Material from this preprint was presented as a tutorial at MECC 2023.

A Tutorial on the Biquad and Bilinear State-Space Structures ⋆

Daniel Y. Abramovitch *

* *Mass Spec Division, Agilent Technologies, 5301 Stevens Creek Blvd., M/S: 3U-WT, Santa Clara, CA 95051 USA (e-mail: danny@agilent.com).*

Abstract: This tutorial paper will focus on the author's state-space structures, the Biquad State Space (BSS) and the Bilinear State Space (BLSS). These two structures are have shown some remarkable advantages in the modeling and control of mechatronic systems, including numerical stability and model explainability. Furthermore, they have the remarkable property that the states of digital BSS and BLSS structures correspond to the states of the analog BSS and BLSS structures, at the outputs of each biquad or bilinear section. This gives the control engineer the ability to connect their digital model far more closely to the physical system, allowing digital "scope probes" to compare the same signals as one might get from the physical system. We believe that adapting these structures will go a long way towards making it easier and more practical to apply state-space methods to higher order models while still retaining physical intuition once they are discretized.

Keywords: Mechatronic systems, state-space modeling, physical realization.

1. MOTIVATION: WHY ANOTHER STATE-SPACE FORM?

One of the easiest ways to clear a room full of practicing mechatronic control engineers is to suggest that they employ state-space methods for the control of their structure with many lightly-damped resonances. State-space models of highly flexible systems can present severe numerical issues. The models derived from physical principles often lack structure. Canonical form models (Kailath (1980a)), 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.

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 (Kailath (1980a)) 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 sensitive 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 tutorial will familiarize the audience with a set of structures that meet these needs. The first of these is the Biquad State Space (BSS) (Abramovitch (2015c,a)), based on the multinotch structure (Abramovitch (2015e,g)). The BSS captures the endearing characteristics of the multinotch while providing the flexibility of model based control. A significant feature of the BSS is the ability to move easily between the states of the continuous and discrete time forms, as they are tightly related to the output of the biquads. The material here has largely been published earlier in the separate works noted, but it is hoped that putting them together in this way will make for a more cohesive picture.

The BSS has several desirable characteristics:

⋆ Daniel Y. Abramovitch is a system architect in the Mass Spec Division at Agilent Technologies.

- 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 with high numerical stability, even through discretization. The Δ-parameters from the multinotch (Abramovitch (2015g)) can be used to improve the numerical accuracy of discretized coefficients (Abramovitch (2015c)), allowing this form to be implemented in fixed point math, such as that found on inexpensive DSP chips and FPGAs.

The BSS is excellent for capturing the flexible modes of a system, but with any system model, we also need to allow for rigid body modes. More specifically, we would also like that rigid body state-space structure to have an equivalence between the continuous and discrete time forms. The discussion of forms not handled well by standard biquads starts in Section 12. Specifically, adding low-pass filters (LPF) is described in Section 13. Looking at separate rigid body states is accomplished via the Bilinear State Space (BLSS) structure, described in Section 15 (Abramovitch (2018)). Together, these allow us to divide and conquer the state-space model. In doing so, they let us stay close to a model parameter representation. This, in turn, allows us to consider the systematic effects of a parameter change without getting lost in numerical differences every time the sample rate changes by 1%.

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.

The BSS does most of the heaving lifting here. It allows us to map a biquad filter cascade into a state-space form. Part of the elegance of this form is that it handles cases with direct feedthrough easily. However, as we adopt continuous-time models of physical dynamics we see many examples of systems without direct feedthrough. As we discretize these we have several choices, some of which preserve the direct feed through in the discrete domain (by mapping zeros at $s = -\infty$) to $z = -1$) and some that break it. Those options will be discussed in Section 14.

The rest of the paper will be organized as follows. The Multinotch, which is the origin of these structures, will be reviewed in Section 2. The Biquad State Space (BSS) will be first described in Section 3. We will explain how this structure can be employed in a current estimator (which would include a Kalman filter) and state feedback in Section 4. We will switch from discrete to continuous-time biquads in Section 5, and show how modulo the z to s switch, the structures look very much the same. We will introduce the Analog Biquad State Space (ABSS) Form in Section 6. This will naturally lead us to discussions of discretization choices for analog biquads in Section 7. Section 8 will get into more of the structural similarities between discrete and continuous forms.

We will then show some examples of the utility of the BSS, with some discrete-time examples in Section 9 and some examples showing how closely the CT and DT responses match in Section 10. We will show one early example of adding rigid body modes to the BSS with a simple double integrator in Section 11, but will later see that this is limited. The whole discussion of handling rigid body modes will begin in earnest in Section 12. First we will discuss continuous-time, rigid-body dynamics and low-pass filters in Section 13. These both exhibit a lack of direct feedthrough, discussed in Section 14. In Section 15, we introduce an unbundled Bilinear State Space (BLSS) form – an option when the rigid body poles and zeros are real – which can easily replace a biquad block without much pain, and allow convenient access to the internal state of a rigid body model, particularly unique in discrete time state-space formulations.

Again, the "divide-and-conquer" nature of building up a state-space via the BSS or BLSS requires that we think about how we will discretize these sections. This is discussed in Section 16. This leads into a discussion of discrete-time, rigid-body models in Section 17, with the goal of matching these to their continuous-time counterparts. Finally, Section 18 will provide a few examples to aide in visualizing these tradeoffs. Again, the goal is to provide the missing algorithmic "LEGO Blocks" needed to make our MNF and BSS models more complete. We end with the proverbial summary in Section 19. This entire approach rests upon being willing to discretize a section of the state-space structure at a time, which flies in the face of the conventional theoretical construct of discretizing the entire model, usually via a Zero-Order Hold (ZOH) Equivalent. This discussion will be presented in (Abramovitch (2023)).

Two brief notes about notation. The first is that the state-space diagrams use the same "mixed-metaphor" combinations of time and frequency notation used in (Franklin et al. (2006)), (Franklin et al. (1998a)), and (Kailath (1980b)). While s blocks and signal derivatives belong to different domains, the meaning should be clear from the context and it provides a compact way to encode information in the block diagrams. Likewise z^{-1} blocks have signals with time shift notation going in and out of them, e.g. $x_k, x(k)$. In difference equations the z^{-1} becomes the unit delay operator, similar to q^{-1}, but we still recognize the that we can also get a frequency response from the structure with z^{-1}. Although, inexact, this usage is common and well understood.

The second is that both continuous and discrete biquads feature the parameter sets of $\{a_{i1}, a_{i2}\}$ and $\{b_{i0}, \tilde{b}_{i1}, \text{ and} \tilde{b}_{i2}\}$. In other contexts, it has been convenient to denote one of those sets with an extra C or D subscript to differentiate one or the other as continuous or discrete (or to move between Roman and Greek alphabets). However, they have seemed cumbersome in the notation of this paper, and so they will be differentiated here by their context. It is in fact the similarity of structures between the discrete and

continuous BSS that is one of the strengths of this format. If one takes a continuous transfer function equation, reparameterizes it as a chain of biquads, puts that chain into the continuous BSS form as done in (Abramovitch (2015b)), and then does a biquad-by-biquad discretization to the discrete BSS (Abramovitch (2015d)), then the outputs of the CT biquads and the DT biquads correspond. The internal coefficients of each biquad form (CT vs. DT) are of course different, but the matrix structures and the biquad input-output relationships are consistent. This allows one to tap a discrete time model signal deep within the state-space structure that is equivalent to a continuous time signal from a physical parameter.

2. THE MULTINOTCH

The development of the BSS starts with the Multinotch, a way of turning a polynomial form IIR filter as diagrammed in Figure 1, into a cascade of biquads with the direct feedthrough coefficients factored out to the end, as shown in Figure 3 (Abramovitch (2015e)). With this factorization, one can discretize each biquad individually so that the discretized biquads have a one to one correspondence with the continuous-time biquads. By judicious choices of the which poles and zeros from the physical model are assigned to each biquad, one can minimize effects of the signals of any one biquad on the others. The Multinotch is a highly efficient digital filter because it not only has greater numerical stability than standard state-space forms, but also allows for precalculation of most of the filter, minimizing the latency between reading a sample and responding to it. This particular form of the direct feed-through scaling allows for precalculation, but others are available if we want the internal states of the filter to be scaled as they are in the system.

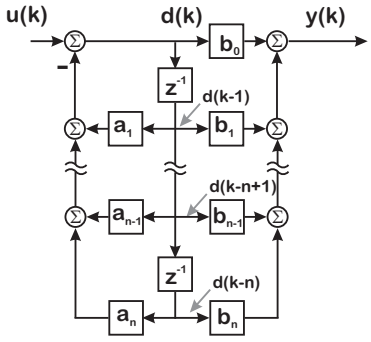

Fig. 1. An nth order polynomial filter in Direct Form II configuration (Oppenheim and Schafer (1975)).

There are many potential causes of time delay in a feedback system, many of which are outlined in (Abramovitch et al. (2023)). Here, we limit ourselves to computational latency. 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 (Franklin et al. (1998b); Åström and Wittenmark (1990)). Figure 2 illustrates how the lack of precalculation makes the closed-loop latency dependent on the controller size. Restructuring the calculation to push as much as possible into precalculation makes the computational latency fixed and shorter. It is relatively

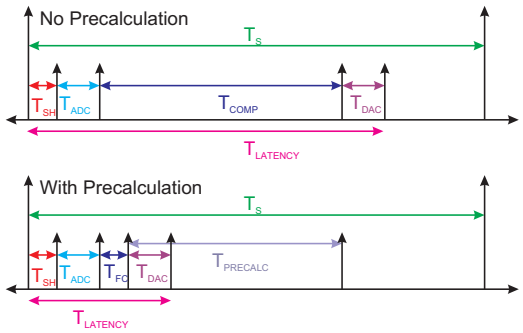

Fig. 2. Input and output timing in a digital control system. The top drawing is without precalculation; the bottom drawing is with. Note that precalculation can be started as soon as the output has been sent to the DAC and therefore is in parallel with the DAC conversion time. The computation time, T_{COMP}, 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.

straightforward to apply precalculation on a controller implemented as an IIR filter as 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 1, 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 (Texas Instruments (1993)):

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

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 (Franklin et al. (1998b)):

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

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

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, $k-1$ (Åström and Wittenmark (1990)). 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.

