2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

A Least Squares Approach to Direct Frequency Response Estimation

Per Hägg, Håkan Hjalmarsson and Bo Wahlberg

Abstract— **Traditionally, the frequency response function has been estimated directly by dividing the discrete Fourier transforms of the output and the input of the system. This approach suffers from leakage errors and noise sensitivity. Lately these errors have been studied in detail. The main observation is that the error has a smooth frequency characteristic that is highly structured. The recently proposed local polynomial method uses this smoothness, and tries to estimate the frequency response function along with a smooth approximation of the error term. In this paper we propose a method, closely related to the local polynomial method, but instead of using the smoothness of the error we explore the structure even further.**

The proposed approach to estimate the frequency response function seems promising, as illustrated by simulations and comparison with current state of the art methods.

I. Introduction

In most system identification problems, the goal is to estimate a parametric model of a system from input and output data. It is, however, common to first identify a non-parametric model of the frequency response function (FRF) of the system to get a feel for the complexity of the problem at hand. Frequency response functions have, of course, many other applications and are used intensively in many engineering fields, for example in audio applications, power systems and vibration analysis.

Traditionally the estimate has been calculated with the empirical transfer function estimate (ETFE) [1], [2], *i.e.,* directly use the discrete Fourier transform. This approach has however some well known inherent drawbacks. Due to finite-length data record and non-periodic signals the estimates suffer from leakage errors, frequency content at one frequency "leaks" into the neighboring frequencies. Also, as there is no data compression, the estimate suffers from noise sensitivity.

One way to cope with these problems is to window the data with some window function [3]. This smoothes the frequency response function over neighboring frequencies. This gives, however, rise to interpolation errors.

In recent years, this leakage error has been studied in detail. The main observation is that the leakage error is highly structured with a smooth frequency characteristic [4], [5]. This smoothness of the error is used in the recently proposed local polynomial method [6], where it is approximated with a truncated Taylor series expansion. A least squares problem is formed and the FRF together with the Taylor series coefficients are estimated. The problem is then solved locally in every frequency, estimating new Taylor coefficients for every local least squares problem. This approach has been shown to be superior over window methods [6].

The main contribution of this paper is to develop a new method of identifying the FRF of linear systems. It is closely related to the local polynomial method but the main difference is that it uses the fact that the structure of the leakage error is known. Exploring this insight leads to one large least squares problem instead of many local problems. By way of extensive simulations we show that this is a promising approach.

The outline of the paper is as follows. In Section II we will generalize the transient error term. The generalization is then used to set up a regression problem in Section III. Section IV compares the proposed method with the related local polynomial method. A numerically efficient way to solve the regression problem is shown in Section V. The performance of the method is verified through simulations in Section VI, while Section VII concludes the paper.

II. Generalization of transient term

Consider the asymptotically stable linear MIMO system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t \end{aligned} \tag{1}$$

where the inputs and outputs are $u_t \in \mathbb{R}^{n_u}$ and $y_t \in \mathbb{R}^{n_y}$. The initial state of the system is x_0.

If we have N samples of the input and output of system (1) then applying the N-point DFT

$$Z_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N} z_t e^{-j\omega t}$$

gives [5]

$$Y_N(\omega_k) = G(e^{j\omega_k})U_N(\omega_k) + \frac{1}{\sqrt{N}} T(e^{j\omega_k}) \tag{2}$$

where

$$T(e^{j\omega_k}) = e^{j\omega_k} C(e^{j\omega_k} I - A)^{-1}(I - A^N)(x_0 - x_p)$$

$$x_p = (I - A^N)^{-1} \sum_{t=0}^{N-1} A^t Bu(N-1-t).$$

where x_p is the initial state that would lead to a periodic output, *i.e.,* $x_N = x_p$. This expression is valid for the frequencies $\omega_k = \frac{2\pi k}{N}, \quad k = 0, \dots, N-1$. The extra term $T(e^{j\omega_k})$ is zero for periodic signals. See [5] for more details.

A generalization of this expression that is valid for all frequencies ω is given in the following lemma:

Lemma 1 (Generalization of transient term): Suppose that both the N samples of the input and output of the

This work was partially supported by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement No. 267381, the Swedish Foundation for Strategic Research, the Swedish Research Council and the Linnaeus Center ACCESS at KTH

The authors are with the Automatic Control Lab and ACCESS, School of Electrical Engineering, KTH, SE-100 44 Stockholm, Sweden. E-mail: {pehagg, hjalmars, bo}@kth.se

