisting of a smart panel with four sensors and four actuators, where it was able to successfully dampen the resonance peaks below 400 Hz.

References:

[1] Igloo handbook. `http://www.actel.com/products/IGLOO`.

[2] TMS320C55x Optimizing C/C++ Compiler User's Guide (Rev. F) 31 Dec 2003. `http://www.ti.com/litv/pdf/spru281f`.

[3] M. J. Balas. Active Control of Flexible Systems. *Journal of Optimization Theory and Applications*, 25(3):415–436, 1978.

[4] J. Bontsema and R. F. Curtain. A Note on Spillover and Robustness for Flexible Systems. *IEEE Transactions on Automatic Control*, 33(6):567–569, 1988.

[5] D. R. Brillinger. *Time series: Data analysis and theory*. Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1981.

[6] M. A. Cantin, Y. Savaria, and Lavoie P. A comparison of automatic word length optimization procedures. In *IEEE International Symposium on Circuits and Systems (ISCAS 2002)*, 2002.

[7] G. A. Constantinides, P. Y. K. Cheung, and Luk. W. Wordlength optimization for linear digital signal processing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(10):1432–1442, 2003.

[8] A. Engel, B. Liebig, and A. Koch. Feasibility analysis of reconfigurable computing in low-power wireless sensor applications. In A. Koch, R. Krishnamurthy, J. McAllister, R. Woods, and T. El-Ghazawi, editors, *Reconfigurable Computing: Architectures, Tools and Applications*, volume 6578 of *Lecture Notes in Computer Science*, pages 261–268. Springer Berlin / Heidelberg, 2011.

[9] Z. Guoming, K. M. Grigoriadis, and R. E. Skelton. Optimal finite wordlength digital control with skewed sampling. In *American Control Conference 1994*, pages 3482–3486, 1994.

[10] O. Janda and U. Konigorski. Identification and Robust Active Damping of a Smart Structure.
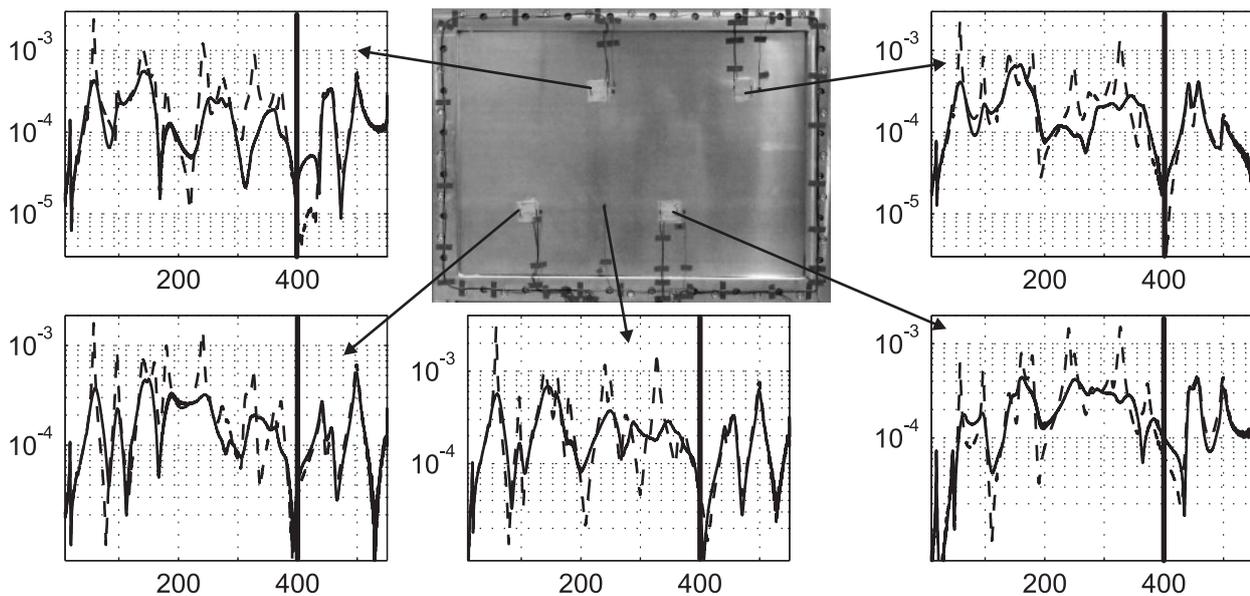
Figure 9: Frequency responses from disturbance force to collocated acceleration sensor and to the four sensors used in the control loop. x-Axis: Frequency in Hz; dashed line: controller off; solid line: controller on

In *4th WSEAS Conference on Engineering Mechanics, Structures and Engineering Geology (EMESEG 2011)*, pages 508–516, 2011.

[11] J. N. Juang. *Applied system identification.* Prentice Hall, Englewood Cliffs, NJ, 1994.

[12] K. Kuhnen and P. Krejci. Compensation of complex hysteresis and creep effects in piezoelectrically actuated systems: A new Preisach modeling approach. *IEEE Transactions on Automatic Control*, 54(3):537–550, 2009.

[13] K. I. Kum and S. Wonyong. Combined word-length optimization and high-level synthesis of digital signal processing systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(8):921–930, 2001.

[14] E. B. Manoukian. *Modern Concepts and Theorems of Mathematical Statistics.* Springer, New York, 1986.

[15] Mathworks. Real-Time Workshop. `http://www.mathworks.com/products/rtw`.

[16] A. V. Oppenheim and C. J. Weinstein. Effects of finite register length in digital filtering and the fast Fourier transform. *Proceedings of the IEEE*, 60(8):957–976, 1972.

[17] R. Pintelon and J. Schoukens. *System identification: A frequency domain approach.* IEEE Press, Piscataway, NJ, 2005.

[18] A. Saberi, B. M. Chen, and P. Sannuti. $\mathcal{H}_2$ *Optimal Control.* Prentice Hall, London, 1995.

[19] J. Schoukens and J. Renneboog. Modeling the noise influence after a discrete Fourier transform. *IEEE Transactions on Instrumentation and Measurement*, 35(3):278–286, 1986.

[20] Synopsys. Synphony Model Compiler. `http://www.synopsys.com/SYSTEMS/BLOCKDESIGN/HLS/Pages/Synphony-Model-Compiler.aspx`.

[21] P. Verboven. *Frequency-Domain System Identification for Modal Analysis.* PhD thesis, Vrije Universiteit Brussel, Brussels, 2002.

[22] E. Wernholt and S. Gunnarson. Analysis of Methods for Multivariable Frequency Response Function Estimation in Closed Loop. In *IEEE Conference on Decision and Control, New Orleans, USA, Dec. 12-14*, pages 4881–4888, 2007.

[23] T. Y. Wu and K. W. Wang. Reduction of Structural Acoustic Radiation Via Left and Right Eigenvector Assignment Approach. *Journal of Intelligent Material Systems and Structures*, 20:2173–2186, 2009.