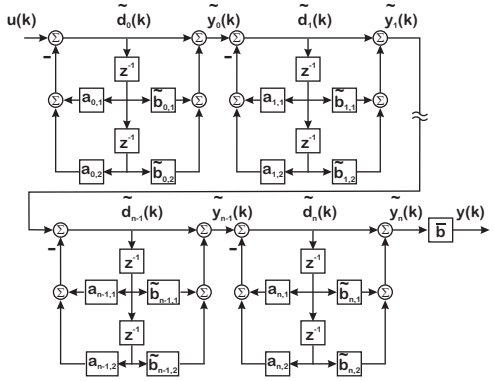

Fig. 3. The updated biquad cascade, with factored out b_0 terms.

The precalculation is helpful, but polynomial filters have poor numerical properties, particularly when the filter has lightly-damped poles and zeros. Since these are common in mechatronic systems, we want to implement our control filters using a biquad cascade that has better numerical properties than a polynomial filter. In (Abramovitch (2015e); Abramovitch and Moon (2012)), 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 (Abramovitch (2015g)) 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 (Abramovitch and Johnstone (2013)). 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. 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 (Kailath (1980a)), 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 11.

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 (Abramovitch (2015a)). 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 9.

For all the advantages of the BSS for flexible modes, we still need to find some way to not only represent rigid body modes, but also to have the internal states of those structures correspond to the internal states of the rigid body, e.g. velocity and position. Figures 18 and 20 show continuous and discrete versions of the Bilinear State Space form (BLSS) (Abramovitch (2018)) which accomplishes that, and can be combined with a cascade of biquads into one overall state-space structure.

The implementation of controllers and filters is most often digital. Whether in transfer function or state-space realizations, filters and controllers often end up as polynomial form, i.e. filters where the numerator and denominator are polynomials in s, z, z^{-1}, or q^{-1}. Consider a continuous time (CT) filter,

$$F(s) = \frac{b_{0,c}s^n + b_{1,c}s^{n-1} + \cdots + b_{n-1,c}s + b_{n,c}}{s^n + a_{1,c}s^{n-1} + \cdots + a_{n-1,c}s + a_{n,c}}, \tag{5}$$

which has to be discretized for implementation on a real-time computer. A function of z^{-1}

$$F(z^{-1}) = \frac{b_0 + b_1 z^{-1} + \cdots + b_{n-1}z^{-n+1} + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_{n-1}z^{-n+1} + a_n z^{-n}}, \tag{6}$$

allows us to express the output directly as a combination of past outputs and inputs:

$$\begin{aligned} yk) \;=\; & -a_1 y(k-1) + \cdots - a_{n-1}y(k-n+1) \\ & -a_n y(k-n) + b_0 u(k) + b_1 u(k-1) + \\ & \cdots + b_{n-1}u(k-n+1) + b_n u(k-n). \end{aligned} \tag{7}$$

The issues with these filters (or canonical state-space forms (Kailath (1980b))) are:

a) While they are compact, they obscure physical intuition.
b) Any residual physical intuition is lost in the discretization process, especially in moving from continuous to discrete state-space models.
c) The discrete polynomial forms are often badly conditioned, especially for high Q dynamics.
d) As the sample rate goes up relative to the dynamics in the filter/state-space, large changes in physical parameters can be scattered into a few bits of the digital polynomial coefficients (Abramovitch (2015h)).

The Multinotch Filter (MNF) (Abramovitch (2015f,h)) was created to address these issues by breaking up polynomial form filters into a serial cascade of biquads (Fig. 4), while maintaining the ability to do precalculation for minimum latency control. The individual biquads are parameterized as:

$$\frac{Y_i(z^{-1})}{U_i(z^{-1})} = B_i(z^{-1}) = b_{i,0}\left(\frac{1 + \tilde{b}_{i,1}z^{-1} + \tilde{b}_{i,2}z^{-2}}{1 + a_{i,1}z^{-1} + a_{i,2}z^{-2}}\right), \text{ with} \tag{8}$$

$$\begin{aligned} U_{i+1}(z^{-1}) &= \tilde{Y}_i(z^{-1}), \quad \text{for } 0 \le i < n, \\ U_0(z^{-1}) &= U(z^{-1}), \quad \text{and} \\ \tilde{Y}_n(z^{-1}) &= \tilde{Y}(z^{-1}). \end{aligned} \tag{9}$$

Biquad State Space (BSS) mapped the multinotch's biquad structure to state space, first in discrete time (DT) (Abramovitch (2015d)) and then in continuous time (CT) (Abramovitch (2015b)). It was found that by discretizing the model on a biquad by biquad basis, the essential structure was preserved across discretization. That is, the outputs of the CT biquads mapped to the outputs of the DT biquads. However, in all that work, there were some fundamental pieces left out. This paper fills in those pieces.

3. THE BIQUAD STATE SPACE (BSS)

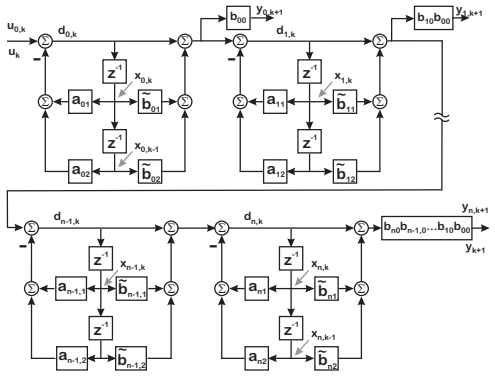

Fig. 4. The updated discrete biquad cascade, with factored out $b_{i,0}$ terms and scaling the output of each block.

The digital version of the BSS shown in Figure 4 looks very similar to the Multinotch of Figure 3, although as we are more focused on accurate modeling than precomputation, we scale the outputs of each biquad to get to the proper output states. This form results in a block upper triangular state transition matrix (Abramovitch (2015c)). If one were to mistakenly substitute $1/s$ for z^{-1}, one would end up with the continuous time structure of Figure 8. Furthermore, if one were to discretize the structure of Figure 8 one biquad at a time, then one would end up with the structure of Figure 4, with the added advantage that the signals at the outputs of the biquads would correspond between the analog and digital versions.

There is no need to repeat the equations of (Abramovitch (2015e)) here, but looking at the structure the multinotch in Figure 4 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 4, $d_{i,k} = x_{i,k+1}$. 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 $x_{i,k+1}$.

We can look at each of these biquad sections as a state-space realization. In this case:

$$\begin{bmatrix} x_{i,k+1} \\ x_{i,k} \end{bmatrix} = \begin{bmatrix} -a_{i1} & -a_{i2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{i,k} \\ x_{i,k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_{i,k} \tag{10}$$

while the state output equation is given by:

$$[\,\tilde{y}_{i,k+1}\,] = [\,\tilde{b}_{i1} - a_{i1} \;\; \tilde{b}_{i2} - a_{i2}\,] \begin{bmatrix} x_{i,k} \\ x_{i,k-1} \end{bmatrix} + [\,1\,]\, u_{i,k} \tag{11}$$

Finally, the properly scaled output is generated via:

$$[\,y_{i,k+1}\,] = [\,b_{i0}\,]\,[\,\tilde{y}_{i,k+1}\,]\,. \tag{12}$$

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:

$$\begin{aligned} u_{i+1,k} &= \tilde{y}_{i,k+1}, \quad \text{for } 0 \le i < n, \\ u_{0,k} &= u_k, \quad \text{and} \\ \tilde{y}_{n,k+1} &= \tilde{y}_{k+1}. \end{aligned} \tag{13}$$

If one is willing to go through the algebraic pain and suffering of applying Equation 13 to each biquad structure a very regular state-space structure results. For a 3-biquad model, we get the state equation of 14. The unscaled output is in Equation 15, both displayed in Figure 5 due to their size. Finally, the properly scaled outputs are generated via:

$$\begin{bmatrix} y_{2,k+1} \\ y_{1,k+1} \\ y_{0,k+1} \end{bmatrix} = \begin{bmatrix} b_{20}b_{10}b_{00} & 0 & 0 \\ 0 & b_{10}b_{00} & 0 \\ 0 & 0 & b_{00} \end{bmatrix} \begin{bmatrix} \tilde{y}_{2,k+1} \\ \tilde{y}_{1,k+1} \\ \tilde{y}_{0,k+1} \end{bmatrix}. \tag{16}$$

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 9.

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.

4. 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

$$\begin{bmatrix} x_{2,k+1} \\ x_{2,k} \\ x_{1,k+1} \\ x_{1,k} \\ x_{0,k+1} \\ x_{0,k} \end{bmatrix} = \begin{bmatrix} -a_{21} & -a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_{11} & -a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{01} & -a_{02} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{2,k} \\ x_{2,k-1} \\ x_{1,k} \\ x_{1,k-1} \\ x_{0,k} \\ x_{0,k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_k \tag{14}$$

$$\begin{bmatrix} \tilde{y}_{2,k+1} \\ \tilde{y}_{1,k+1} \\ \tilde{y}_{0,k+1} \end{bmatrix} = \begin{bmatrix} \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 0 & 0 & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \end{bmatrix} \begin{bmatrix} x_{2,k} \\ x_{2,k-1} \\ x_{1,k} \\ x_{1,k-1} \\ x_{0,k} \\ x_{0,k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u_k \tag{15}$$

Fig. 5. State equations for discrete time Biquad State Space (BSS) with scalar output scaling.

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 2.

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 17. For the SISO case, there will only be a single output, and so the output equations become what is shown in Equation 18.