978-1-61284-799-3/11/$26.00 ©2011 IEEE

system (1) are zero-padded with $2JN$ zeros, with J an integer. Then the relation between the $N_e = (2J+1)N$-point DFT of the zero-padded input, $U_{N_e}(\omega)$, and output, $Y_{N_e}(\omega)$, is

$$Y_{N_e}(\omega) = G_0(e^{j\omega})U_{N_e}(\omega) + Y_N^{tra}(\omega) + \frac{1-e^{-j\omega N}}{\sqrt{N}} Y_p(\omega) \tag{3}$$

where Y_N^{tra} is the DFT of the transient due to the fact that $x_0 \neq x_p$, *i.e.,*

$$\begin{aligned} Y_N^{tra}(\omega) &\triangleq \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k^{tra} e^{-j\omega k} \\ y_k^{tra} &= CA^k(x_0 - x_p) \in \mathbb{R}^{n_y \times 1} \end{aligned}$$

and

$$\begin{aligned} Y_p(\omega) &\triangleq \sum_{k=0}^{\infty} y_k^p e^{-j\omega k} \\ y_k^p &= CA^k x_p \in \mathbb{R}^{n_y \times 1} \end{aligned}$$

is just the Discrete-Time Fourier Transform (DTFT) of the system response when $x(0) = x_p$ and $u \equiv 0$.

Proof: See Appendix I. ■

The expression (3) is, unlike (2), valid for all frequencies ω. It collapses to (2) for $\omega = 2\pi k/N, k = 0, 1, 2, \ldots$.

III. Estimation of frequency response function

In this section we use the generalized expression for the transient term to form a regression problem to estimate the frequency response function.

A. Regression problem

We start by looking at the relation between $G_0(e^{j\omega})$ and $G_0(e^{j\bar{\omega}})$ for two given frequencies ω and $\bar{\omega}$. The difference can be expressed as

$$\begin{aligned} &G_0(e^{j\omega}) - G_0(e^{j\bar{\omega}}) \\ &= \sum_{k=0}^{\infty} CA^k B e^{-j\omega(k+1)} - \sum_{k=0}^{\infty} CA^k B e^{-j\bar{\omega}(k+1)} \\ &= \sum_{k=0}^{\infty} CA^k B (e^{-j\omega(k+1)} - e^{-j\bar{\omega}(k+1)}) \\ &= \sum_{k=1}^{\infty} g_k \varphi_k(\omega, \bar{\omega}) \end{aligned} \tag{4}$$

where $g_k = CA^{k-1}B \in \mathbb{R}^{n_y \times n_u}, k = 1, 2, \ldots$ are the impulse response coefficients of the system and $\varphi_k(\omega, \bar{\omega}) = e^{-j\omega k} - e^{-j\bar{\omega}k}$.

Inserting (4) into (3) gives

$$\begin{aligned} Y_{N_e}(\omega) &= \left(G_0(e^{j\bar{\omega}}) + \sum_{k=1}^{\infty} g_k \varphi_k(\omega, \bar{\omega}) \right) U_{N_e}(\omega) \\ &+ Y_N^{tra}(\omega) + \frac{1-e^{-j\omega N}}{\sqrt{N}} Y_p(\omega) \\ &= G_0(e^{j\bar{\omega}})U_{N_e}(\omega) + \sum_{k=1}^{\infty} g_k \varphi_k(\omega, \bar{\omega}) U_{N_e}(\omega) \\ &+ \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k^{tra} e^{-j\omega k} + \frac{1-e^{-j\omega N}}{\sqrt{N}} \sum_{k=0}^{\infty} y_k^p e^{-j\omega k}. \end{aligned} \tag{5}$$

We want to use (5) to form a regression problem. The goal is to estimate $G_0(e^{j\omega})$ for the frequencies $\omega = \frac{2\pi s}{N}$, $s = 0, \ldots, N-1$ and hence it is natural to set

$$\bar{\omega} = \omega_{s,0} \triangleq \frac{2\pi s}{N}.$$

If we use (5) directly to set up the regression problem we would have to estimate infinitely many terms $G_0(e^{j\omega_{s,0}})$, y_k^{tra}, y_k^p and g_k from finite data. Thus we have to make some approximations.

The terms in the sums $y_k^{tra} = CA^k(x_0 - x_p)$, $y_k^p = CA^k x_p$ and $g_k = CA^{k-1}B$ all tends to zero when k tends to infinity since we have assumed an asymptotically stable system. Therefore the sums are approximated with the following truncations

$$\begin{aligned} Y_N^{tra} &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k^{tra} e^{-j\omega k} \approx \frac{1}{\sqrt{N}} \sum_{k=0}^{n_1-1} y_k^{tra} e^{-j\omega k} \\ Y_p(\omega) &= \sum_{k=0}^{\infty} y_k^p e^{-j\omega k} \approx \sum_{k=0}^{n_2-1} y_k^p e^{-j\omega k} \\ &\sum_{k=1}^{\infty} g_k \varphi_k(\omega, \bar{\omega}) \approx \sum_{k=1}^{n_3} g_k \varphi_k(\omega, \bar{\omega}). \end{aligned}$$

Finally it has to be decided for which frequencies the regression problem will be set up in. We will use the $2L$ neighboring frequencies to $\omega_{s,0}$, *i.e.,*

$$\omega = \omega_{s,l} \triangleq \frac{2\pi}{(2J+1)N}((2J+1)s + l), l = -L, \ldots L$$

where J is the over-sampling factor. The number of neighboring frequencies L is a design parameter and has to be chosen by the user.

