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

The Continuous Time Biquad State Space Structure

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. In [1] we discussed discrete state space structure that allowed for minimal latency estimation and control while preserving the numerical properties of biquad filters. In this paper, we discuss a continuous time version, suitable for modeling from physical principles.**

I. INTRODUCTION

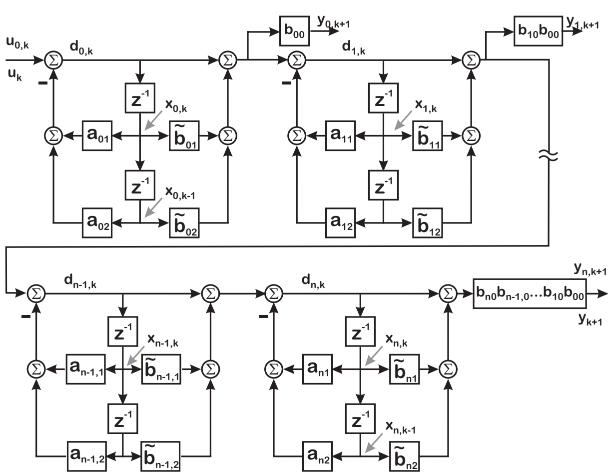

Fig. 1. The discrete time, Biquad State Space structure as described in [1].

State space models of highly flexible systems can present severe numerical issues. The models derived from physical principles often lack structure and have large parameter sets. On the other hand, canonical form models [2] reduce the number of parameters (and therefore computational operations) to a minimum set equivalent to those in a transfer function form. However, in doing so for anything more sophisticated than a second order model, most – if not all – physical intuition is lost. Furthermore, the compaction of these parameters into a canonical set often results in parameters that are highly sensative to small changes in the underlying physical parameters. Such models often fail when used with systems of higher order. Furthermore, even if the models are usable in continuous time, they can become even more sensitive and far less physical once the system is discretized. This is particularly true for mechatronic systems, which often are characterized by a "rigid body" model followed by multiple sharp resonances and anti-resonances.

All of this creates a situation where state space approaches are used only by experts in the field, while more basic, physically intuitive approaches continue to dominate in industrial applications. These intuitive methods may work fine when the system is low order, but they break down as the system complexity rises. What is needed is a form that can capture higher order dynamics in a way that maintains physical intuition and preserves numerical accuracy through the discretization process.

This paper presents a new state space form, the analog Biquad State Space, based on the multinotch structure [3], [4]. The Biquad State Space (BSS) [1] has several desirable characteristics:

- It uses a structure based on a serialized biquad filters which can be physically matched to resonance/anti-resonance pairs observed in measurements.
- The number of parameters is comparable to that of a canonical form, although many appear multiple times in the matrix structures.
- The basic structure remains the almost the same through discretization.
- The underlying biquad structure leads to a state space structure that is numerically very stable, even through discretization. The Δ-parameters from the multinotch [4] can be used to improve the numerical accuracy of discretized coefficients [1], allowing this form to be implemented in fixed point math, such as that found on inexpensive DSP chips and FPGAs.

The Biquad State Space (BSS) was introduced in [1] discrete time, state space form that allowed the numerical resiliancy of serial cascades of biquad filters to be moved into the state space world, while allowing for precalculation (Figure 1). This gave the structure excellent numerics and fixed and low latency, as previously described in [3]. The numerical robustness shown there is useful, even when minimal latency control was not an issue.

The rest of the paper will be organized as follows. Continuous time biquads will be discussed in Section II. The analog Biquad State Space form will be described in Section III. The invariance of the BSS under discretization will be discussed in Section IV. Some discussion of the matrix form will be in Section V. Discussion of lack of direct feedthrough will be in Section VI. Finally, Section VII gives some examples that show the effectiveness of the BSS in modeling flexible

*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

[1]Some have suggested Abramovitch State Space, but it suffers from a bad acronym.

dynamic systems, such as mechatronics.

