2015 American Control Conference
Palmer House Hilton
July 1-3, 2015. Chicago, IL, USA

The Discrete Time Biquad State Space Structure: Low Latency with High Numerical Fidelity

Daniel Y. Abramovitch*

***Abstract*— State space models of highly flexible systems can present severe numerical issues. The models derived from physical principles often lack structure. Canonical form models, are compact, but obscure any physical structure and can have coefficients that are highly sensitive to model parameters. What is needed is a form that has the compact representation of the canonical forms, the physicality of the forms derived from physical equations, and maintain numerical accuracy and physical intuition, even after discretization. This paper presents a new state space form, the Biquad State Space (BSS), based on the multinotch structure [1], [2]. We will will show that the BSS captures the endearing characteristics of the multinotch while providing the flexibility of model based control. This paper will present the basic structure in discrete time form which most closely matches the multinotch. Forms not specific to minimum latency control, including a continous time version, will be discussed in [3].**

I. Introduction

One of the aspects of digital control that gets brief mention in control textbooks and research papers is the implementation of low latency control. It is well understood that latency, including computational latency, erodes phase margin by adding negative phase. Some textbooks mention precalculating operations which do not depend upon the current input in the preceding sample interval [5], [6]. The savings in latency are illustrated in Figure 1.

In the Single-Input, Single-Output (SISO) case this is tedious, but relatively straightforward if the controller can be cast into the form of a high order polynomial filter. This is shown in Figure 2, and represented as transfer functions in the unit delay operator, z^{-1}:

$$\frac{Y(z^{-1})}{U(z^{-1})} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n}}. \quad (1)$$

This gets implemented in a filter as [7]:

$$\begin{aligned} y_k &= -a_1 y_{k-1} - a_2 y_{k-2} - \ldots - a_n y_{k-n} \\ &\quad + b_0 u_k + b_1 u_{k-1} + \ldots + b_n u_{k-n}. \end{aligned} \quad (2)$$

Looking at (2), we see that y_k depends mostly on previous inputs and outputs. The only current value needed is u_k and this is only multiplied by b_0. So we can break this up into [5]:

$$y_k = b_0 u_k + \text{prec}_k, \text{ where} \quad (3)$$

$$\begin{aligned} \text{prec}_k &= -a_1 y_{k-1} - \ldots - a_n y_{k-n} \\ &\quad + b_1 u_{k-1} + \ldots + b_n u_{k-n}. \end{aligned} \quad (4)$$

*Daniel Y. Abramovitch is a system architect in the Mass Spectrometry Division, Agilent Technologies, 5301 Stevens Creek Blvd., M/S: 3U-DG, Santa Clara, CA 95051 USA, danny@agilent.com

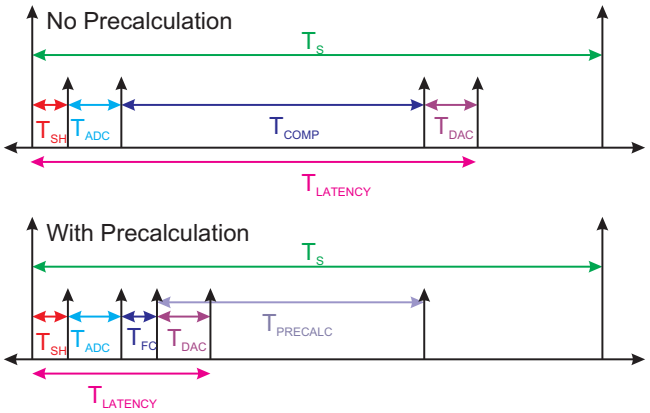

Fig. 1. Input and output timing in a digital control system. The top drawing is without precalculation; the bottom drawing is with. Note that precalculation can be started as soon as the output has been sent to the DAC and therefore is in parallel with the DAC conversion time. The computation time, T_{COMP}, of the top diagram is now split into $T_{PRECALC} + T_{FC}$ where $T_{PRECALC}$ is the computation time needed for the precalculation and T_{FC} is the time needed for the final calculation after the input sample. Modulo some small programming overhead, the split time should equal the total computation time. Here T_{SH}, T_{ADC}, and T_{DAC} represent the sample and hold, ADC conversion, and DAC conversion times, respectively.