With these approximations the regression problem becomes

$$\begin{aligned} \hat{Y}_{N_e}(\omega_{s,l}) &= G_0(\omega_{s,0})U_{N_e}(\omega_{s,l}) \\ &+ \frac{1}{\sqrt{N}} \sum_{k=0}^{n_1-1} y_k^{tra} e^{-j\omega_{s,l}k} \\ &+ \frac{1-e^{-j\omega_{s,l}N}}{\sqrt{N}} \sum_{k=0}^{n_2-1} y_k^p e^{-j\omega_{s,l}k} \\ &+ \sum_{k=1}^{n_3} g_k \varphi_k(\omega_{s,l}, \omega_{s,0}) U_{N_e}(\omega_{s,l}), \\ l &= -L, \ldots, L. \end{aligned} \tag{6}$$

By letting $s = 0, \ldots, N-1$, $(2L+1)N$ equations in the $N + n_1 + n_2 + n_3$ unknowns $\{G_0(\omega_{s,0}\}_{s=0}^{N-1}$, $\{y_k^{tra}\}_{k=0}^{n_1-1}$, $\{y_k^p\}_{k=0}^{n_2-1}$ and $\{g_k\}_{k=1}^{n_3}$ are obtained. Solving this least squares problem gives estimates of all aforementioned variables; in particular the estimate of the entire frequency response is obtained.

B. Choosing parameter values

The performance of the method is highly dependent on the values of the design parameters n_1, n_2, n_3, L and J. These values have to be chosen by the user. At present, it is hard to give any general rules on how to chose the parameters. Instead we will try to deduce some rules of thumb.

First we take a look at n_1, n_2 and n_3 which are the number of parameters that are estimated in $y_k^{tra} = CA^k(x_0 - x_p)$, $y_k^p = CA^k x_p$ and $g_k = CA^{k-1}B$ respectively. We see that all three terms decays as A^k. Hence it is natural to assume that the number of terms that should be included in the estimation is about the same. Thus $n_1 = n_2 = n_3 = n$ could be a reasonable choice. For systems where the impulse response decreases quickly (A^k decreases quickly) compared to the number of samples N, n could be chosen "small" compared to N and vice versa for systems with long impulse responses. Based on extensive, but not exhaustive, simulations a choice of the parameters that gives good performance in many cases is $n_1 = n_2 = n_3 = 20$, $J = 1$ and $L = 10$. These values should be seen as a rule of thumb as the optimal parameter values are case dependent.

IV. Comparison with local polynomial method

The presented method is very much inspired by the local polynomial method presented and analyzed in a series of papers [6], [7], [8]. We highlight some of the main differences between the methods.

- The local polynomial method uses the fact that the transient term is smooth and approximates it with a Taylor series expansion. The proposed method instead explores the structure (3) of the transient term when making the approximation.
- The same "surplus" variables $\{y_k^{tra}\}_{k=0}^{n_1-1}$, $\{y_k^p\}_{k=0}^{n_2-1}$ and $\{g_k\}_{k=1}^{n_3}$ are used for all frequencies. This leads to a large least squares problem as shown before. The local polynomial method uses different parameters for all frequencies and can hence be solved locally. This leads to a small least square problem in each frequency. By using the same parameters for all frequencies it should be possible to decrease the noise induced error at a cost of increased computational complexity.

V. Efficient computations

The least squares problem previously showed has $(2L+1)N$ equations and $N + n_1 + n_2 + n_3$ unknowns. Even for modest N, n_1, n_2 and n_3 the direct solution becomes computational prohibitive. The problem has, however, a special structure that can be explored.

The regressor (6) can be written as

$$Y = \Phi_1 G + \Phi_2 \theta$$

where

$$Y_k = \begin{bmatrix} Y_{N_e}(\omega_{k,-L}) \\ \vdots \\ Y_{N_e}(\omega_{k,L}) \end{bmatrix} \in \mathbb{C}^{(2L+1)\times n_y}$$

$$Y = \begin{bmatrix} Y_0 \\ \vdots \\ Y_{N-1} \end{bmatrix} \in \mathbb{C}^{(2L+1)N\times n_y}$$

$$G = \begin{bmatrix} G_0(e^{j\omega_{0,0}}) \\ G_0(e^{j\omega_{1,0}}) \\ \vdots \\ G_0(e^{j\omega_{N-1,0}}) \end{bmatrix} \in \mathbb{C}^{n_u N\times n_y}$$

and Φ_1 is a regressor matrix, θ contains the "surplus-variables" with the corresponding regressor matrix Φ_2. The complete expressions for the matrices are given in Appendix II.