Finally, the properly scaled time update output is generated via a single multiplication of concatenated feedthrough coefficients, in a similar manner to (Abramovitch (2015e)).

$$\bar{y}_k = \bar{y}_{2,k} = b_{20}b_{10}b_{00}\tilde{y}_{2,k}. \tag{19}$$

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

$$e_k = y_{meas,k} - \bar{y}_k, \text{ and} \tag{20}$$

$$\hat{x}_k = \bar{x}_k + L_c e_k. \tag{21}$$

Now, Equation 20 involves one subtraction. Equation 21 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

$$u_{fb,k} = K_{fb}\hat{x}_k = K_{fb}\bar{x}_k + K_{fb}L_c e_k, \tag{22}$$

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

5. CONTINUOUS TIME BIQUADS

A standard Single-Input, Single-Output (SISO) transfer function is shown in Figure 7, and represented as transfer function by

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + b_2 s^{n-2} + \ldots + b_n}{s^n + a_1 s^{n-1} + a_2 s^{n-2} + \ldots + a_n}. \tag{23}$$

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

$$\frac{Y(s)}{U(s)} = \left(\frac{b_{00}s^2 + b_{01}s + b_{02}}{s^2 + a_{01}s + a_{02}}\right)\left(\frac{b_{10}s^2 + b_{11}s + b_{12}}{s^2 + a_{11}s + a_{12}}\right)\cdots \left(\frac{b_{m0}s^2 + b_{m1}s + b_{m2}}{s^2 + a_{m1}s + a_{m2}}\right). \tag{24}$$

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 (Abramovitch (2015e,c)), there can be advantages to factoring out the direct feedthrough gains, resulting in

$$\frac{Y(s)}{U(s)} = b_{00}\left(\frac{s^2 + \tilde{b}_{01}s + \tilde{b}_{02}}{s^2 + a_{01}s + a_{02}}\right) b_{10}\left(\frac{s^2 + \tilde{b}_{11}s + \tilde{b}_{12}}{s^2 + a_{11}s + a_{12}}\right)\cdots b_{m0}\left(\frac{s^2 + \tilde{b}_{m1}s + \tilde{b}_{m2}}{s^2 + a_{m1}s + a_{m2}}\right), \tag{25}$$

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:

$$\tilde{H}(s) = \left(\frac{s^2 + \tilde{b}_{01}s + \tilde{b}_{02}}{s^2 + a_{01}s + a_{02}}\right)\left(\frac{s^2 + \tilde{b}_{11}s + \tilde{b}_{12}}{s^2 + a_{11}s + a_{12}}\right)\cdots \left(\frac{s^2 + \tilde{b}_{m1}s + \tilde{b}_{m2}}{s^2 + a_{m1}s + a_{m2}}\right). \tag{26}$$

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

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

$$B(s) = K\left(\frac{s^2 + 2\varsigma_n\omega_n s + \omega_n^2}{s^2 + 2\varsigma_d\omega_d s + \omega_d^2}\right) \tag{27}$$

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

$$\begin{aligned} \ddot{x} \;+\; & 2\varsigma_d\omega_d\dot{x} + \omega_d^2 x = u \\ y \;=\; & K\left(\ddot{x} + 2\varsigma_n\omega_n\dot{x} + \omega_n^2 dx\right) \end{aligned} \tag{28}$$

$$\begin{bmatrix} \bar{x}_{2,k} \\ \bar{x}_{2,k-1} \\ \bar{x}_{1,k} \\ \bar{x}_{1,k-1} \\ \bar{x}_{0,k} \\ \bar{x}_{0,k-1} \end{bmatrix} = \begin{bmatrix} -a_{21} & -a_{22} & \tilde{b}_{11} - a_{11} & \tilde{b}_{12} - a_{12} & \tilde{b}_{01} - a_{01} & \tilde{b}_{02} - a_{02} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_{11} & -a_{12} & \tilde{b}_{01} - a_{01} & \tilde{b}_{02} - a_{02} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{01} & -a_{02} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_{2,k-1} \\ \hat{x}_{2,k-2} \\ \hat{x}_{1,k-1} \\ \hat{x}_{1,k-2} \\ \hat{x}_{0,k-1} \\ \hat{x}_{0,k-2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_{k-1}. \tag{17}$$

$$[\,\bar{\tilde{y}}_{2,k}\,] = [\,\tilde{b}_{21} - a_{21}\;\; \tilde{b}_{22} - a_{22}\;\; \tilde{b}_{11} - a_{11}\;\; \tilde{b}_{12} - a_{12}\;\; \tilde{b}_{01} - a_{01}\;\; \tilde{b}_{02} - a_{02}\,] \begin{bmatrix} \hat{x}_{2,k-1} \\ \hat{x}_{2,k-2} \\ \hat{x}_{1,k-1} \\ \hat{x}_{1,k-2} \\ \hat{x}_{0,k-1} \\ \hat{x}_{0,k-2} \end{bmatrix} + [\,1\,]\, u_{k-1} \tag{18}$$

Fig. 6. Time update equations for discrete time Biquad State Space (BSS) with scalar output scaling.

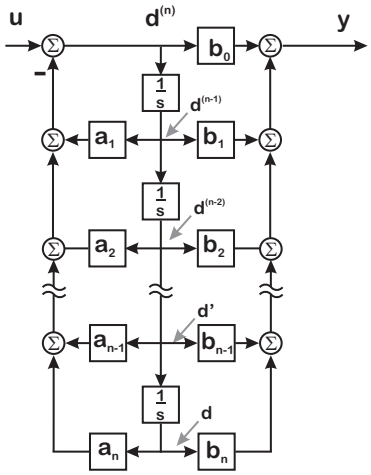

Fig. 7. An nth order continuous-time, polynomial filter in Direct Form II configuration similar to the discrete-time filter form in (Abramovitch (2015c); Oppenheim and Schafer (1975)).

This can be represented in state-space form as:

$$\begin{bmatrix} \ddot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -2\varsigma_d\omega_d & -\omega_d^2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{29}$$

and

$$y = K\,[\,2\varsigma_n\omega_n\;\; \omega_n^2\,] \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + K\ddot{x} \tag{30}$$

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

$$\begin{aligned} y \;=\; & K\,[\,2\varsigma_n\omega_n\;\; \omega_n^2\,] \begin{bmatrix} \dot{x} \\ x \end{bmatrix} \\ & -K\,[\,2\varsigma_d\omega_d\;\; \omega_d^2\,] \begin{bmatrix} \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} K \\ 0 \end{bmatrix} u \end{aligned} \tag{31}$$

$$\begin{aligned} y \;=\; & [\,K(2\varsigma_n\omega_n - 2\varsigma_d\omega_d)\;\; K(\omega_n^2 - \omega_d^2)\,] \begin{bmatrix} \dot{x} \\ x \end{bmatrix} \\ & + \begin{bmatrix} K \\ 0 \end{bmatrix} u \end{aligned} \tag{32}$$

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 (Abramovitch (2015e)) and (Abramovitch (2015c)), so we replace these resonance parameters with filter coefficients:

$$\begin{bmatrix} \ddot{x}_i \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} -a_{i1} & -a_{i2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_i \tag{33}$$

while the state output equation is given by:

$$[\,\tilde{y}_i\,] = [\,\tilde{b}_{i1} - a_{i1}\;\; \tilde{b}_{i2} - a_{i2}\,] \begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + [\,1\,]\, u_i \tag{34}$$

Finally, the properly scaled output is generated via:

$$[\,y_i\,] = [\,b_{i0}\,]\,[\,\tilde{y}_i\,]\,. \tag{35}$$

6. THE ANALOG BIQUAD STATE SPACE (ABSS) FORM

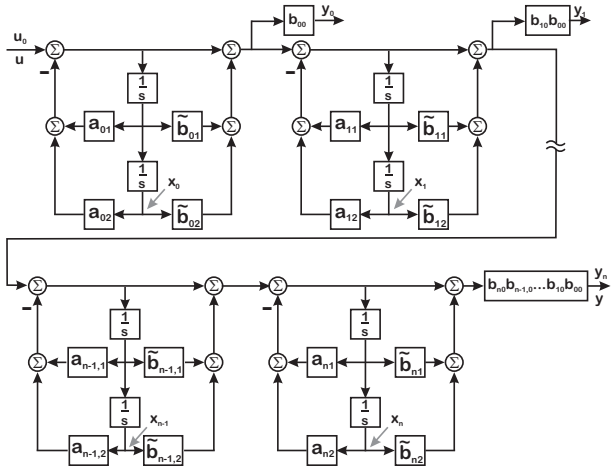

Fig. 8. The analog biquad cascade, with factored out $b_{i,0}$ terms and scaling the output of each block. This is completely analogous to the digital form of Figure 4.

If we have multiple biquads of the form shown in Equations 33, 34, and 35, we can chain these together by noting that:

$$\begin{aligned} u_i &= \tilde{y}_i, \quad \text{for } 0 \le i < n, \\ u_0 &= u, \quad \text{and} \\ \tilde{y}_n &= \tilde{y}. \end{aligned} \tag{36}$$

If one is willing to go through the algebraic pain and suffering of applying Equation 36 to each biquad structure

$$\begin{bmatrix} \ddot{\tilde{x}}_2 \\ \dot{\tilde{x}}_2 \\ \ddot{\tilde{x}}_1 \\ \dot{\tilde{x}}_1 \\ \ddot{\tilde{x}}_0 \\ \dot{\tilde{x}}_0 \end{bmatrix} = \begin{bmatrix} -a_{21} & -a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_{11} & -a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{01} & -a_{02} \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\tilde{x}}_2 \\ \tilde{x}_2 \\ \dot{\tilde{x}}_1 \\ \tilde{x}_1 \\ \dot{\tilde{x}}_0 \\ \tilde{x}_0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u \tag{37}$$

$$\begin{bmatrix} \tilde{y}_2 \\ \tilde{y}_1 \\ \tilde{y}_0 \end{bmatrix} = \begin{bmatrix} \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & \tilde{b}_{11}-a_{11} & \tilde{b}_{12}-a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 0 & 0 & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \end{bmatrix} \begin{bmatrix} \dot{\tilde{x}}_2 \\ \tilde{x}_2 \\ \dot{\tilde{x}}_1 \\ \tilde{x}_1 \\ \dot{\tilde{x}}_0 \\ \tilde{x}_0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u. \tag{38}$$

Fig. 9. State equations for continuous time Biquad State Space (BSS) with scalar output scaling.

a very regular state-space structure results. For a 3-biquad model (with no $b_{i0} = 0$), we get the state equation of 37. The unscaled output is in Equation 38, both displayed in Figure 9 due to their size. Finally, the properly scaled outputs are generated via:

$$\begin{bmatrix} y_2 \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} b_{20}b_{10}b_{00} & 0 & 0 \\ 0 & b_{10}b_{00} & 0 \\ 0 & 0 & b_{00} \end{bmatrix} \begin{bmatrix} \tilde{y}_2 \\ \tilde{y}_1 \\ \tilde{y}_0 \end{bmatrix}. \tag{39}$$

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.
- 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,

$$\tilde{b}_{i1} - a_{i1} = 2\varsigma_{in}\omega_{in} - 2\varsigma_{id}\omega_{id} \text{ and } \tilde{b}_{i2} - a_{i2} = \omega_{in}^2 - \omega_{id}^2. \tag{40}$$

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.