We can see that prec_k depends only on previous values of y_k and u_k. This means that prec_k can be computed for step k immediately after the filter has produced the output for time index, $k-1$ [6]. When the sample at time step k, u_k, comes into the filter, it need merely be multiplied by b_0 and added to prec_k to produce the filter output. Thus, the delay between the input of u_k and the output of y_k is small and independent of the filter length. Small latency improves performance, but fixed latency implies predictable behavior, which may be more critical in debugging real time system.

In [1], [8], the multinotch was introduced as a discrete time filter whose structure allowed for fixed and low latency between the most recent signal input and the filter output, while having the excellent numerical properties inherent in biquad structures. In [2] we demonstrated a filter coefficient adjustment, the Δ coefficients, which allowed high numerical fidelity even when the sample frequency was several orders of magnitude higher than that of the dynamic feature being filtered. Both of these papers implement the filter in a transfer function form.

This paper will demonstrate how to adapt the multinotch for state space structures [9]. We will see that the same basic principles can be used to improve the computational latency and numerical fidelity of current mode observers, thereby allowing state feedback with fixed and low latency.

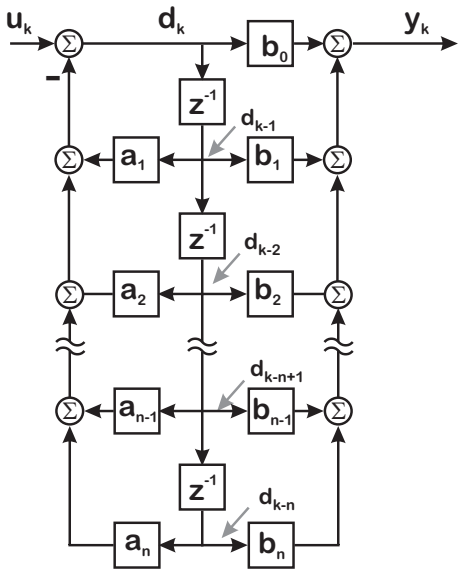

Fig. 2. An nth order polynomial filter in Direct Form II configuration [4].

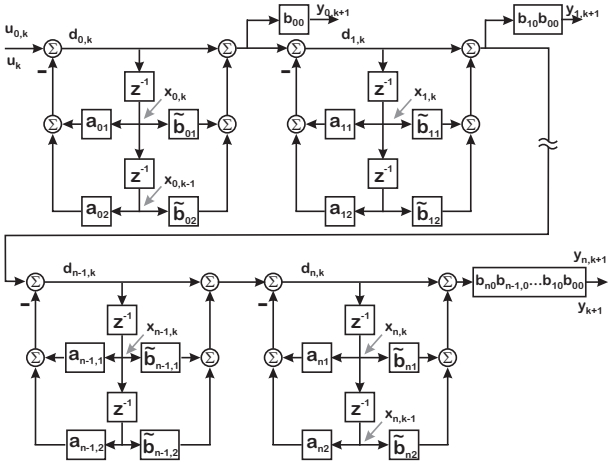

Fig. 3. The updated discrete biquad cascade, with factored out $b_{i,0}$ terms and scaling the output of each block.

Furthermore, state space models of highly flexible systems can present severe numerical issues. The models derived from physical principles often lack structure. Canonical form models [10], are compact, but obscure any physical structure and can have coefficients that are highly sensitive to model parameters. What is needed is a form that has the compact representation of the canonical forms, the physicality of the forms derived from physical equations, and maintain numerical accuracy and physical intuition, even after discretization.

While the multinotch was applied primarily to shaping loop dynamics with high Q resonances and anti-resonances, a good state space model also needs to be able to account for low frequency and rigid body dynamics. This will be demonstrated, using the classic double integrator as an example, in Section V.