This is a *block angular least squares problem*. Numerous methods exploiting this special structure of the problem has been proposed, see for example [9]. Here we will use orthogonal transformations to solve the problem more efficiently. We start by making the orthogonal decomposition

$$\Phi_2 = \Phi_2^{\|} + \Phi_2^{\perp}$$

where $\Phi_2^{\perp}$ is orthogonal to both Φ_1 and $\Phi_2^{\|}$, *i.e.,*

$$\begin{aligned} \Phi_2^{\perp} &= (I - \Phi_1(\Phi_1^*\Phi_1)^{-1}\Phi_1^*)\Phi_2 \\ &= \mathrm{Diag}\{(I - U_k(U_k^*U_k)^{-1}U_k^*)\}\Phi_2 \\ &= \begin{bmatrix} (I - U_0(U_0^*U_0)^{-1}U_0^*)\Phi_2^0 \\ \vdots \\ (I - U_{N-1}(U_{N-1}^*U_{N-1})^{-1}U_{N-1}^*)\Phi_2^{N-1} \end{bmatrix}. \end{aligned} \qquad (7)$$

This gives

$$Y = \Phi_1 G + \Phi_2^{\|}\theta + \Phi_2^{\perp}\theta.$$

Multiplying both sides with $(\Phi_2^{\perp})^*$ yields

$$\begin{aligned} (\Phi_2^{\perp})^* Y &= \underbrace{(\Phi_2^{\perp})^*\Phi_1}_{=0} G + \underbrace{(\Phi_2^{\perp})^*\Phi_2^{\|}}_{=0}\theta + (\Phi_2^{\perp})^*\Phi_2^{\perp}\theta \\ &= (\Phi_2^{\perp})^*\Phi_2^{\perp}\theta. \end{aligned} \qquad (8)$$

Now a least squares estimate $\hat{\theta}$ of θ can be calculated from (8). Finally the least squares estimate $\hat{G}$ of G can be computed as

$$\begin{aligned} \hat{G} &= (\Phi_1^*\Phi_1)^{-1}\Phi_1^*\Delta Y = \mathrm{Diag}\{(U_k^*U_k)^{-1}U_k^*\}\Delta Y \\ &= \begin{bmatrix} (U_k^*U_k)^{-1}U_k^*\Delta Y_0 \\ \vdots \\ (U_{N-1}^*U_{N-1})^{-1}U_{N-1}^*\Delta Y_{N-1} \end{bmatrix} \end{aligned} \qquad (9)$$

where

$$\Delta Y = \begin{bmatrix} \Delta Y_0 \\ \vdots \\ \Delta Y_{N-1} \end{bmatrix} \triangleq Y - \Phi_2\hat{\theta}. \qquad (10)$$

To summarize, $\hat{G}$ defined in (9), can be calculated as

i) Compute the projections $\Phi_2^{\perp}$ from (7). This corresponds to calculating N matrix inversions $(U_k^*U_k)^{-1}$ of size $n_u \times n_u$.
ii) Solve the system of $n_y(n_1 + n_2 + n_u n_3)$ linear equations (8) in the $n_y(n_1 + n_2 + n_u n_3)$ unknowns θ.
ii) Compute ΔY defined in (10).
iv) Compute $\hat{G}$ from (9). This involves inverting N matrices of size $n_u \times n_u$.

VI. Simulation results

In this section the performance of the proposed method is evaluated through simulations. The results are compared with the current state of the art method to estimate the frequency response function, the local polynomial method.

A. Random systems

We start the comparison by apply the two methods to a large amount of randomized linear systems. The input-output relation for the simulated data is

$$y_t = G_0(q)u_t + H_0(q)e_t$$

where y_t is the output, u_t the input and e_t is white Gaussian noise. In each simulation both the system model, $G_0(q)$, and the noise model $H_0(q)$ are randomized with the MATLAB command *drss* and then normalized to get H_2 norm equal to one. Furthermore the order of the system, the order of the noise model, the length of the sampled data sequence, N, and the noise variance are independently randomized according to Table I.

TABLE I
SYSTEM PARAMETERS

Parameter	Distribution
Order of system model G	Uniform$[1, 20]$
Order of noise model H	Uniform$[1, 20]$
Samples, N	Uniform$[50, 600]$
Noise variance, Var e_t	Uniform$[0, 1.5]$
Initial state x_0	Normal$(0, I)$

The input is chosen as white Gaussian noise with unit variance and new realizations of the noise and the input are generated in each simulation.