7. 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 (Abramovitch (2015e)) and the Δ coefficients of (Abramovitch (2015g)) work extremely well. On the other hand, representing a double integrator as a discrete biquad can be accomplished using a Trapezoidal Rule equivalent (Franklin et al. (1998b)) as described in (Abramovitch (2015c)).
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.

8. THE MATRICES, RELOADED

Generating coefficients from continuous time biquad parameters is discussed in some detail in (Abramovitch (2015e)) and (Abramovitch (2015g)). 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.

Both the state transition matrix (14) and the state matrix of (37) have very regular, block upper triangular forms. 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

$$\left[\tilde{b}_{i,1} - a_{i,1} \;\; \tilde{b}_{i,2} - a_{i,2} \right] \tag{41}$$

on the top row. The top rows of these blocks represent the feedthrough of the biquad states to the other states. This is true for both continuous and discrete-time forms, although the meanings of the coefficients differs. Likewise in the DT output matrix of (15) or the CT one of (38), these same subsections in (41) 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 (16) (or (39) for CT).

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 (41), 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. Assuming two zeros accompanying the two poles of the biquad and thinking of this from a root-locus perspective, one would see that such an arrangement would minimize the effect of the biquad's poles and zeros on the closed-loop poles that were not close to any of those dipole pairs.

9. DISCRETE TIME EXAMPLES

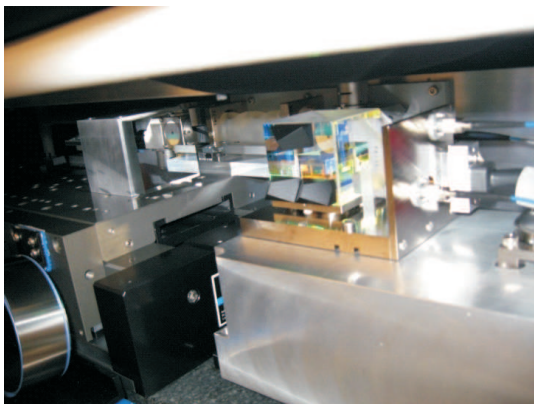

Fig. 10. Laboratory system: Aerotech air bearing linear stage, including linear grating for position measurement. The Aerotech system implements a PID controller and samples the data at 8 kHz. It has a built in swept-sine measurement. In the center of the image is a laser interferometer (IF) to provide 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 1. Model parameters from curve fit of Aerotech frequency response data.

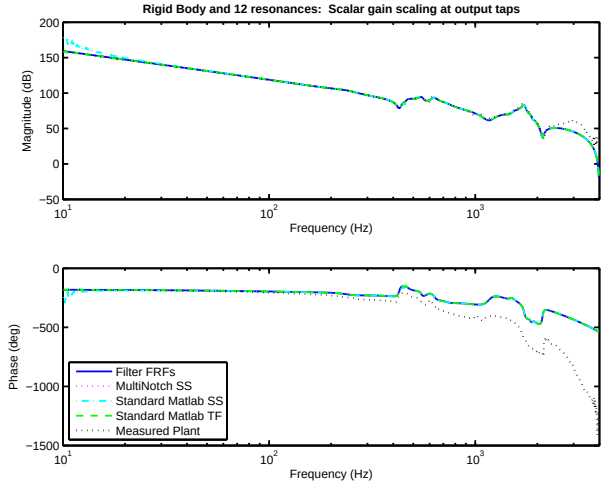

Fig. 11. Comparing state-space forms to AeroTech stage frequency response. Modeling the system with first 12 biquads and a rigid body, there is no discernible difference in the plots. The measured plant exhibits a phase roll off at high frequency not fit by the biquads.

In order to demonstrate the numerical improvements arising from the BSS structure, an example is take from measurements of an Aerotech linear stage used in experiments for the Quintessential Phase project (Johnstone and Abramovitch (2013)). The Aerotech single axis stage as shown in Figure 10. 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 (Johnstone and Abramovitch (2013)) 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

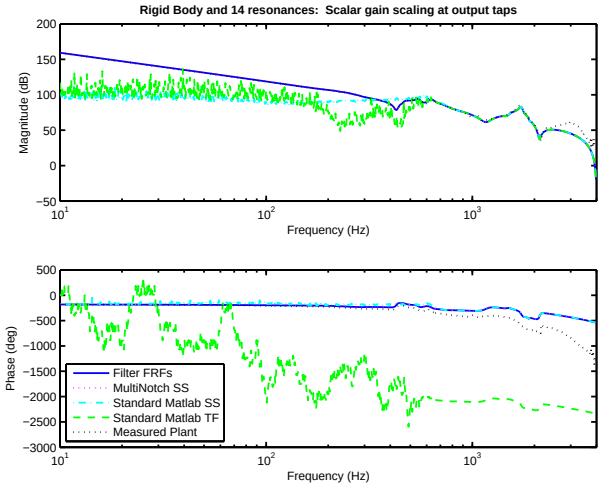

Fig. 12. Comparing state-space forms to Aerotech stage frequency response. Modeling the system with first 14 biquads and a rigid body, we start seeing significant differences in the different methods of realizing the state-space form. The conventional methods are clearly not matching the measured Aerotech frequency response, while the BSS method is.

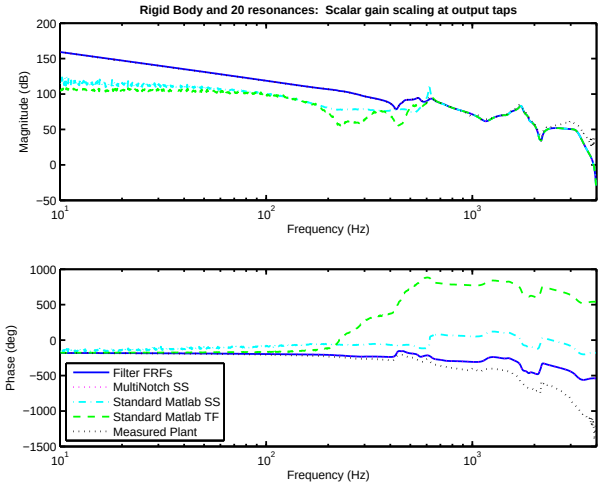

Fig. 13. Comparing state-space forms to Aerotech stage frequency response. Modeling the system with first 20 biquads and a rigid body, there is a massive difference between the conventional methods and the BSS method.

was constructed using Aerotech parameters. A FRF for this controller was synthesized on the same frequency axis as the stage open loop 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 1.

In order to compare the BSS 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 BSS 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 11, 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 modeled in the rigid body or the biquads.

However, just the addition of two more biquads in Figure 12, 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 13, 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 BSS continues to match the original measurement.

10. MATCHING CT AND DT RESPONSES

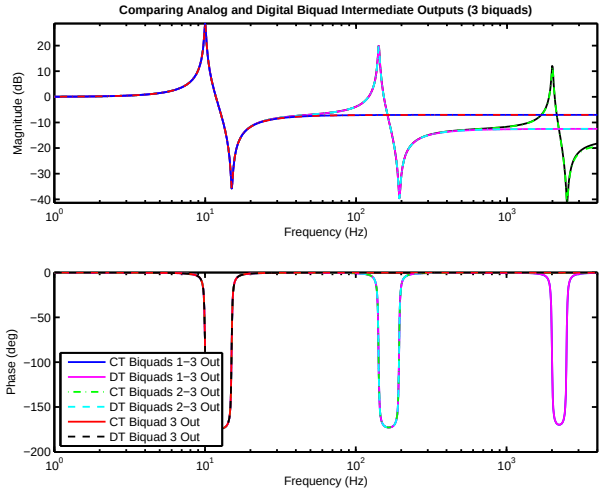

Fig. 14. Comparing analog and discrete BSS outputs. With 3 biquads, the outputs of each analog biquad section was plotted against it's digital version. This demonstrates the invariance of the biquad outputs under discretization.

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 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 14 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 BSS form, convert that to a discrete time BSS form for implementation, and then easily extract information about the continuous model from the discrete model results.

11. ADDING RIGID BODY DYNAMICS TO BSS: 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

$$D_T(z^{-1}) = K\left(\frac{T}{2}\right)^2\left(\frac{1+z^{-1}}{1-z^{-1}}\right)^2. \tag{42}$$

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

$$\tilde{D}_T(z^{-1}) = \left(\frac{1+z^{-1}}{1-z^{-1}}\right)^2 = \frac{1+2z^{-1}+z^{-2}}{1-2z^{-1}+z^{-2}} \tag{43}$$

from which we can extract the time domain equations

$$d_k - 2d_{k-1} + d_{k-2} = u_k. \tag{44}$$

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

$$x_{k+1} = 2x_k - x_{k-1} + u_k \tag{45}$$

and

$$\tilde{y}_{k+1} = x_{k+1} + 2x_k + x_{k-1}. \tag{46}$$

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

$$\begin{aligned}\tilde{y}_{k+1} &= x_{k+1} + 2x_k + x_{k-1}\\ &= 2x_k - x_{k-1} + u_k + 2x_k + x_{k-1}\\ &= 4x_k + u_k.\end{aligned} \tag{47}$$

We put this in state-space form as:

$$\begin{bmatrix} x_{k+1} \\ x_k \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k. \tag{48}$$

The output is defined as:

$$[\tilde{y}_{k+1}] = [\,4\;\;0\,]\begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} + [\,1\,]\,u_k. \tag{49}$$

Finally,

$$[y_{k+1}] = \left[KT^2/4\right][\tilde{y}_{k+1}]. \tag{50}$$

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

$$\begin{aligned}&a_1 = -2,\ a_2 = 1,\\ &\tilde{b}_1 = 2, \quad \tilde{b}_1 = 1, \quad \text{and } b_0 = \frac{KT^2}{4}.\end{aligned} \tag{51}$$

12. RIGID BODY MODES AND THE BILINEAR STATE SPACE (BLSS) STRUCTURE

The BSS handles most of the second-order portions of responses, but it has some limitations. In particular, state-space models or filters may require low pass or high pass sections. High pass filters (HPF) are typically modeled as proper transfer functions, but low pass filters (LPF) often exhibit a pole-zero excess in the continuous time (CT) form. This is a necessity for the frequency response to go to 0 at $s = \infty$.