The numerical benefits of this form exist even when low latency is not a consideration, so we will show forms of the structure applicable in offline modeling and simulation in [3]. Finally, we will show a modeling example from experimental data of a mechatronics system where the Biquad State Space (BSS) form holds numerical accuracy far beyond conventional methods, as will become obvious in the examples of Section VII.

While the structure is quite regular and works for large or small numbers of biquads, the regular pattern becomes obvious in the three biquad case. Thus, most of the structural equations will be three biquad ones. The format considerations of this will mean that many of these matrix equations are in two column figures, but seeing the matrices in this way makes the structural properties fairly obvious. This will result in some of the larger equations being pushed into two column figures.

II. The Biquad Decomposition of Digital Filters

In [1], we discussed how a higher order Single-Input, Single-Output (SISO) digital filter, such as that in Equation 1, can be factored into a chain of second order filters known as biquads. This has been well established for a long time. However, until [1], using biquads [7] in digital feedback control meant that precalculation [5], [6] to reduce computational latency was limited to only the first biquad block since all downstream blocks needed the final output of the first block to do any computations. The multinotch, by factoring out the direct feedthrough coefficients, and only multiplying them in at the output, removed that constraint, allowing the numerical advantages of biquad decomposition to be coupled with the low latency advantages of precalculation.

There is no need to repeat the equations of [1] here, but looking at the structure the multinotch in Figure 3 there are a few things to note before generating our first state space form:

- The delay terms in the biquads are equivalent to states in a state space structure, but they are offset in time. Looking at Figure 3, $d_{i,k} = x_{i,k+1}$. That is, the digital filter approach defines delays on the input of time shifts (z^{-1}) while standard state space notation defines states on the outputs of time shifts.
- While $\tilde{y}_{i,k+1}$ depends on $x_{i,k+1}$, it can be recalculated as a weighted sum of prior delays and the current input. That is, we can calculate $\tilde{y}_{i,k+1}$ in parallel to $x_{i,k+1}$.

III. A Biquad State Space Form

So far, we have not done anything not already in [1]. However, we can look at each of these biquad sections as a state space realization. In this case:

$$\begin{bmatrix} x_{i,k+1} \\ x_{i,k} \end{bmatrix} \begin{bmatrix} -a_{i1} & -a_{i2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{i,k} \\ x_{i,k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_{i,k} \quad (5)$$

while the state output equation is given by:

$$\begin{bmatrix} \tilde{y}_{i,k+1} \end{bmatrix} = \begin{bmatrix} \tilde{b}_{i1} - a_{i1} & \tilde{b}_{i2} - a_{i2} \end{bmatrix} \begin{bmatrix} x_{i,k} \\ x_{i,k-1} \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} u_{i,k} \tag{6}$$

Finally, the properly scaled output is generated via:

$$\begin{bmatrix} y_{i,k+1} \end{bmatrix} = \begin{bmatrix} b_{i0} \end{bmatrix} \begin{bmatrix} \tilde{y}_{i,k+1} \end{bmatrix}. \tag{7}$$

The indexing of $\tilde{y}_{i,k+1}$ and $y_{i,k+1}$ are a bit odd because since we have direct feedthrough in our structure, $\tilde{y}_{i,k+1}$ depends on $x_{i,k+1}$ as well as $x_{i,k}$, $x_{i,k-1}$, and $u_{i,k}$. Thus, it's cleaner in what follows to call the biquad outputs, $\tilde{y}_{i,k+1}$ and $y_{i,k+1}$, respectively. We chain these together by noting that:

$$\begin{array}{lcll} u_{i+1,k} & = & \tilde{y}_{i,k+1}, & \text{for } 0 \leq i < n, \\ u_{0,k} & = & u_k, & \text{and} \\ \tilde{y}_{n,k+1} & = & \tilde{y}_{k+1}. & \end{array} \tag{8}$$