The used settings are $n_1 = n_2 = n_3 = 20$, $J = 1$ and $L = 10$ for the proposed method and the recommended values [6] $R = 2$ and $n = 3$ for the local polynomial method. A total of 4000 simulations are performed and the ratio between the mean square errors (MSE)

$$MSE = \frac{1}{N}\sum_{k=0}^{N-1}\left|G_0(e^{j\frac{2\pi k}{N}}) - \hat{G}(k)\right|^2$$

of the proposed method and the local polynomial method is calculated after each simulation. A histogram of the ratios is shown in Fig. 1. Values above 1 means that the local polynomial method has a lower MSE and a value below 1 means that the proposed method has a lower MSE. It is clearly seen that the proposed method in general gives lower MSE with an average improvement of about a factor 9. In 98% of the simulations the proposed method performed better.

The average time per iteration was 4.8 s for the proposed method and 0.06 s for the local polynomial method. Hence the gain in performance comes with a large increase of computational complexity.

B. Resonant system

Here we study a single resonant system. The performance will be compared to the local polynomial method and to the Blackman-Tukey spectral analysis method [3].

Consider the continuous time system

$$G_0(s) = \frac{\omega_1^2}{s^2 + 2\xi\omega_1 s + \omega_1^2} + \frac{\omega_2^2}{s^2 + 2\xi\omega_2 s + \omega_2^2}$$

where $\omega_1 = 5$, $\omega_2 = 3\omega_1$ and $\xi = 0.1$. The system is sampled with sampling period $T_s = 0.1$.

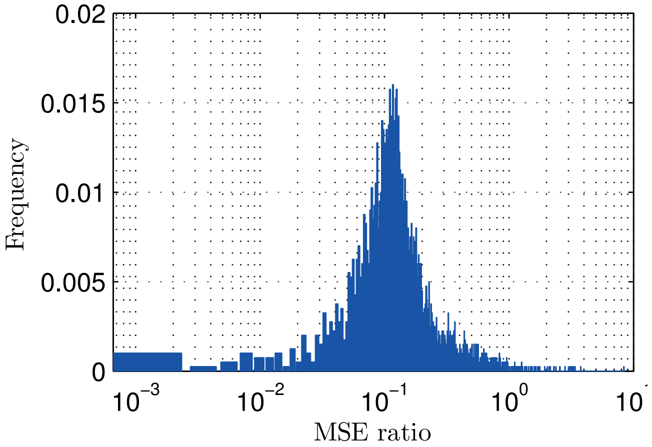

Fig. 1. Histogram of the ratio between the MSE of the proposed method and the local polynomial method for the 4000 simulations.

Again the input is chosen as white Gaussian noise with unit variance and the output of the system is disturbed with white Gaussian noise with variance λ_e. The estimation error will be assessed via Monte-Carlo simulations where new realizations of both the noise and the input are generated.

The used settings are the recommended $n_1 = n_2 = n_3 = 20$, $L = 10$ and $J = 1$ for the proposed method and $R = 2$ and $n = 3$ for the local polynomial method. $N = 100$ samples are collected of the output. A Hann window is used in the Blackman-Tukey method. A good window width for this case is found to be 45 lags.

Fig. 2 shows $\left|G_0(e^{j\omega})\right|$ together with the mean of the estimation errors over 500 Monte Carlo simulations in the noise free case ($\lambda_e = 0$) for both the new method and the local polynomial method. The used parameters are not optimal for this problem. By means of a search in the parameter space the values $n_1 = n_2 = n_3 = 36$, $J = 1$ and $L = 30$ are found to give good performance for this system and noise setting. The resulting mean over the estimation error, for this setting, is also shown in Fig. 2.

The MSE is 0.31 for the new method, 0.57 for the local polynomial method and 0.66 for the Blackman-Tukey method. The "best" parameters for the proposed method yield a MSE of 0.08.

Fig. 3 shows the result under the same conditions as before, save for that the noise variance is $\lambda_e = 0.3$. This gives a MSE of 0.44 for the new method, 1.09 for the local polynomial method and 0.77 for the Blackman-Tukey method. The parameters $n_1 = 31$, $n_2 = n_3 = 25$, $J = 1$ and $L = 34$ yield a MSE of 0.34. For this setup the average time per iteration was 0.5 s for the proposed method and $8 \cdot 10^{-6}$ s for the local polynomial method, respectively.

The experiments are repeated, but this time with a uniformly distributed white input signal with unit variance. No significant difference in the results with this input signal is observed.