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. Continuous Time Biquads

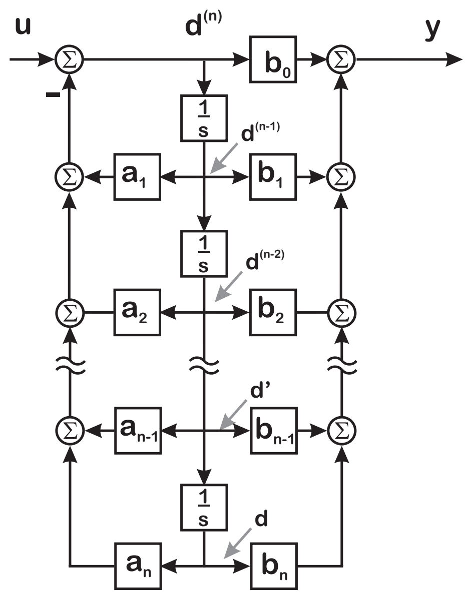

Fig. 2. An nth order continuous-time, polynomial filter in Direct Form II configuration similar to the discrete-time filter form in [1], [5].

A standard Single-Input, Single-Output (SISO) transfer function is shown in Figure 2, and represented as transfer function by

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

We can consider such high order polynomial transfer functions as filter models and factor these into a series of biquad filters such as:

$$\begin{aligned} \frac{Y(s)}{U(s)} \quad = \quad & \left(\tfrac{b_{00}s^2+b_{01}s+b_{02}}{s^2+a_{01}s+a_{02}}\right)\left(\tfrac{b_{10}s^2+b_{11}s+b_{12}}{s^2+a_{11}s+a_{12}}\right)\cdots \\ & \left(\tfrac{b_{m0}s^2+b_{m1}s+b_{m2}}{s^2+a_{m1}s+a_{m2}}\right). \end{aligned} \tag{2}$$

If n is even, then the number of biquads, m is set to $n/2$. If n is odd, then there are are $(n+1)/2$ biquads, but the last one is first order filter (by setting $b_{m,2} = a_{m,2} = 0$. As was shown in [3], [1], there can be advantages to factoring out the direct feedthrough gains, resulting in

$$\begin{aligned} \frac{Y(s)}{U(s)} \quad = \quad & b_{00}\left(\tfrac{s^2+\tilde{b}_{01}s+\tilde{b}_{02}}{s^2+a_{01}s+a_{02}}\right) b_{10}\left(\tfrac{s^2+\tilde{b}_{11}s+\tilde{b}_{12}}{s^2+a_{11}s+a_{12}}\right)\cdots \\ & b_{m0}\left(\tfrac{s^2+\tilde{b}_{m1}s+\tilde{b}_{m2}}{s^2+a_{m1}s+a_{m2}}\right), \end{aligned} \tag{3}$$

where $\tilde{b}_{ij} = \frac{b_{ij}}{b_{i0}}$. Note that if $b_{i0} = 0$ and $b_{i1} \neq 0$, then the factored out gain is b_{i1}. Likewise if both b_{i0} and b_{i1} are 0, then then the factored out gain is b_{i2}.

If we call this transfer function $H(s)$, then $H(s) = b_{m0}\cdots b_{10}b_{20}\tilde{H}(s)$ which gives:

$$\begin{aligned} \tilde{H}(s) \quad = \quad & \left(\tfrac{s^2+\tilde{b}_{01}s+\tilde{b}_{02}}{s^2+a_{01}s+a_{02}}\right)\left(\tfrac{s^2+\tilde{b}_{11}s+\tilde{b}_{12}}{s^2+a_{11}s+a_{12}}\right)\cdots \\ & \left(\tfrac{s^2+\tilde{b}_{m1}s+\tilde{b}_{m2}}{s^2+a_{m1}s+a_{m2}}\right). \end{aligned} \tag{4}$$

Again, if one of the b_{i0} terms is 0 it is replaced by the first non-zero b_{i1} or b_{i2} term.