If one is willing to go through the algebraic pain and suffering of applying Equation 8 to each biquad structure a very regular state space structure results. For a 3-biquad model, we get the state equation of 9. The unscaled output is in Equation 10, both displayed in Figure 4 due to their size. Finally, the properly scaled outputs are generated via:

$$\begin{bmatrix} y_{2,k+1} \\ y_{1,k+1} \\ y_{0,k+1} \end{bmatrix} = \begin{bmatrix} b_{20}b_{10}b_{00} & 0 & 0 \\ 0 & b_{10}b_{00} & 0 \\ 0 & 0 & b_{00} \end{bmatrix} \begin{bmatrix} \tilde{y}_{2,k+1} \\ \tilde{y}_{1,k+1} \\ \tilde{y}_{0,k+1} \end{bmatrix}. \tag{11}$$

One key of this form is that the generation of the state update (output vector) involves:

- Multiplication of the prior state vector by the state transition matrix (state output matrix) – none of which involves the current input. This can therefore be done in a precalculation step.
- Addition of the unscaled current input to each product row of the above multiplication. This can be parallelized so that the latency once the current input is available is that of a single addition.

Looking at this critically, the state transition and output matrices are always multiplied by the old state, and therefore could be precalculated in any form. If there is no direct feedthrough from the input to the output, such a model can be used without incurring much delay. However, the BSS is structured so that direct feedthrough from the input to the output needs one addition and one multiplication per output. This is a big benefit for using state space in real time control. The fact that the BSS also provides excellent numerical properties as will be seen in the example of Section VII.

The generation of the final, scaled output takes a single multiplication per output. Therefore, updating the state and output using the BSS using precalculation has a computational latency of two operations: one addition and one multiplication.

IV. The Matrices, Reloaded

Generating coefficients from continuous time biquad parameters is discussed in some detail in [1] and [2]. Suffice it to say that continuous time, physical parameters can be mapped into the discrete time biquads which form the basis of our state matrices.

The state transition matrix in Equation 9 has a very regular, block upper triangular form. On the block diagonals are 2×2 blocks with the biquad denominator parameters (from which we can extract the model poles). Below the diagonal blocks are empty, while above the diagonal blocks is a repeated set of 2×2 blocks with $0s$ on the lower rows and

$$\begin{bmatrix} \tilde{b}_{i,1} - a_{i,1} & \tilde{b}_{i,2} - a_{i,2} \end{bmatrix} \tag{12}$$

on the top row. The top rows of these blocks represent the feedthrough of the biquad states to the other states. Likewise in the output matrix of Equation 10, these same subsections in (12) represent the feedthrough of the biquad states to the outputs. Note that in both of these matrix equations, the input is passed unscaled to the states and unscaled outputs. The gain scaling is applied in (11).

Note that while these matrices are denser than a typical canonical form, many of the needed multiplications and additions are repeated, so that proper coding of the state and unscaled output updates makes this form no more computationally intense than a canonical form.

The above the block diagonal blocks are governed by the terms in (12), and these terms are determined by how the overall system model is partitioned into biquads. One way to minimize these terms is to arrange the pole-zero groupings so that each biquad consists of poles and zeros that are closest to each other.

V. Adding Rigid Body Dynamics: Double Integrator

For modeling any real mechatronic system, there will have to be some sort of rigid body or low frequency resonance model. In this section, we will show how to add a double integrator to this biquad structure. The simplest way, of course, would be if the double integrator could just be modeled as a biquad. Defining our double integrator as $D(s) = K/s^2$ and applying the Trapezoidal rule yields

$$D_T(z^{-1}) = K\left(\frac{T}{2}\right)^2 \left(\frac{1+z^{-1}}{1-z^{-1}}\right)^2. \tag{13}$$