VII. CONCLUSIONS

We have introduced a new method of identifying the frequency response function of a linear multiple input multiple

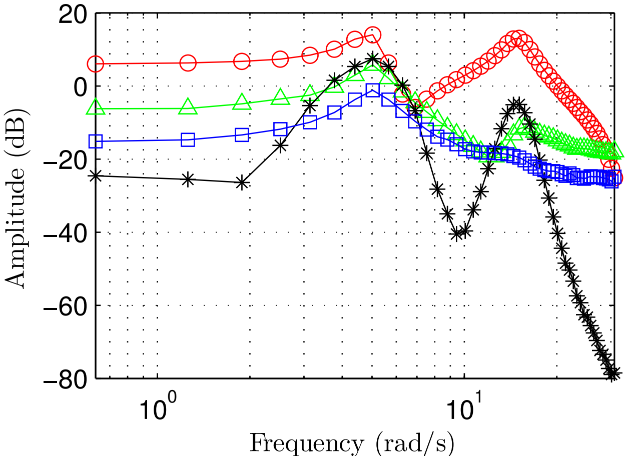

Fig. 2. Mean of the estimation error for the noise free case for the proposed method with recommended parameters (— △ —), for the proposed method with "best" parameters (—□—) and the local polynomial method (— ∗ —). True frequency response $G_0(e^{j\omega})$ is shown as (— ○ —).

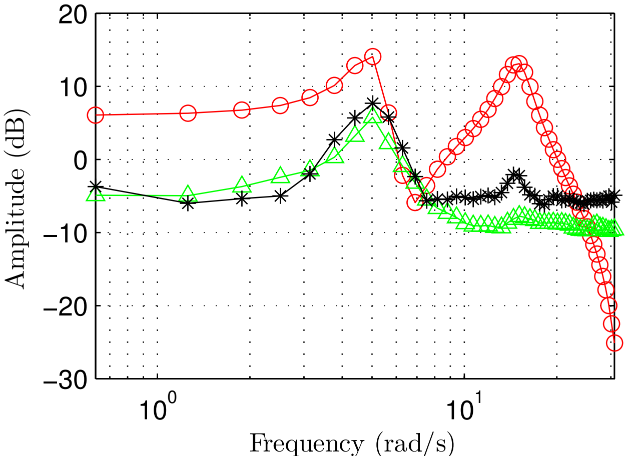

Fig. 3. Mean of the estimation error with noise for the proposed method (— △ —) and the local polynomial method (—∗—). True frequency response $G_0(e^{j\omega})$ is shown as (— ○ —).

output system. The method is closely related to the local polynomial method [6]. The difference is that instead of a polynomial approximation the structure of the problem is exploited. Oversampling or zero padding is also used, it is however, not fully understood why this gives better estimates.

The performance of the method was evaluated through numerical simulations. It shows promising performance compared to current state of the art methods for estimating the frequency response function. However, a thorough theoretical analysis of the method has to be done to verify the performance and to gain more insight into the method. From this theoretical analysis it might be possible to get a better understanding on how the parameter values should be chosen.

REFERENCES

[1] L. Ljung, *System identification : theory for the user*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999.

[2] R. Pintelon and J. Schoukens, *System identification : a frequency domain approach*. New York: IEEE Press, 2001.

[3] P. Stoica and R. Moses, *Spectral analysis of signals*. Upper Saddle River, N.J.: Pearson Prentice Hall, 2005.

[4] R. Pintelon, J. Schoukens, and G. Vandersteen, “Frequency domain system identification using arbitrary signals,” *Automatic Control, IEEE Transactions on*, vol. 42, no. 12, pp. 1717 –1720, Dec. 1997.

[5] T. McKelvey, “Frequency domain identification,” in *Preprints of the 12th IFAC Symposium on System Identification, Santa Barbara, USA.*, 2000.

[6] J. Schoukens, G. Vandersteen, K. Barbé, and R. Pintelon, “Nonparametric preprocessing in system identification: a powerful tool,” *European Journal of Control*, vol. 3-4, pp. 260–274, 2009.

[7] R. Pintelon, J. Schoukens, G. Vandersteen, and K. Barbé, “Estimation of nonparametric noise and FRF models for multivariable systems–Part I: Theory,” *Mechanical Systems and Signal Processing*, vol. 24, no. 3, pp. 573 – 595, 2010.

[8] ——, “Estimation of nonparametric noise and FRF models for multivariable systems–Part II: Extensions, applications,” *Mechanical Systems and Signal Processing*, vol. 24, no. 3, pp. 596 – 616, 2010.

[9] M. Cox, *Reliable numerical computation*. Oxford Univ. Press, 1990, ch. 13 The leasts-squares solution of linear equations with block-angular observation matrix, pp. 227–240.

APPENDIX I
PROOF OF GENERALIZED TRANSIENT TERM

Proof: If the sequence of N samples of the output y_t is zero padded with $2JN$ zeros then the zero padded signal, y_t^z can be expressed as

$$y_t^z = \bar{y}_t - y_{t-N}^p \tag{11}$$

where $\bar{y}_t$ is the output of the system when the input u_t, zero padded with $2J$ zeros, is used and y_{t-N}^p is defined as

$$y_t^p = \begin{cases} CA^t x_N & t \geq 0 \\ 0 & t < 0. \end{cases} \tag{12}$$

The signal y_t^p is hence the transient that stems from the nonzero state of the system at time N. See Fig. 4 for a sketch of the signals.