Returning to a more modal representation, a single biquad can be represented as:

$$B(s) = K\left(\frac{s^2 + 2\varsigma_n\omega_n s + \omega_n^2}{s^2 + 2\varsigma_d\omega_d s + \omega_d^2}\right) \tag{5}$$

which in turn can be represented in a two step differential form as:

$$\begin{aligned} \ddot{x} \quad &+ \quad 2\varsigma_d\omega_d\dot{x} + \omega_d^2 x = u \\ y \quad &= \quad K\left(\ddot{x} + 2\varsigma_n\omega_n\dot{x} + \omega_n^2 dx\right) \end{aligned} \tag{6}$$

This can be represented in state space form as:

$$\left[\begin{array}{c} \ddot{x} \\ \dot{x} \end{array}\right] = \left[\begin{array}{cc} -2\varsigma_d\omega_d & -\omega_d^2 \\ 1 & 0 \end{array}\right]\left[\begin{array}{c} \dot{x} \\ x \end{array}\right] + \left[\begin{array}{c} 1 \\ 0 \end{array}\right] u \tag{7}$$

and

$$y = K\left[\begin{array}{cc} 2\varsigma_n\omega_n & \omega_n^2 \end{array}\right]\left[\begin{array}{c} \dot{x} \\ x \end{array}\right] + K\ddot{x} \tag{8}$$

but we need to get rid of $\ddot{x}$ and get the output in terms of the actual state vector:

$$\begin{aligned} y \quad = \quad & K\left[\begin{array}{cc} 2\varsigma_n\omega_n & \omega_n^2 \end{array}\right]\left[\begin{array}{c} \dot{x} \\ x \end{array}\right] \\ & -K\left[\begin{array}{cc} 2\varsigma_d\omega_d & \omega_d^2 \end{array}\right]\left[\begin{array}{c} \dot{x} \\ x \end{array}\right] + \left[\begin{array}{c} K \\ 0 \end{array}\right] u \end{aligned} \tag{9}$$

$$\begin{aligned} y \quad = \quad & \left[\begin{array}{cc} K(2\varsigma_n\omega_n - 2\varsigma_d\omega_d) & K(\omega_n^2 - \omega_d^2) \end{array}\right]\left[\begin{array}{c} \dot{x} \\ x \end{array}\right] \\ & + \left[\begin{array}{c} K \\ 0 \end{array}\right] u \end{aligned} \tag{10}$$

What is important in this structure is that the output depends on the first two states and the input. In this case, the input can feed through directly. Now, we would like to move this to a more general form such as we had in [3] and [1], so we replace these resonance parameters with filter coefficients:

$$\left[\begin{array}{c} \ddot{x}_i \\ \dot{x}_i \end{array}\right]\left[\begin{array}{cc} -a_{i1} & -a_{i2} \\ 1 & 0 \end{array}\right]\left[\begin{array}{c} \dot{x}_i \\ x_i \end{array}\right] + \left[\begin{array}{c} 1 \\ 0 \end{array}\right] u_i \tag{11}$$

while the state output equation is given by:

$$\left[\begin{array}{c} \tilde{y}_i \end{array}\right] = \left[\begin{array}{cc} \tilde{b}_{i1} - a_{i1} & \tilde{b}_{i2} - a_{i2} \end{array}\right]\left[\begin{array}{c} \dot{x}_i \\ x_i \end{array}\right] + \left[\begin{array}{c} 1 \end{array}\right] u_i \tag{12}$$

Finally, the properly scaled output is generated via:

$$\left[\begin{array}{c} y_i \end{array}\right] = \left[\begin{array}{c} b_{i0} \end{array}\right]\left[\begin{array}{c} \tilde{y}_i \end{array}\right]. \tag{13}$$