Neglecting the gain, $K\left(\frac{T}{2}\right)^2$, we define

$$\tilde{D}_T(z^{-1}) = \left(\frac{1+z^{-1}}{1-z^{-1}}\right)^2 = \frac{1+2z^{-1}+z^{-2}}{1-2z^{-1}+z^{-2}} \tag{14}$$

from which we can extract the time domain equations

$$d_k - 2d_{k-1} + d_{k-2} = u_k. \tag{15}$$

Remembering that in the traditional state-space notation $x_{k+1} = d_k$ we get

$$x_{k+1} = 2x_k - x_{k-1} + u_k \tag{16}$$

and

$$\tilde{y}_{k+1} = x_{k+1} + 2x_k + x_{k-1}. \tag{17}$$

$$\begin{bmatrix} x_{2,k+1} \\ x_{2,k} \\ x_{1,k+1} \\ x_{1,k} \\ x_{0,k+1} \\ x_{0,k} \end{bmatrix} = \begin{bmatrix} -a_{21} & -a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_{11} & -a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{01} & -a_{02} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{2,k} \\ x_{2,k-1} \\ x_{1,k} \\ x_{1,k-1} \\ x_{0,k} \\ x_{0,k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_k \quad (9)$$

$$\begin{bmatrix} \tilde{y}_{2,k+1} \\ \tilde{y}_{1,k+1} \\ \tilde{y}_{0,k+1} \end{bmatrix} = \begin{bmatrix} \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 0 & 0 & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \end{bmatrix} \begin{bmatrix} x_{2,k} \\ x_{2,k-1} \\ x_{1,k} \\ x_{1,k-1} \\ x_{0,k} \\ x_{0,k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u_k \quad (10)$$

Fig. 4. State equations for discrete time biquad state space with scalar output scaling.

Note that $\tilde{y}_{k+1}$ depends upon x_{k+1} which we have defined in terms of previous values of x_k and the current input, u_k, so we can make the substitutions to get

$$\begin{aligned} \tilde{y}_{k+1} &= x_{k+1} + 2x_k + x_{k-1} \\ &= 2x_k - x_{k-1} + u_k + 2x_k + x_{k-1} \\ &= 4x_k + u_k. \end{aligned} \quad (18)$$

We put this in state space form as:

$$\begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k. \quad (19)$$

The output is defined as:

$$[\tilde{y}_{k+1}] = \begin{bmatrix} 4 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} u_k. \quad (20)$$

Finally,

$$[y_{k+1}] = \left[KT^2/4\right] [\tilde{y}_{k+1}]. \quad (21)$$

This is great news. What we have seen is that we can treat a double integrator as a digital biquad, and so we can drop it right into our structure, simply by choosing

$$\begin{aligned} &a_1 = -2, \quad a_2 = 1, \\ &\tilde{b}_1 = 2, \quad \tilde{b}_1 = 1, \quad \text{and } b_0 = \tfrac{KT^2}{4}. \end{aligned} \quad (22)$$

VI. Current Estimator and State Feedback

In a prediction estimator, the measurement error is formed using the previous measurement and a state output generated entirely from quantities available before the current measured output. This means that the BSS does not have a significant latency advantage in a predictor form observer, simply because the latter already has a full sample of latency. A current estimator, on the other hand, depends on the current measurement. It is for this type of estimator where we can get some latency savings as shown in Figure 1.

In order to use our form in an observer, we need to generate time update and measurement update equations. For the 3-biquad case, the time update equation is given by Equation 23. For the SISO case, there will only be a single output, and so the output equations become what is shown in Equation 24.

Finally, the properly scaled time update output is generated via a single multiplication of concatenated feedthrough coefficients, in a similar manner to [1].

$$\bar{y}_k = \bar{y}_{2,k} = b_{20}b_{10}b_{00}\tilde{y}_{2,k}. \quad (25)$$

For the SISO measurement update, the equations are quite simple:

$$e_k = y_{meas,k} - \bar{y}_k, \text{ and} \quad (26)$$

$$\hat{x}_k = \bar{x}_k + L_c e_k. \quad (27)$$

Now, Equation 26 involves one subtraction. Equation 27 involves one multiply and addition for each state, but these are independent and so can be done in parallel. The latency then, for the state state update, is that of 2 multiplies, plus 3 add/subtract operations, independent of the size of the state. To use the state estimate in state feedback would require

$$u_{fb,k} = K_{fb}\hat{x}_k = K_{fb}\bar{x}_k + K_{fb}L_c e_k, \quad (28)$$

in which the $K_{fb}\bar{x}_k$ and the $K_{fb}L_c$ products can be recalculated. Thus for a SISO system, state feedback involves one more multiply and one addition.