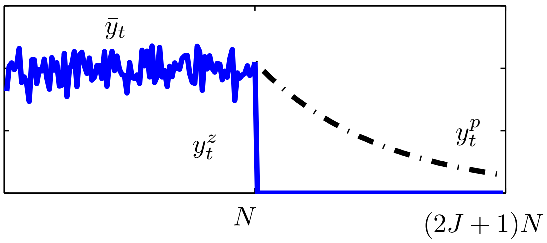

Fig. 4. Relationship between the signals for the zero padded output.

Computing the $N_e = (2J+1)N$-point DFT of the zero-padded output (11) yields

$$Y_{N_e}(\omega) = G_0(e^{j\omega})U_{N_e}(\omega) + Y_{N_e}^{tra}(\omega) - Y_{N_e}^p(\omega).$$

The term

$$\begin{aligned} Y_{N_e}^{tra}(\omega) =& \frac{1}{\sqrt{N}} e^{j\omega} C(e^{j\omega}I - A)^{-1}(I - A^{N_e})(x_0 - x_p^e), \\ x_p^e =& (I - A^{N_e})^{-1} A^{N_e - N}(I - A^N)x_p \end{aligned}$$

is just the transient term from [5] for the zero padded signal $\bar{y}_t$.

The third term is the DFT of y^p_{t-N}, (12), *i.e.*,

$$
\begin{aligned}
Y^p_{N_e}(\omega) &= \frac{1}{\sqrt{N}}\sum_{t=0}^{N_e-1} y^p_{t-N}e^{-j\omega t} = \frac{1}{\sqrt{N}}\sum_{t=N}^{N_e-1} CA^{t-N}e^{-j\omega t}x_N \\
&= \{s=t-N\} = \frac{1}{\sqrt{N}}C\sum_{s=0}^{N_e-N-1} A^s e^{-j\omega s}e^{-j\omega N}x_N \\
&= \frac{1}{\sqrt{N}}e^{j\omega}C(e^{j\omega}I-A)^{-1}(e^{-j\omega N}I-A^{N_e-N})x_N.
\end{aligned}
$$

The state of the system at time N can be expressed as

$$x_N = A^N x_0 + (I-A^N)x_p.$$

Combining the above equations give

$$
\begin{aligned}
&Y_{N_e}(\omega) = G_0(e^{j\omega})U_{N_e}(\omega) + Y^{tra}_{N_e}(\omega) - Y^p_{N_e}(\omega) \\
&= G_0(e^{j\omega})U_{N_e}(\omega) \\
&+ \frac{1}{\sqrt{N}}e^{j\omega}C(e^{j\omega}I-A)^{-1}(I-A^{N_e})(x_0-x^e_p) \\
&- \frac{1}{\sqrt{N}}e^{j\omega}C(e^{j\omega}I-A)^{-1}(e^{-i\omega N}I-A^{N_e-N})x_N \\
&= G_0(e^{j\omega})U_{N_e}(\omega)\frac{1}{\sqrt{N}}e^{j\omega}C(e^{j\omega}I-A)^{-1} \\
&\times \Big((I-A^{N_e})x_0 - A^{N_e-N}(I-A^N)x_p \\
&- (e^{-j\omega N}I-A^{N_e-N})(A^N x_0 + (I-A^N)x_p)\Big) \\
&= G_0(e^{j\omega})U_{N_e}(\omega) \\
&+ \frac{1}{\sqrt{N}}e^{j\omega}C(e^{j\omega}I-A)^{-1}(I-e^{-j\omega N}A^N)(x_0-x_p) + \\
&\frac{1-e^{-j\omega N}}{\sqrt{N}}e^{j\omega}C(e^{j\omega}I-A)^{-1}x_p.
\end{aligned}
\tag{13}
$$

The second term of (13) can be rewritten as

$$
\begin{aligned}
&\frac{1}{\sqrt{N}}e^{j\omega}C(e^{j\omega}I-A)^{-1}(I-e^{-j\omega N}A^N)(x_0-x_p) \\
&= \frac{1}{\sqrt{N}}C(I-Ae^{-j\omega})^{-1}(I-e^{-j\omega N}A^N)(x_0-x_p) \\
&= \left\{(I-T)^{-1}(I-T^N) = \sum_{k=0}^{N-1}T^k\right\} \\
&= \frac{1}{\sqrt{N}}C\sum_{k=0}^{N-1}e^{-j\omega k}A^k(x_0-x_p) \\
&= \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}CA^k(x_0-x_p)e^{-j\omega k} \\
&= \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}y^{tra}_k e^{-j\omega k} \triangleq Y^{tra}_N(\omega).
\end{aligned}
$$