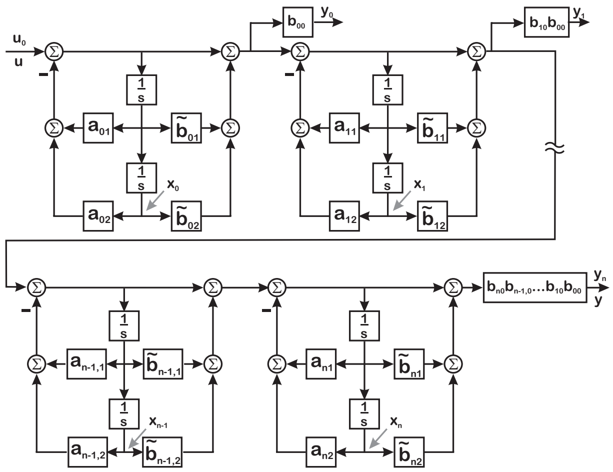

Fig. 3. The analog biquad cascade, with factored out $b_{i,0}$ terms and scaling the output of each block. This is completely analogous to the digital form of Figure 1.

III. The Analog Biquad State Space Form

If we have multiple biquads of the form shown in Equations 11, 12, and 13, we can chain these together by noting that:

$$\begin{aligned} u_i &= \tilde{y}_i, && \text{for } 0 \leq i < n, \\ u_0 &= u, && \text{and} \\ \tilde{y}_n &= \tilde{y}. \end{aligned} \tag{14}$$

If one is willing to go through the algebraic pain and suffering of applying Equation 14 to each biquad structure a very regular state space structure results. For a 3-biquad model (with no $b_{i0} = 0$), we get the state equation of 15. The unscaled output is in Equation 16, both displayed in Figure 4 due to their size. Finally, the properly scaled outputs are generated via:

$$\begin{bmatrix} y_2 \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} b_{20}b_{10}b_{00} & 0 & 0 \\ 0 & b_{10}b_{00} & 0 \\ 0 & 0 & b_{00} \end{bmatrix} \begin{bmatrix} \tilde{y}_2 \\ \tilde{y}_1 \\ \tilde{y}_0 \end{bmatrix}. \tag{17}$$

This structure has a very regular iteration which continues with the addition of extra biquads. It is worth noting several properties of this structure.

- First of all, it is a relatively sparse structure where a lot of the multiplies are 1.
- Secondly, we have put off multiplying by gain terms until the end. This provides the same input output behavior as the transfer function model, although the internal states may not be scaled the same way as the internal signals in the biquad chain. We will discuss alternate choices of where to assign gain scaling in [6].
- The eigenvalues of the state matrix are still defined by the denominator terms of the transfer function, and these show up in the "block diagonals" of the state matrix.
- The off diagonals contain differences of the numerator and denominator coefficients. Proper selection of these terms can minimize these differences and keep the size of the off diagonal terms well constrained.

It is worth discussing what it means to select these terms, the $\tilde{b}_{i1} - a_{i1}$ and $\tilde{b}_{i2} - a_{i2}$ terms. In the case of a biquad,

$$b_{i1} - a_{i1} = 2\varsigma_{in}\omega_{in} - 2\varsigma_{id}\omega_{id} \text{ and } b_{i2} - a_{i2} = \omega_{in}^2 - \omega_{id}^2. \tag{18}$$

This structure then allows the designer to pick pole/zero or resonance/ant-resonance combinations that minimize the off diagonal terms in the system matrix, the $\tilde{b}_{i1} - a_{i1}$ and $\tilde{b}_{i2} - a_{i2}$ terms, as well as their effect on the output.

What we will see in the next section, is that the biquad matrix structure is the same for discrete time biquads, although the physical interpretation of the coefficients is different. However, it is helpful to keep in mind the similarity of the numerical structures.