VII. Examples

In order to demonstrate the numerical improvements arising from the biquad state space structure, an example is take from measurements of an Aerotech linear stage used in experiments for the Quintessential Phase project [11]. The Aerotech single axis stage as shown in Figure 6. The Aerotech 3300 stage controller includes a PID like feedback controller along with a feedforward portion. The sample rate for these is 8 kHz. In order to obtain a clean frequency response, the Eric Johnstone [11] turned off the feedforward compensator and then used the Aerotech controller's built in swept-sine functionality. A 1000 point swept-sine frequency response function (FRF) was taken on a logarithmic frequency axis from 10 Hz to 4 kHz. The Aerotech controller returned an open loop FRF, which was uploaded to Matlab. There a model of the Aerotech PID was constructed using Aerotech parameters. A FRF for this controller was synthesized on the same frequency axis as the stage open loop

$$\begin{bmatrix} \bar{x}_{2,k} \\ \bar{x}_{2,k-1} \\ \bar{x}_{1,k} \\ \bar{x}_{1,k-1} \\ \bar{x}_{0,k} \\ \bar{x}_{0,k-1} \end{bmatrix} = \begin{bmatrix} -a_{21} & -a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_{11} & -a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{01} & -a_{02} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{2,k-1} \\ \hat{x}_{2,k-2} \\ \hat{x}_{1,k-1} \\ \hat{x}_{1,k-2} \\ \hat{x}_{0,k-1} \\ \hat{x}_{0,k-2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_{k-1}. \quad (23)$$

$$\begin{bmatrix} \tilde{\bar{y}}_{2,k} \end{bmatrix} = \begin{bmatrix} \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \end{bmatrix} \begin{bmatrix} \hat{x}_{2,k-1} \\ \hat{x}_{2,k-2} \\ \hat{x}_{1,k-1} \\ \hat{x}_{1,k-2} \\ \hat{x}_{0,k-1} \\ \hat{x}_{0,k-2} \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} u_{k-1} \quad (24)$$

Fig. 5. Time update equations for discrete time biquad state space with scalar output scaling.

Fig. 6. Laboratory system: Aerotech air bearing linear stage, including linear grating for position measurement. The Aerotech system implements a PID controller and samples the data at 8 kHz. It has a built in swept-sine measurement. In the center of the image is a laser interferometer (IF) to provide an alternate measurement of the stage position. The stage itself is to the left of the IF.

20 Biquad Fit Parameters for Aerotech Stage				
Biquad #	$f_{N,n}$ **(Hz)**	Q_n	$f_{N,d}$ **(Hz)**	Q_d
1	2116.5	46.9420	1871.2	9.0188
2	1162.6	9.3768	1301.8	1.9112
3	619.9	4.8004	631.5	13.2948
4	1792.6	13.1546	1726.6	27.3882
5	702.0	0.3261	1374.2	0.1245
6	428.7	25.2154	449.4	7.3544
7	559.7	10.4940	549.4	18.6003
8	248.3	2.5601	241.9	3.2069
9	1891.1	31.8588	1874.3	21.0130
10	1484.5	14.0540	1506.3	11.2718
11	720.5	5.0254	718.5	7.0045
12	458.0	24.6218	459.3	19.0552
13	287.8	9.7413	286.8	9.8888
14	225.7	14.7047	225.4	14.0349
15	3590.2	7.4186	3203.0	10.2562
16	2159.3	60.0000	2143.5	21.8389
17	1947.3	11.7009	1954.3	9.3325
18	1982.2	9.2703	1982.1	9.2825
19	1936.4	9.2163	1936.5	9.2002
20	2128.2	60.0000	2121.7	84.6660

TABLE I
MODEL PARAMETERS FROM CURVE FIT OF AEROTECH FREQUENCY RESPONSE DATA.

response measurement, and this controller FRF was divided out of the open loop FRF to obtain a "plant" FRF. This plant FRF was fit to a stage model that consisted of a double integrator plus 20 analog biquads. The biquads are ranked in order of significance on the frequency response so that if one wants to simplify the model, one removes the latter biquads. The identified model parameters are in Table I.