Similarly, state-space models would be limited without being able to add in rigid body dynamics. Assuming that rigid body dynamics can be modeled as one or more second-order sections, we can focus on a basic second order, rigid body model and note the following characteristics:

1) The CT models usually have a pole-zero excess.
2) They are often not resonant and therefore are more likely to have distinct and real poles.
3) Unlike the LPF, there is usually some benefit to accessing the internal "physical" state of the rigid body model.

We may wish to have low pass and/or high pass filters in our Multinotch, or single or double integrators in our BSS model of a physical system. We might note that while it is theoretically possible to add integrators into a Multinotch, say as part of a controller that includes integral action, practical implementation of integrators usually involves useful nonlinearities, such as integrator anti-windup (Åström and Murray (2016)) which necessitated the integrators being broken off from the rest of the filter. However, in linear state-space models, single or double integrators are common in rigid body models. .

The common feature of most of these filters is the lack of direct feed through from the input to the output of any one filter stage, at least in the continuous time model. As noted briefly in (Abramovitch (2015d,b)) this lack of direct feedthrough affects the propagation of states and the structure of the state-space matrices.

13. CONTINUOUS-TIME, RIGID-BODY DYNAMICS AND LOW PASS FILTERS

The most common rigid body models we might see would be a double integrator as shown in the tortured biquad form of Fig. 15 with $a_1 = 0$ or an integrator plus low pass form $(a_1 > 0)$, where there is a real stable pole in place of one of the integrators. Spring-mass-damper actuators, such as those in an atomic force microscope (AFM) (Schitter et al. (2006); Abramovitch et al. (2007)) would require a more difficult access to velocity. The double integrator is modeled as:

$$D(s) = \frac{K}{s^2}, \tag{52}$$

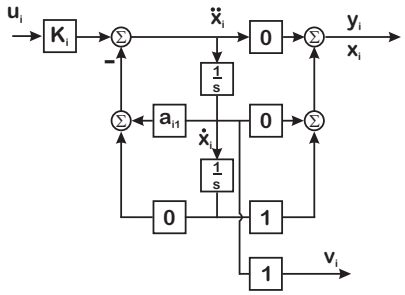

Fig. 15. Continuous time (CT) rigid body biquad. Setting $a_1 = 0$ turns it into a CT double integrator..

while the single pole rigid body is modeled as

$$D(s) = \frac{Ka_1}{s(s+a_1)}. \tag{53}$$

where a_1 might be viscous friction applying velocity feedback.

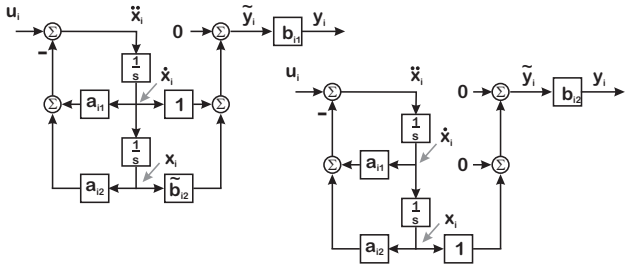

Fig. 16. Analog biquads without direct feedthrough. On the left, $b_{i0} = 0$. On the right, both b_{i0} and $b_{i1} = 0$. In either case, the leading gain is the gain of the highest order numerator term that has a non-zero coefficient. In both, the lack of direct feedthrough means that the output is 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.

Let's consider a few forms of continuous time low pass filters (CT-LPF). When possible, we will set the DC gain to 1 as a common scaling. A pair of first order models are presented in:

$$L_{1,a}(s) = \frac{a}{s+a} \text{ and} \tag{54}$$

$$L_{1,b}(s) = \left(\frac{a}{b}\right)\frac{s+b}{s+a}. \tag{55}$$

In the case of (54), it is low pass because the "zero" is at infinite frequency. At some point, for positive a, it has to roll off. Equation (55) is only a low pass filter if $0 \leq a < b$. It doesn't have infinite rejection at infinite frequency. It is a lag filter, where the response at low frequency is higher than the response at high frequency, and the level of attenuation is set by the distance between a and b. Our method of translating filters from continuous time to discrete time in the multi-notch is based on pole-zero mapping, and this has worked fine as long as the zeros were finite, so there is no problem with (55). Likewise, there would be no problem with:

$$L_{2,b2}(s) = \left(\frac{a_2}{b_2}\right)\frac{s^2+b_1 s+b_2}{s^2+a_1 s+a_2}. \tag{56}$$

Equation (54) is a different matter as are:

$$L_{2,b0}(s) = \left(\frac{a_2}{b_2}\right)\frac{b_2}{s^2+a_1 s+a_2}, \text{ and} \tag{57}$$

$$L_{2,b1}(s) = \left(\frac{a_2}{b_1}\right)\frac{s+b_1}{s^2+a_1 s+a_2}. \tag{58}$$

Equations (54), (57), and (58) all have zeros when $|s| \longrightarrow \infty$ or when s is evaluated on the $j\omega$ axis, when $|\omega| \longrightarrow \infty$. A general form for such structures is diagrammed in Fig. 16 and will be discussed in Section 14.

14. 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 (Abramovitch (2015c)) 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.

Furthermore, our focus on LPF and rigid body models raised the importance of entering models with no direct feedthrough into the BSS and MNF. 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, as described in Section 16. 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, we need to know how to accommodate this.

Figure 16 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 (33) is unchanged. However, the state output equations change a lot. In the left hand case, Equation 34 becomes

$$[\,\tilde{y}_i\,] = \begin{bmatrix}1 & \tilde{b}_{i2}\end{bmatrix}\begin{bmatrix}\dot{x}_i \\ x_i\end{bmatrix} + [\,0\,]\,u_i \tag{59}$$

where $\tilde{b}_{i2} = b_{i2}/b_{i1}$ and (35) becomes:

$$[\,y_i\,] = [\,b_{i1}\,][\,\tilde{y}_i\,]. \tag{60}$$

In the right hand case, Equation 34 becomes

$$[\,\tilde{y}_i\,] = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}_i \\ x_i\end{bmatrix} + [\,0\,]\,u_i \tag{61}$$

and (35) becomes:

$$[\, y_i \,] = [\, b_{i2} \,][\, \tilde{y}_i \,]. \tag{62}$$

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 63. The unscaled output is in Equation 64, both displayed in Figure 17 due to their size. Finally, the properly scaled outputs are generated via:

$$\begin{bmatrix} y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} b_{30}b_{20}b_{1x}b_{00} & 0 & 0 & 0 \\ 0 & b_{20}b_{1x}b_{00} & 0 & 0 \\ 0 & 0 & b_{12}b_{00} & 0 \\ 0 & 0 & 0 & b_{00} \end{bmatrix} \begin{bmatrix} \tilde{y}_3 \\ \tilde{y}_2 \\ \tilde{y}_1 \\ \tilde{y}_0 \end{bmatrix}, \tag{65}$$

where $b_{ix} = b_{i1}$ if $b_{i1} \neq 0$ and $b_{ix} = b_{i2}$ if $b_{i1} = 0$. In Equations 63 and 64 $\tilde{b}_{11} = 1$ 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.

While lack of direct feedthrough may be unavoidable in continuous time, the real design choice comes when we wish to discretize continuous time models without direct feedthrough and represent them in an equivalent discrete time BSS or MNF. The key is preserving the structure so that we can automate the addition of extra blocks while maintaining readability and explainability.

15. BILINEAR STATE SPACE (BLSS) FORM

One of the issues with biquads is that while the continuous time biquads map to the discrete time biquads, and the input-output relationships hold, the internal states might not represent the physical states. With oscillatory (complex roots) numerators and denominators, this may not be overly important. However, rigid-body portions of the model often have real roots. For these there is often an advantage to accessing the individual physical states and in having these states map to discrete time models. To do this, we suggest the Bilinear State Space form (BLSS), which opens up the biquad in the case of real and distinct poles and zeros. The familiar CT and DT state equations are:

$$\dot{x} = F_C x + G_C u,\ y = H_C x + D_C u \text{ and} \tag{66}$$

$$x(k+1) = Fx(k) + Gu(k),\ y = Hx(k) + Du(k), \tag{67}$$

respectively. The continuous and discrete matrices have the same structure, but the interpretation of the internal $\{a_{i1}, a_{i2}, \tilde{b}_{i0}, \tilde{b}_{i1}, \text{ and } \tilde{b}_{i2}\}$ coefficients change in going from continuous to discrete time. The DT matrices, $\{F, G, H, \text{ and} D\}$ are given by:

$$F = \begin{bmatrix} -a_{i+1,1} & b_{i,0}(\tilde{b}_{i1} - a_{i1}) \\ 0 & -a_{i,1} \end{bmatrix}, \tag{68}$$

$$H = \begin{bmatrix} b_{i+1,0}(\tilde{b}_{i+1,1} - a_{i+1,1}) & b_{i+1,0}b_{i,0}(\tilde{b}_{i,1} - a_{i,1}) \\ 0 & b_{i,0}(\tilde{b}_{i,1} - a_{i,1}) \end{bmatrix}, \tag{69}$$

$$G = \begin{bmatrix} b_{i,0} \\ 0 \end{bmatrix}, \text{ and } D = \begin{bmatrix} b_{i+1,0}b_{i,0} \\ b_{i,0} \end{bmatrix}, \text{ where} \tag{70}$$

$$u_i = y_{i-1} \text{ for } i = 1, \ldots, n. \tag{71}$$

The CT matrices, $\{F_C, G_C, H_C, \text{ and} D_C\}$ are given by:

$$F_C = \begin{bmatrix} -a_{i+1,1} & b_{i,0}(\tilde{b}_{i1} - a_{i1}) \\ 0 & -a_{i,1} \end{bmatrix} \tag{72}$$

$$H_C = \begin{bmatrix} b_{i+1,0}(\tilde{b}_{i+1,1} - a_{i+1,1}) & b_{i+1,0}b_{i,0}(\tilde{b}_{i,1} - a_{i,1}) \\ 0 & b_{i,0}(\tilde{b}_{i,1} - a_{i,1}) \end{bmatrix} \tag{73}$$

$$G_C = \begin{bmatrix} b_{i,0} \\ 0 \end{bmatrix}, \text{ and } D_C = \begin{bmatrix} b_{i+1,0}b_{i,0} \\ b_{i,0} \end{bmatrix}, \text{ where} \tag{74}$$

$$u_i = y_{i-1} \text{ for } i = 1, \ldots, n. \tag{75}$$

The general continuous time BLSS model is diagrammed in Fig. 18 which simplifies to Fig.19 for the rigid body models we have discussed. The general discrete time model is diagrammed in Fig. 20. We will see that these forms are extremely useful with adding a rigid body section to BSS models, since we can access the states such as velocity and position easily.