IV. Discretization of the Analog BSS

One major difference in using the BSS compared to general textbook methods is that we choose to discretize the BSS on a biquad by biquad basis. While this looses the satisfaction of analytical mathematical exactness, it does have the following positive properties:

1) Discretization approximations, and therefore discretization errors, are on a biquad by biquad basis. This has the potential to bound the error growth as the number of biquads (and therefore the number of states) grows.
2) The discretization method most appropriate to any one biquad can be applied independently of how adjacent biquads are discretized. For example, with lightly damped resonance/anti-resonance pairs, the pole zero mapping used in [3] and the Δ coefficients of [4] work extremely well. On the other hand, representing a double integrator as a discrete biquad can be accomplished using a Trapezoidal Rule equivalent [7] as described in [1].
3) Moreover, discretizing on a biquad by biquad basis means that the digital BSS for a given system has largely the same block structure as its analog BSS.

If one considers debugging a physical system, the importance of the last item cannot be overstated. The "invariance under discretization" means that a discrete state space model can be compared to an analog state space model or to modal test points on a physical system. It means that we can closely relate the digital state space model to the physics of the problem, and therefore to physical measurements of real systems.

V. The Matrices, Reloaded

Generating coefficients from continuous time biquad parameters is discussed in some detail in [3] and [4]. 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 15 has a very regular, block upper triangular form. On the block diagonals

$$\begin{bmatrix} \ddot{\tilde{x}}_2 \\ \dot{\tilde{x}}_2 \\ \ddot{\tilde{x}}_1 \\ \dot{\tilde{x}}_1 \\ \ddot{\tilde{x}}_0 \\ \dot{\tilde{x}}_0 \end{bmatrix} = \begin{bmatrix} -a_{21} & -a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_{11} & -a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{01} & -a_{02} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\tilde{x}}_2 \\ \tilde{x}_2 \\ \dot{\tilde{x}}_1 \\ \tilde{x}_1 \\ \dot{\tilde{x}}_0 \\ \tilde{x}_0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u \tag{15}$$

$$\begin{bmatrix} \tilde{y}_2 \\ \tilde{y}_1 \\ \tilde{y}_0 \end{bmatrix} = \begin{bmatrix} \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 0 & 0 & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \end{bmatrix} \begin{bmatrix} \dot{\tilde{x}}_2 \\ \tilde{x}_2 \\ \dot{\tilde{x}}_1 \\ \tilde{x}_1 \\ \dot{\tilde{x}}_0 \\ \tilde{x}_0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u. \tag{16}$$

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

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

$$\left[\begin{array}{cc} \tilde{b}_{i,1} - a_{i,1} & \tilde{b}_{i,2} - a_{i,2} \end{array}\right] \tag{19}$$

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 16, these same subsections in (19) 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 (17).

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 (19), 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.

VI. Handling Lack of Direct Feedthrough

One of the nice properties of the BSS is that it handles direct feedthrough from the input to the output in a systematic structure. In the discrete time world, we can provide direct feedthrough for models of analog systems by choice of discretization method. For example, the analog double integrator inserted into the discrete BSS in [1] was discretized with the Trapezoidal Rule approximation, which gave it direct feedthrough. In the analog world, the rationale for this does not exist and since most mechatronic systems have some sort of low frequency behavior that has a pole zero excess (e.g. double integrator, simple resonance), we need to know how to accommodate this.

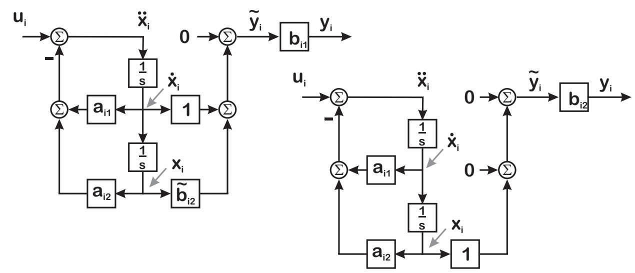

Fig. 5. Analog biquads without direct feedthrough. On the left, $b_{i0} = 0$. On the right, both b_{i0} and $b_{i1} = 0$. In either case, the leading gain is the gain of the highest order numerator term that has a non-zero coefficient. In either case, the lack of direct feedthrough means that the output is only determined by the state of the block. All downstream blocks from this one will not have direct feedthrough from the cascade input to the cascade output.