In order to compare the biquad state space to more conventional methods, the fit parameters were then used to generate both transfer function models and state space models in Matlab. The linear system concatenation functions were used for both of these. From these high order models, Bode plots were generated to compare to the original measurement. Similarly, model terms were used to construct a biquad state space structure and again, a Bode plot was generated. Note that these plots are not made using fixed point math, but with all terms represented in Matlab's dual precision floating point format.

In Figure 7, we see that with up to 12 biquads and a rigid body, all the methods produce essentially equivalent Bode plots, that match the magnitude data exceptionally well. The phase features are matched, with the exception of the general rolloff that can be attributed to time delay not modelled in the rigid body or the biquads.

However, just the addition of two more biquads in Figure 8, we see that the two "conventional" methods deviate significantly from the measured frequency response. At 20 biquads plus the rigid body as shown in Figure 9, it is very clear that the conventional methods are so affected by numerical issues that they cannot come close to representing the measurement, either a low frequency or high frequency. In both of these cases, we see that the biquad state space continues to match the original measurement.

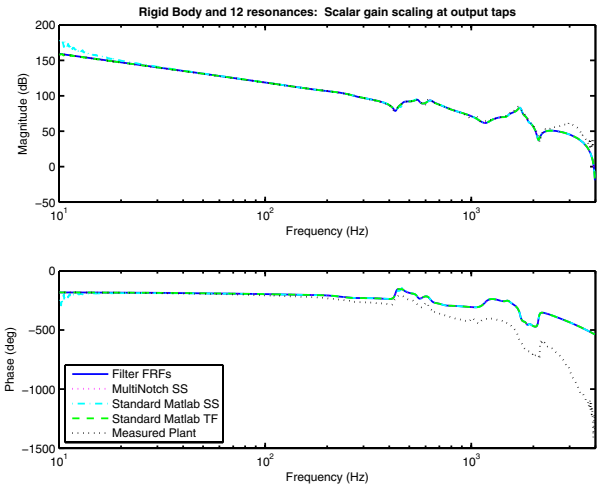

Fig. 7. Comparing state space forms to AeroTech stage frequency response. Modeling the system with first 12 biquads and a rigid body, there is no discernible difference in the plots. The measured plant exhibits a phase roll off at high frequency not fit by the biquads.

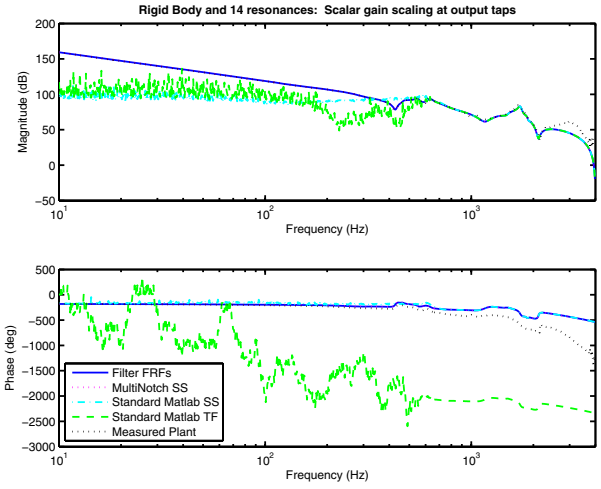

Fig. 8. Comparing state space forms to Aerotech stage frequency response. Modeling the system with first 14 biquads and a rigid body, we start seeing significant differences in the different methods of realizing the state space form. The conventional methods are clearly not matching the measured Aerotech frequency response, while the biquad state space method is.