16. DISCRETIZATION CHOICES

With the MNF, and the BSS, discretization was easily done via pole-zero mapping so long as the continuous time numerator and denominator were of the same order (Abramovitch (2015f,h,d,b)). The Bode plot comparisons in (Abramovitch (2015b)) gave confidence that this captured the zero behavior. When we are dealing with a pole-zero excess of 1 or 2 in a block – such as we have with CT low pass filters and CT rigid body models, we need to consider some choices for placing the CT "zeros at ∞" (Franklin et al. (1998a)). We have essentially 4 choices:

- Map one or two CT zeros at $s = -\infty$ to $z = -\infty$. While this takes sampling delay into account, it is the least favored of these methods for any filter that will be used in a feedback mechanism as the zeros at $-\infty$ will pull a corresponding number of closed-loop poles towards them and out of the unit circle.
- Map the CT zeros at $s = -\infty$ to $z = 0$. These tend to cancel pure delays and are conservative in that they minimize the phase effect of the denominator. This is what is done in typical PID discretization (Abramovitch (2015j)) and the conservatism of the zero at $z = 0$ helps stabilize the overall loop. The caveat here is that in the PID usage, the backwards rule is part of the construction of the controller implementation, not trying to accurately model the physical system. In the latter case, the backwards rule equivalents are not the most accurate match to the continuous0time filter model.
- Map the CT zeros at $s = -\infty$ to $z = -1$. This corresponds to the trapezoidal rule equivalent and is the most accurate Bode plot magnitude match for the continuous-time model.

$$\begin{bmatrix} \dddot{\tilde{x}}_3 \\ \ddot{\tilde{x}}_3 \\ \dddot{\tilde{x}}_2 \\ \ddot{\tilde{x}}_2 \\ \dddot{\tilde{x}}_1 \\ \ddot{\tilde{x}}_1 \\ \dddot{\tilde{x}}_0 \\ \ddot{\tilde{x}}_0 \end{bmatrix} = \begin{bmatrix} -a_{31} & -a_{32} & \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_{21} & -a_{22} & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{11} & -a_{12} & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -a_{01} & -a_{02} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\tilde{x}}_3 \\ \dot{\tilde{x}}_3 \\ \ddot{\tilde{x}}_2 \\ \dot{\tilde{x}}_2 \\ \ddot{\tilde{x}}_1 \\ \dot{\tilde{x}}_1 \\ \ddot{\tilde{x}}_0 \\ \dot{\tilde{x}}_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u \tag{63}$$

$$\begin{bmatrix} \tilde{y}_3 \\ \tilde{y}_2 \\ \tilde{y}_1 \\ \tilde{y}_0 \end{bmatrix} = \begin{bmatrix} \tilde{b}_{31}-a_{31} & \tilde{b}_{32}-a_{32} & \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 0 & 0 & \tilde{b}_{21}-a_{21} & \tilde{b}_{22}-a_{22} & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & \tilde{b}_{11} & \tilde{b}_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tilde{b}_{01}-a_{01} & \tilde{b}_{02}-a_{02} \end{bmatrix} \begin{bmatrix} \dddot{\tilde{x}}_3 \\ \ddot{\tilde{x}}_3 \\ \dddot{\tilde{x}}_2 \\ \ddot{\tilde{x}}_2 \\ \dddot{\tilde{x}}_1 \\ \ddot{\tilde{x}}_1 \\ \dddot{\tilde{x}}_0 \\ \ddot{\tilde{x}}_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u. \tag{64}$$

Fig. 17. State equations for continuous time BSS with scalar output scaling. Biquad 1 lacks direct feedthrough.

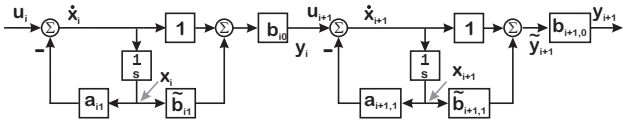

Fig. 18. Continuous time Bilinear State Space (CT-BLSS) form.

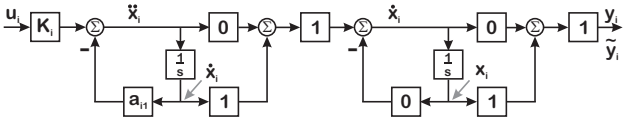

Fig. 19. Continuous time rigid body BLSS model. Note that rather than indexing the second stage as $i+1$, we stick with i but label the level of integration on the signals.

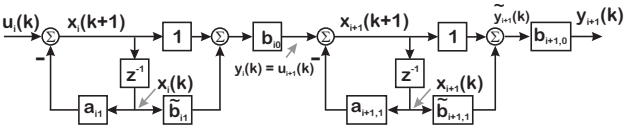

Fig. 20. Discrete time Bilinear State Space form (DT-BLSS).

- Do some combination of the above choices.

While assigning one of the CT zeros at $s=-\infty$ to $z=-\infty$ is a traditional way of incorporating delay (Franklin et al. (1998a)) and results in no direct feedthrough. Assigning both DT zeros this way results in a discrete biquad that looks like

$$B_{i,FR}(z) = \frac{b_{i,2}z^{-2}}{1+a_{i,1}z^{-1}+a_{i,2}z^{-2}}, \tag{76}$$

which is similar to a forward rectangular rule equivalent, $\frac{1}{s} \longrightarrow \frac{Tz^{-1}}{1-z^{-1}}$. This will not have direct feedthrough and thus will require a discrete time block structure similar to the continuous time ones shown in Section 14. Recent work suggests designing for a system with minimum delay and then backing off bandwidth to accommodate the phase due to measured delay (Butterworth et al. (2011); Abramovitch (2015i),Abramovitch:23,AbramovitchA:23).

If we consider the conservative backwards rule construction, $\frac{1}{s} \longrightarrow \frac{T}{1-z^{-1}}$, we end up with

$$B_{i,FR}(z) = \frac{b_{i,2}z^{-2}}{1+a_{i,1}z^{-1}+a_{i,2}z^{-2}}, \tag{77}$$

where we see that the numerator delay from (76) has been completely eliminated. The BSS block will have a standard structure, just with $\tilde{b}_{i,1}$ and $\tilde{b}_{i,2}=0$. Finally, if we choose the trapezoidal rule equivalent, $\frac{1}{s} \longrightarrow \frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}$, we end up with

$$B_{i,TR}(z) = b_{i,0}\frac{1+2z^{-1}+z^{-2}}{1+a_{i,1}z^{-1}+a_{i,2}z^{-2}}, \tag{78}$$

with two zeros at $z=-1$. Finally, we might try tweaking the phase of the model by choosing one zero at $z=0$ and one at $z=-1$, in which case we would have

$$B_{i,TBR}(z) = b_{i,0}\frac{1+z^{-1}}{1+a_{i,1}z^{-1}+a_{i,2}z^{-2}}. \tag{79}$$

These are standard BSS blocks with direct feedthrough, but with particular values for $\tilde{b}_{i,1}$ and $\tilde{b}_{i,2}$. If we want a rigid body model with the BSS or a low pass filter that works well with the BSS or MNF, it is best to avoid zeros at $-\infty$.

17. DISCRETE-TIME, RIGID-BODY MODELS

In Section 11 (Abramovitch (2015d)), we showed how using a trapezoidal rule equivalent on a double integrator preserved the feedthrough. With discrete equivalent forms of the model in Fig. 15, we run into the issue that we cannot readily access a reasonable velocity estimate from these models. Looking at the ZOH equivalent (Franklin et al. (1998a)) model in Fig. 21 or the trapezoidal rule equivalent in Fig. 23, we can easily extract an acceleration estimate and/or a position estimate, but velocity would require some new combination of the states. In

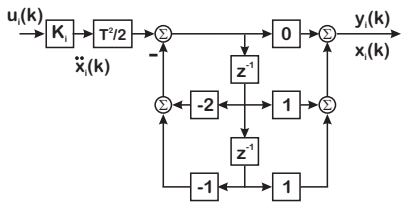

Fig. 21. Discrete double integrator biquad model (ZOH equivalent). Notice when viewed this way, the integrators are unbalanced in that the first one is implemented differently from the second.

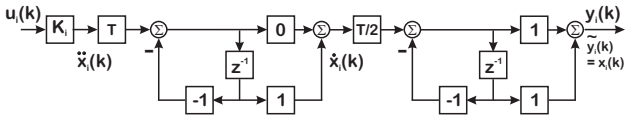

Fig. 22. Discrete double integrator BLSS model (ZOH equivalent). In this drawing, we've chosen to make the index, i, as with a biquad stage, but we are explicitly labeling the different integration levels.

practical use of state-space models for motion control of mechatronic systems, it seems highly illogical to go to the trouble of generating a state-space model and not be able to easily access velocity. Furthermore, the ZOH equivalent is not applied to just the double integrator, but to the entire plant model. This confounds our ability to match discrete and analog states in any way. In the companion tutorial (Abramovitch (2023)) we will discuss when the extra numerical accuracy of the ZOH can be forgone in favor of the ability to understand and debug our discrete-time, state-space models.

A word about indexing here. While normally we would want to index these blocks as we do with any of our biquad blocks, say block i, all the subscripts can make the text Byzantine at first glance. Instead, we use index 0 for the first integrator, and 1 for the second integrator, realizing that the readers will be able to add the appropriate offset to the equations. For clarity, the drawings index the stages all as i, but noting the integration level of each block.