Figure 5 shows two examples of biquads tasked with modeling such systems. On the left side is a biquad model for a system where only $b_{i0} = 0$. This would model a pole zero excess of 1. On the right, both b_{i0} and b_{i1} are 0. In either case, we factor out the non-zero b_{ij} corresponding to the highest order. This will be used in our downstream gain calculations. Note that when any such biquad is in the chain, the direct feedthrough from the system input, u, to any of the downstream inputs and outputs, u_j and y_k for $j > i$ and $k \geq i$, is 0. This affects the form of our state matrices.

In both cases, the state equation from (11) is unchanged. However, the state output equations change a lot. In the left hand case, Equation 12 becomes

$$\left[\ \tilde{y}_i\ \right] = \left[\ 1 \quad \tilde{b}_{i2}\ \right] \begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + \left[\ 0\ \right] u_i \tag{20}$$

where $\tilde{b}_{i2} = b_{i2}/b_{i1}$ and (13) becomes:

$$\left[\ y_i\ \right] = \left[\ b_{i1}\ \right] \left[\ \tilde{y}_i\ \right]. \tag{21}$$

In the right hand case, Equation 12 becomes

$$\left[\ \tilde{y}_i\ \right] = \left[\ 0 \quad 1\ \right] \begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + \left[\ 0\ \right] u_i \tag{22}$$

and (13) becomes:

$$\left[\ y_i\ \right] = \left[\ b_{i2}\ \right] \left[\ \tilde{y}_i\ \right]. \tag{23}$$

This may seem like an awful lot of bookkeeping for such a simple concept, but doing this bookkeeping allows us to maintain a the overall system structure, which allows us to write scripts and programs to build up BSS matrices from individual biquad models.

To illustrate this, consider a 4-biquad system model. We choose $b_{i0} = 0$ for biquad 1. (Here the first biquad in the chain is biquad 0 and the last one is biquad 3. Again, algebraic pain and suffering results in a very regular state space structure. For our 4-biquad model, we get the state equation of 24. The unscaled output is in Equation 25, both displayed in Figure 6 due to their size. Finally, the properly scaled outputs are generated via:

$$\begin{bmatrix} y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} b_{30}b_{20}b_{1x}b_{00} & 0 & 0 & 0 \\ 0 & b_{20}b_{1x}b_{00} & 0 & 0 \\ 0 & 0 & b_{12}b_{00} & 0 \\ 0 & 0 & 0 & b_{00} \end{bmatrix} \begin{bmatrix} \tilde{y}_3 \\ \tilde{y}_2 \\ \tilde{y}_1 \\ \tilde{y}_0 \end{bmatrix}, \tag{26}$$

where $b_{ix} = b_{i1}$ if $b_{i1} \neq 0$ and $b_{ix} = b_{i2}$ if $b_{i1} = 0$. In Equations 24 and 25 $\tilde{b}_{11} = 1$ and $\tilde{b}_{12} = b_{12}/b_{11}$ if $b_{10} = 0$ and $b_{11} \neq 0$. Similarly, if $b_{10} = 0$ and $b_{11}00$, then $\tilde{b}_{11} = 0$ and $\tilde{b}_{12} = 1$. Note that the direct feedthrough from the input to any outputs downstream of biquad 1 is blocked. Also note that direct feedthrough of any states upstream of biquad 1 to any states downstream of biquad 1 is also blocked. Like the input, those states affect the downstream states through the output of biquad 1. However, it is clear that they still have a regular structure.