VIII. Conclusions

The biquad state space (BSS) form adapts the multinotch filter [1] for state space use, preserving the latter's excellent numerical properties. One form of the BSS has the same minimum latency behavior that makes the multinotch so useful for real-time control. Like the multinotch, this "scalar output gain" version of the BSS has computational latency after the input sample that is independent of the number of states. When used as an observer for state feedback in a SISO system, the extra latency is due to n_{states} multiplications (which can be done in parallel) and a sum of these products. Depending upon the computational architecture and the number of products, the addition can also be accomplished in 1 to $n_{states} - 1$ operations, the latter if addends must be summed 2 at a time.

Even when doing off line modeling, the examples in Section VII demonstrate how the BSS preserves numerical fidelity in the state space model. It also preserves the physical intuition of the analog parameters in the digital state space matrices, which is extremely helpful in debugging physical systems. This will be discussed in [3].

(The author would like to thank Eric Johnstone of KeySight Technologies (formerly Agilent) for the inspiration for this project, as well as his skill using Maple to debug algebraic typos.)

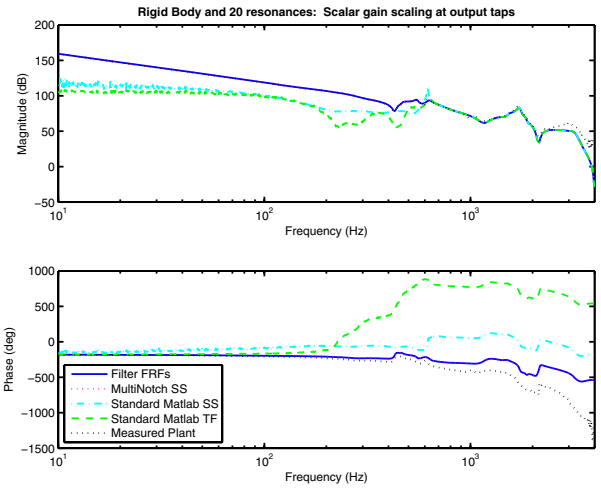

Fig. 9. Comparing state space forms to Aerotech stage frequency response. Modeling the system with first 20 biquads and a rigid body, there is a massive difference between the conventional methods and the biquad state space method.

References

[1] D. Y. Abramovitch, "The Multinotch, Part I: A low latency, high numerical fidelity filter for mechatronic control systems," in *Submitted to the 2015 American Control Conference*, (Chicago, IL), AACC, IEEE, July 2015.

[2] D. Y. Abramovitch, "The Multinotch, Part II: Extra precision via Δ coefficients," in *Submitted to the 2015 American Control Conference*, (Chicago, IL), AACC, IEEE, July 2015.

[3] D. Y. Abramovitch, "The continuous time biquad state space structure," in *Submitted to the 2015 American Control Conference*, (Chicago, IL), AACC, IEEE, July 2015.

[4] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, N. J.: Prentice Hall, 1975.

[5] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*. Menlo Park, California: Addison Wesley Longman, third ed., 1998.

[6] K. J. Åström and B. Wittenmark, *Computer Controlled Systems, Theory and Design*. Englewood Cliffs, N.J. 07632: Prentice Hall, second ed., 1990.

[7] Texas Instruments, *TMS320C4x User's Guide*, 1993.

[8] D. Y. Abramovitch and C. R. Moon, "Cascaded digital filters with reduced latency," International Application Published Under the Patent Cooperation Treaty WO 2012/118483, World Intellectual Property Organization, September 9 2012.

[9] D. Y. Abramovitch and E. Johnstone, "State space system simulator utilizing bi-quadratic blocks to simulate lightly damped resonances," International Application Published Under the Patent Cooperation Treaty WO 2013/130076, World Intellectual Property Organization, September 6 2013.

[10] T. Kailath, *Linear Systems*. Englewood Cliffs, N.J. 07632: Prentice-Hall, 1980.

[11] E. Johnstone and D. Y. Abramovitch, "Quintessential phase: A method of mitigating turbulence effects in interferometer measurements of precision motion," in *Proceedings of the 2013 American Control Conference*, (Washington, DC), AACC, IEEE, June 17–19 2013.