The BLSS model of Fig. 19 exposes velocity. Discretizing this model with a ZOH equivalent leads to the model of Fig. 22, while a trapezoidal rule equivalent can be found in Fig. 24.

We break up the ZOH equivalent as

$$\begin{aligned} D_{ZOH}(z) &= K\frac{T^2}{2}\frac{(z+1)}{(z-1)^2} \\ &= KT\left(\frac{1}{z-1}\right)\left(\frac{T}{2}\right)\left(\frac{z+1}{z-1}\right). \end{aligned} \tag{80}$$

The blocks end up with the equations of:

$$x_{0,k+1} = KTu_{0,k} + x_{0,k} \text{ and} \tag{81}$$

$$y_{0,k} = x_{0,k}, \tag{82}$$

where $u_0 = u$ and $y_1 = y$. With $u_{1,k} = y_{0,k}$ we have:

$$x_{1,k+1} = \frac{T}{2}u_{1,k} + x_{1,k} = x_{1,k} + \frac{T}{2}x_{0,k} \text{ and} \tag{83}$$

$$y_{1,k} = x_{1,k} + x_{1,k+1} = 2x_{1,k} + \frac{T}{2}x_{0,k}. \tag{84}$$

Together, these become:

$$\begin{bmatrix} x_{1,k+1} \\ x_{0,k+1} \end{bmatrix} = \begin{bmatrix} 1 & \frac{T}{2} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} 0 \\ KT \end{bmatrix} u_k. \tag{85}$$

The output is defined as:

$$\begin{bmatrix} y_{1,k} \\ y_{0,k} \end{bmatrix} = \begin{bmatrix} 2 & \frac{T}{2} \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u_k. \tag{86}$$

This is the textbook model for a double integrator (Franklin et al. (1998a)), but now we can access the velocity output, $y_{0,k}$ directly. Of course, for a true ZOH-equivalent, the entire model would have been discretized at once, rather than simply the double integrator.

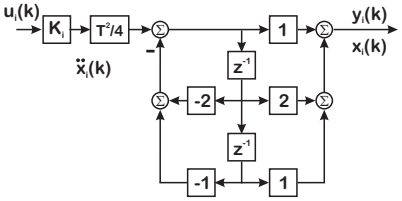

Fig. 23. Discrete double integrator biquad model (trapezoidal rule equivalent).

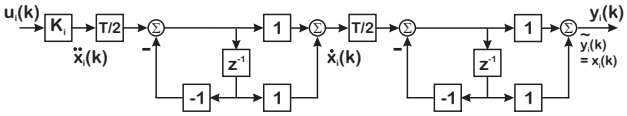

Fig. 24. Discrete double integrator BLSS model (trapezoidal rule equivalent). In this drawing, we've chosen to make the index, i, as with a biquad stage, but we are explicitly labeling the different integration levels.

This does not have direct feedthrough, unlike the trapezoidal rule model of (87). We break that up as follows:

$$\begin{aligned} D_{TR}(z) &= K\frac{T^2}{4}\frac{(z+1)^2}{(z-1)^2} \\ &= K\left(\frac{T}{2}\right)\left(\frac{z+1}{z-1}\right)\left(\frac{T}{2}\right)\left(\frac{z+1}{z-1}\right). \end{aligned} \tag{87}$$

The blocks end up with the equations of:

$$x_{0,k+1} = K\frac{T}{2}u_{0,k} + x_{0,k} \text{ and} \tag{88}$$

$$y_{0,k} = x_{0,k} + x_{0,k+1} = 2x_{0,k} + K\frac{T}{2}u_{0,k}, \tag{89}$$

where $u_0 = u$ and $y_1 = y$. With $u_{1,k} = y_{0,k}$ we have:

$$\begin{aligned} x_{1,k+1} &= \frac{T}{2}u_{1,k} + x_{1,k} \\ &= 2x_{1,k} + Tx_{0,k} + K\frac{T^2}{4}u_k \text{ and} \end{aligned} \tag{90}$$

$$\begin{aligned} y_{1,k} &= x_{1,k} + x_{1,k+1} = 2x_{1,k} + K\frac{T}{2}u_{1,k} \\ &= 2x_{1,k} + Tx_{0,k} + K\frac{T^2}{4}u_k. \end{aligned} \tag{91}$$

Together, these become:

$$\begin{bmatrix} x_{1,k+1} \\ x_{0,k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} K\left(\frac{T}{2}\right)^2 \\ K\left(\frac{T}{2}\right) \end{bmatrix} u_k. \tag{92}$$

The output is defined as:

$$\begin{bmatrix} y_{1,k} \\ y_{0,k} \end{bmatrix} = \begin{bmatrix} 2 & T \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} K\left(\frac{T}{2}\right)^2 \\ K\left(\frac{T}{2}\right) \end{bmatrix} u_k. \qquad (93)$$

With this implementation of the trapezoidal rule equivalent, we can also access the velocity output, $y_{0,k}$ directly. This has direct feedthrough, and is probably the closest simple equivalent to to continuous time form from a frequency response perspective.

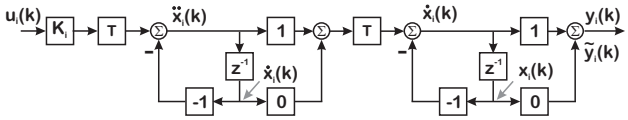

Fig. 25. Discrete double integrator BLSS model (backwards rectangular rule equivalent).

Note the key difference internally is that we have scaled the integration of the intermediate state structure to more closely match the continuous time form. This kind of scaling was scrupulously avoided for high Q filters in (Abramovitch (2015f,b)), but should present no problem with the rigid body modes.

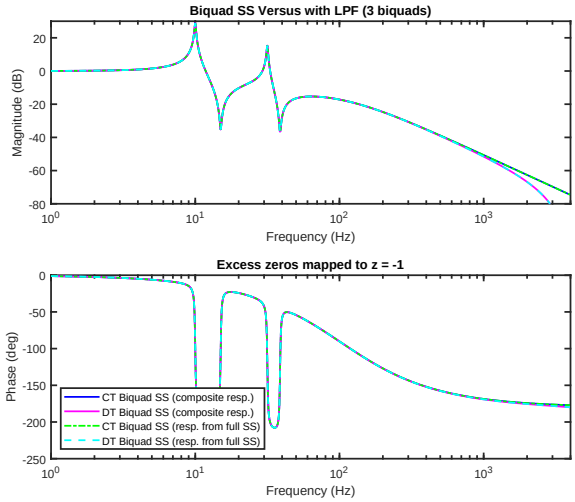

Fig. 26. BSS with three biquads including a low pass filter. The CT biquad plots include a composite of the individual CT biquad Bode plots (blue) and a Bode of the complete CT BSS structure (green). The DT biquad plots also include a composite of individual DT biquad Bode plots (magenta) and a Bode of the complete DT BSS structure (cyan). The match of the complete structures to the composites show that the CT and DT BSS structures have properly represented the series connection of the individual biquads. The match between CT and DT show that we have properly discretized the low pass filter in our BSS. The phase match of the DT curves with the CT curves is based on the zeros at $z = -1$ due to the use of a Trapezoidal rule equivalent discretization.

Fig. 26 demonstrates the Bode plot of a BSS model with a low pass filter (LPF) as part of a 3-biquad model.

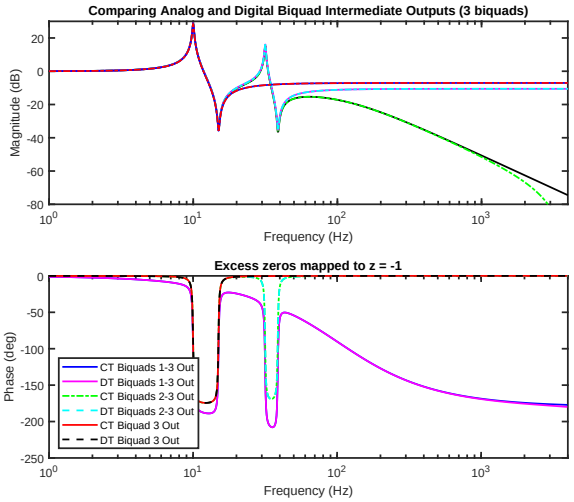

Fig. 27. BSS with three biquads including a low pass filter in Biquad 1. This plot compares the Bode responses of the individual CT and DT biquad sections. The outputs of biquad 3 and biquad 2 show the magnitude and phase flattening out at high frequency (due to the matched number of poles and zeros). Once the response of biquad 1 is added in, we see the low pass rolloff of Figure 26. At each biquad output, the match between continuous and discrete responses is incredibly close, a unique and useful feature of this

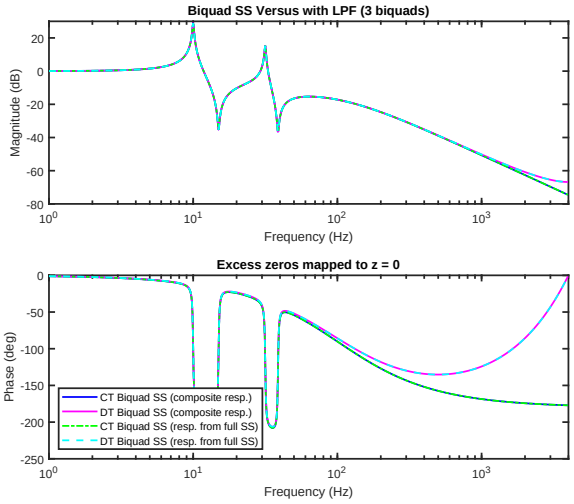

Fig. 28. BSS with three biquads including a low pass filter. The CT biquad plots include a composite of the individual CT biquad Bode plots (blue) and a Bode of the complete CT BSS structure (green). The DT biquad plots also include a composite of individual DT biquad Bode plots (magenta) and a Bode of the complete DT BSS structure (cyan). The match of the complete structures to the composites show that the CT and DT BSS structures have properly represented the series connection of the individual biquads. The match between CT and DT show that we have properly discretized the low pass filter in our BSS. The curl up of the phase back to 0 in the DT curves is based on the zeros at $z = 0$ due to the use of a backwards rectangular rule equivalent discretization.