VII. Examples

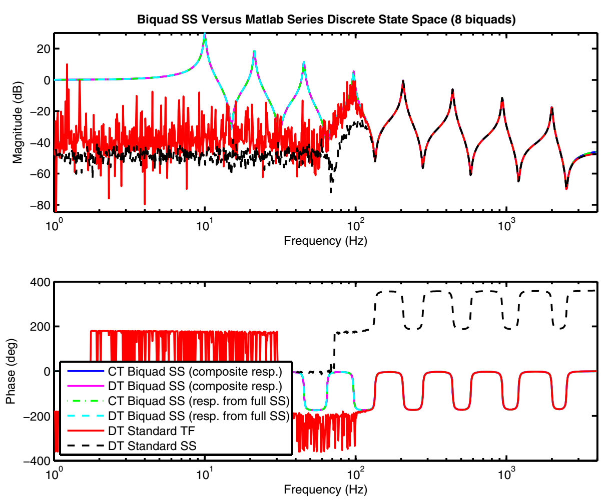

Fig. 7. Comparison of Bode plots from continuous and discrete BSS, as well as standard discrete transfer function and state space forms. The conventional methods fall apart with 8 biquads, while the BSS methods retain their numerical integrity.

In order to demonstrate the numerical improvements arising from the biquad state space structure, some simple examples were generated in Matlab. A serial biquad structure was generated from resonance and anti-resonance parameters. The natural frequencies of the resonances were logarithmically spaced between 10 Hz and 2000 Hz, while the natural frequencies of the anti-resonances were spaced between 15 Hz and 2500 Hz. Numerator and denominator damping

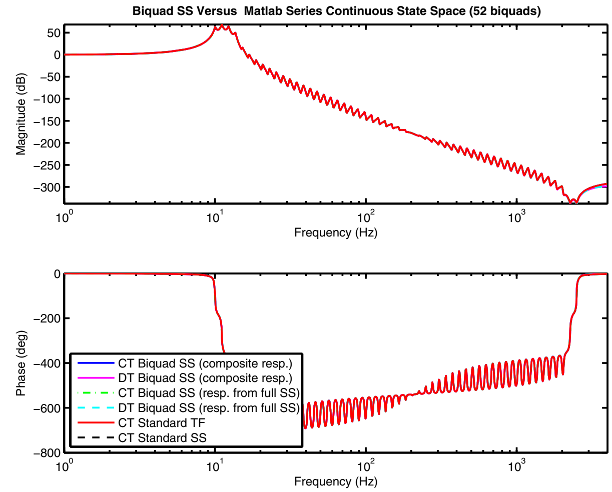

Fig. 8. Comparison of Bode plots from continuous and discrete BSS, as well as standard continuous transfer function and state space forms. All the methods retain numerical integrity with 52 biquads.

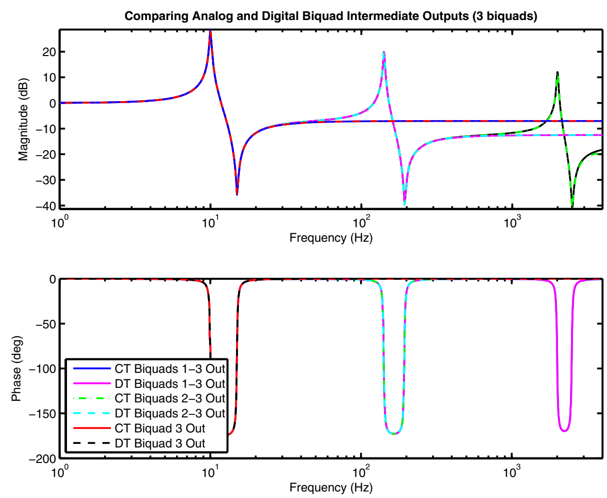

Fig. 9. Comparing analog and discrete BSS outputs. With 3 biquads, the outputs of each analog biquad section was plotted against it's digital version. This demonstrates the invariance of the biquad outputs under discretization.

factors were set at 0.01. The sample frequency was chosen at 8 kHz for the discrete biquads. The script could then specify any number of resonance/anti-resonance pairs to fill that frequency range. As the baseline, the frequency responses of each individual biquad was generated and these responses were summed to create a composite response. Since the responses were generated from individual biquads, it was thought that they would be less susceptible to numerical issues.