The third term of (13) can be rewritten as

$$
\begin{aligned}
&\frac{1-e^{-j\omega N}}{\sqrt{N}}e^{j\omega}C(e^{j\omega}I-A)^{-1}x_p \\
&= \frac{1-e^{-j\omega N}}{\sqrt{N}}\sum_{k=0}^{\infty}CA^k x_p e^{-j\omega k} \\
&= \frac{1-e^{-j\omega N}}{\sqrt{N}}\sum_{k=0}^{\infty}y^p_k e^{-j\omega k} \\
&\triangleq \frac{1-e^{-j\omega N}}{\sqrt{N}}Y_p(\omega).
\end{aligned}
$$

We see that (13) is independent of the oversampling factor J. The factor J can hence be chosen arbitrary large and the frequency resolution could be made arbitrary small. Hence the expression is valid for all ω. ∎

APPENDIX II
MATRICES FOR THE REGRESSOR PROBLEM

$$
U_k = \begin{bmatrix} U^T_{N_e}(\omega_{k,-L}) \\ \vdots \\ U^T_{N_e}(\omega_{k,L}) \end{bmatrix} \in \mathbb{C}^{(2L+1)\times n_u}
$$

$$
\Phi_1 = \begin{bmatrix} U_0 & 0 & \cdots & 0 \\ 0 & U_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & U_{N-1} \end{bmatrix} \in \mathbb{C}^{(2L+1)N\times n_u N}
$$

$$
\theta = \begin{bmatrix} (y^{tra}_0)^T \\ \vdots \\ (y^{tra}_{n_1-1})^T \\ (y^p_0)^T \\ \vdots \\ (y^p_{n_2-1})^T \\ g^T_1 \\ \vdots \\ g^T_{n_3} \end{bmatrix} \in \mathbb{C}^{(n_1+n_2+n_u n_3)\times n_y}
$$

$$
\Psi_{k,1} = \begin{bmatrix} \frac{1}{\sqrt{N}} & \frac{e^{-j\omega_{k,-L}}}{\sqrt{N}} & \cdots & \frac{e^{-j\omega_{k,-L}(n_1-1)}}{\sqrt{N}} \\ \frac{1}{\sqrt{N}} & \frac{e^{-j\omega_{k,-L+1}}}{\sqrt{N}} & \cdots & \frac{e^{-j\omega_{k,-L+1}(n_1-1)}}{\sqrt{N}} \\ \vdots & \vdots & & \vdots \\ \frac{1}{\sqrt{N}} & \frac{e^{-j\omega_{k,L}}}{\sqrt{N}} & \cdots & \frac{e^{-j\omega_{k,L}(n_1-1)}}{\sqrt{N}} \end{bmatrix}
\in \mathbb{C}^{(2L+1)\times n_1}
$$

$$
\Psi_{k,2} = \begin{bmatrix} \frac{1-e^{-j\omega_{k,-L}N}}{\sqrt{N}} & \cdots & \frac{1-e^{-j\omega_{k,-L}N}}{\sqrt{N}}e^{-j(n_2-1)\omega_{k,-L}} \\ \vdots & & \vdots \\ \frac{1-e^{-j\omega_{k,L}N}}{\sqrt{N}} & \cdots & \frac{1-e^{-j\omega_{k,-L}N}}{\sqrt{N}}e^{-j(n_2-1)\omega_{k,L}} \end{bmatrix}
\in \mathbb{C}^{(2L+1)\times n_2}
$$

$$
\Psi_{k,3} = \begin{bmatrix} \varphi_1(\omega_{k,-L},\omega_{k,0})U^T_{N_e}(\omega_{k,-L}) & \cdots & \varphi_{n_3}(\omega_{k,-L},\omega_{k,0})U^T_{N_e}(\omega_{k,-L}) \\ \vdots & & \vdots \\ \varphi_1(\omega_{k,L},\omega_{k,0})U^T_{N_e}(\omega_{k,L}) & \cdots & \varphi_{n_3}(\omega_{k,L},\omega_{k,0})U^T_{N_e}(\omega_{k,L}) \end{bmatrix} \in \mathbb{C}^{(2L+1)\times n_u n_3}
$$

$$
\Phi_2 = \begin{bmatrix} \Psi_{0,1} & \Psi_{0,2} & \Psi_{0,3} \\ \vdots & \vdots & \vdots \\ \Psi_{N-1,1} & \Psi_{N-1,2} & \Psi_{N-1,3} \end{bmatrix}
\in \mathbb{C}^{(2L+1)N\times(n_1+n_2+n_u n_3)}
$$