Note the close match between the composite responses (individual biquad frequency responses combined) versus responses extracted directly from the full BSS structure.

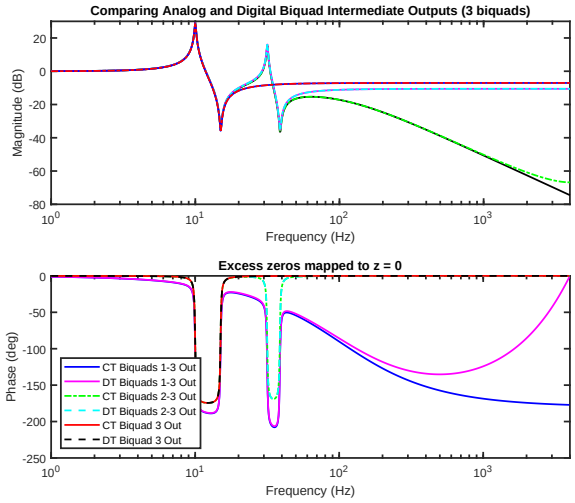

Fig. 29. BSS with three biquads including a low pass filter in Biquad 1. This plot compares the Bode responses of the individual CT and DT biquad sections. The outputs of biquad 3 and biquad 2 show the magnitude and phase flattening out at high frequency (due to the matched number of poles and zeros). Once the response of biquad 1 is added in, we see the low pass rolloff of Figure 28. At each biquad output, the match between continuous and discrete responses is incredibly close, a unique and useful feature of this structure.

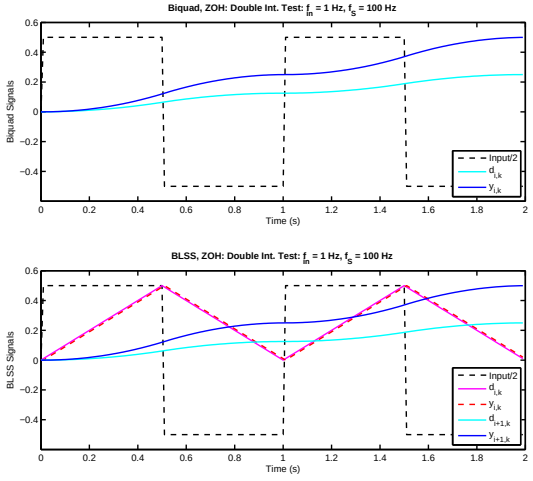

Fig. 30. Double integrator with square wave input. Implemented using a ZOH equivalent biquad (top) and BLSS (bottom).

The significance of this is that while the discrete time LPF can be modeled with direct feedthrough, the continuous time LPF cannot. Nevertheless, they produce responses that match very well. Fig. 27 shows that the CT and DT match is across all the biquad outputs, as previously demonstrated without the LPF in (Abramovitch (2015b)). Note the close phase match to the continuous-time responses provided by choosing $z = -1$ for the location of the extra zeros, corresponding to the trapezoidal rule equivalent. This contrasts with the phase curl back to $0°$ when the excess zeros are placed at $z = 0$, which correlates with a backwards rectangular rule equivalent.

Fig. 30 shows at double integrator, discretized using a ZOH equivalent, implemented as a biquad (top) and a BLSS

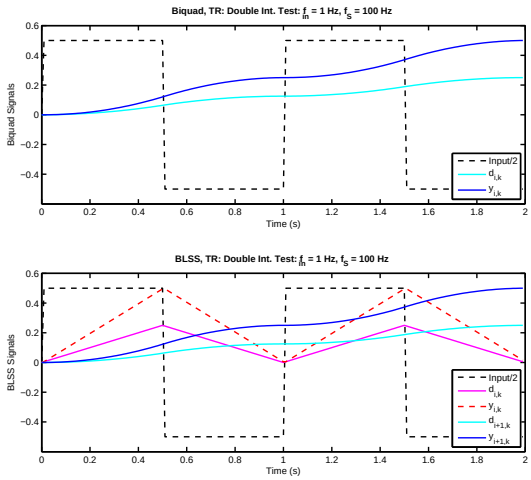

Fig. 31. Double integrator with square wave input. Implemented using a trapezoidal rule equivalent biquad (top) and BLSS (bottom).

(bottom). The input-output behavior is consistent, but we now have access to the internal intermediate state with the BLSS. Fig.31, repeats the simulation using a Trapezoidal rule equivalent. In both cases, the BLSS gives us access to the output of the first stage, which is can be interpreted as velocity. We see further, that using the trapezoidal rule equivalent adjusts the scale of the internal state of the first bilinear block to make the integrators balanced. The BLSS block is a logical addition to state-space models needing access to both position and velocity.

19. SUMMARY

This tutorial paper has introduced the reader to the BSS and BLSS structures, the relationship between continuous-time and discrete-time states, their numerical properties, how to combine them, and how they might be useful for implementing state-space methods reliably on lightly-damped mechatronic structures. The use of the divide and conquer discretization that results in so much connection between continuous and discrete states is in contrast with the "discretize entire model all at once" method that is inherent in the zero-order hold (ZOH) equivalent. This tradeoff will be discussed in a companion paper (Abramovitch (2023)).

REFERENCES

Abramovitch, D.Y., Andersson, S.B., Pao, L.Y., and Schitter, G. (2007). A tutorial on the mechanisms, dynamics, and control of atomic force microscopes. In *Proc. Amer. Ctrl. Conf.* New York, NY.

Abramovitch, D.Y. (2015a). The continuous time biquad state space structure. In *Proceedings of the 2015 American Control Conference*, 4168–4173. AACC, IEEE, Chicago, IL.

Abramovitch, D.Y. (2015b). The continuous time biquad state space structure. In *Proc. Amer. Ctrl. Conf.* IEEE, Chicago.

Abramovitch, D.Y. (2015c). The discrete time biquad state space structure: Low latency with high numerical fidelity. In *Proceedings of the 2015 American Control Conference*, 2813–2818. AACC, IEEE, Chicago, IL.

Abramovitch, D.Y. (2015d). The discrete time biquad state space structure: Low latency with high numerical fidelity. In *Proc. Amer. Ctrl. Conf.* IEEE, Chicago.

Abramovitch, D.Y. (2015e). The Multinotch, Part I: A low latency, high numerical fidelity filter for mechatronic control systems. In *Proceedings of the 2015 American Control Conference*, 2161–2166. AACC, IEEE, Chicago, IL.

Abramovitch, D.Y. (2015f). The Multinotch, Part I: A low latency, high numerical fidelity filter for mechatronic control systems. In *Proc. Amer. Ctrl. Conf.* IEEE, Chicago.

Abramovitch, D.Y. (2015g). The Multinotch, Part II: Extra precision via Δ coefficients. In *Proceedings of the 2015 American Control Conference*, 4137–4142. AACC, IEEE, Chicago, IL.

Abramovitch, D.Y. (2015h). The Multinotch, Part II: Extra precision via Δ coefficients. In *Proc. Amer. Ctrl. Conf.* IEEE, Chicago.

Abramovitch, D.Y. (2015i). Trying to keep it real: 25 years of trying to get the stuff I learned in grad school to work on mechatronic systems. In *Proc. Multi-Conf. Sys. & Ctrl.* IEEE, Sydney.

Abramovitch, D.Y. (2015j). A unified framework for analog and digital PID controllers. In *Proc. Multi-Conf. Sys. & Ctrl.* IEEE, Sydney.

Abramovitch, D.Y. (2018). Adding rigid body modes and low-pass filters to the biquad state space and multinotch. In *Proceedings of the 2018 American Control Conference*, 6024–6030. AACC, IEEE, Milwaukee, WI.

Abramovitch, D.Y. (2023). A discussion on discretization and practical tradeoffs of the ZOH equivalent. In *2023 Modeling, Estimation, and Control Conference*. AACC, IFAC, Lake Tahoe, NV.

Abramovitch, D.Y., Andersson, S.B., Leang, K.K., Nagel, W.S., and Ruben, S.D. (2023). A tutorial on real-time computing issues for control systems. In *Proceedings of the 2023 American Control Conference*, 3751–3768. AACC, IEEE, San Diego, CA.

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

Abramovitch, D.Y. and Johnstone, E. (2013). 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.

Abramovitch, D.Y. and Moon, C.R. (2012). Cascaded digital filters with reduced latency. International Application Published Under the Patent Cooperation Treaty WO 2012/118483, World Intellectual Property Organization.

Åström, K.J. and Wittenmark, B. (1990). *Computer Controlled Systems, Theory and Design*. Prentice Hall, Englewood Cliffs, N.J. 07632, second edition.

Åström, K.J. and Murray, R.M. (2016). *Feedback Systems*. Princeton Univ. Press, 2nd edition.

Butterworth, J.A., Pao, L.Y., and Abramovitch, D.Y. (2011). Fitting discrete-time models to frequency responses for systems with transport delay. In *ASME Int. Mechanical Engr. Congress & Exposition*. ASME.

Franklin, G.F., Powell, J.D., and Emami-Naeini, A. (2006). *Feedback Control of Dynamic Systems*. Prent. Hall, 5th edition.

Franklin, G.F., Powell, J.D., and Workman, M.L. (1998a). *Digital Control of Dynamic Systems*. Add. Wesl. Long., 3rd edition.

Franklin, G.F., Powell, J.D., and Workman, M.L. (1998b). *Digital Control of Dynamic Systems*. Addison Wesley Longman, Menlo Park, California, third edition.

Johnstone, E. and Abramovitch, D.Y. (2013). Quintessential Phase: A method of mitigating turbulence effects in interferometer measurements of precision motion. In *Proceedings of the 2013 American Control Conference*. AACC, IEEE, Washington, DC.

Kailath, T. (1980a). *Linear Systems*. Prentice-Hall, Englewood Cliffs, N.J. 07632.

Kailath, T. (1980b). *Linear Systems*. Prentice-Hall.

Oppenheim, A.V. and Schafer, R.W. (1975). *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, N. J.

Schitter, G., Fantner, G.E., Thurner, P., Adams, J., and Hansma, P.K. (2006). Design and characterization of a novel scanner for high-speed atomic force microscopy. In *Proc. 4th IFAC-Symp. Mech. Sys.*

Texas Instruments (1993). *TMS320C4x User's Guide*. Texas Instruments.