In order to compare the biquad state space to more conventional methods, the resonance/anti-resonance 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 composite Bode plots described above. These are the “standard” or “conventional” methods. Similarly, model terms were used to construct both continuous and discrete

$$
\begin{bmatrix} \ddot{\tilde{x}}_3 \\ \dot{\tilde{x}}_3 \\ \ddot{\tilde{x}}_2 \\ \dot{\tilde{x}}_2 \\ \ddot{\tilde{x}}_1 \\ \dot{\tilde{x}}_1 \\ \ddot{\tilde{x}}_0 \\ \dot{\tilde{x}}_0 \end{bmatrix} = \begin{bmatrix} -a_{31} & -a_{32} & \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_{21} & -a_{22} & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{11} & -a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -a_{01} & -a_{02} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\tilde{x}}_3 \\ \tilde{x}_3 \\ \dot{\tilde{x}}_2 \\ \tilde{x}_2 \\ \dot{\tilde{x}}_1 \\ \tilde{x}_1 \\ \dot{\tilde{x}}_0 \\ \tilde{x}_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u \qquad (24)
$$

$$
\begin{bmatrix} \tilde{y}_3 \\ \tilde{y}_2 \\ \tilde{y}_1 \\ \tilde{y}_0 \end{bmatrix} = \begin{bmatrix} \tilde{b}_{31}-a_{31} & \tilde{b}_{32}-a_{32} & \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 0 & 0 & \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \end{bmatrix} \begin{bmatrix} \dot{\tilde{x}}_3 \\ \tilde{x}_3 \\ \dot{\tilde{x}}_2 \\ \tilde{x}_2 \\ \dot{\tilde{x}}_1 \\ \tilde{x}_1 \\ \dot{\tilde{x}}_0 \\ \tilde{x}_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u. \qquad (25)
$$

Fig. 6. State equations for continuous time biquad state space with scalar output scaling. Biquad 1 lacks direct feedthrough.

biquad state space structures and again, Bode plots were 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.

Figure 7 shows an 8 biquad case where the conventional discrete time structures (transfer function and state space) have numerical difficulties. Note that both continuous and discrete BSS structures produce plots right on top of the composite plots.

In Figure 8 the composite plots and continuous and discrete BSS structure plots are compared to conventional continuous methods. In this case all methods produce identical plots, even with a 52 biquad structure.

Figure 9 plots a 3-biquad structure, and in this case we plot neither conventional methods nor the composite plots. Instead we tap off the individual biquad outputs of the first, second, and third biquads so that we can demonstrate the almost exact match of the discrete biquads to the continuous biquads.

As mentioned earlier, this "invariance under discretization" is a very useful property. In particular, it allows one to construct an analog model from physical principles, convert this model to an analog Biquad State Space form, convert that to a discrete time Biquad State Space form for implementation, and then easily extract information about the continuous model from the discrete model results.

VIII. Conclusions

The biquad state space (BSS) form adapts the multinotch filter [3] for state space use, preserving the latter's excellent numerical properties, as described in [1]. 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. Moreover, both the continuous BSS and the discrete BSS of [1] have a regular and repeatable structure, as with a canonical form. This makes it relatively straightforward to generate the matrices from modal parameters in an automated fashion.

(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.)

References

[1] D. Y. Abramovitch, "The discrete time biquad state space structure: Low latency with high numerical fidelity," in *Submitted to the 2015 American Control Conference*, (Chicago, IL), AACC, IEEE, July 2015.

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

[3] 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.

[4] 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.

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

[6] D. Y. Abramovitch, "Variations on the biquad state space structure," in *Submitted to the 2015 Multi-conference on Systems and Control*, (Sydney, Australia), IEEE, IEEE, September 2015.

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

[8] D. Y. Abramovitch and E. Johnstone, "State space system simulator utilizing bi-quadratic blocks to simulate lightly damped resonances," Patent Application PCT/US1227149, Agilent Technologies, Santa Clara, CA USA, February 29 2012.

[9] 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.