2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Adding Rigid Body Modes and low pass Filters to the Biquad State Space and Multinotch

Daniel Y. Abramovitch*

***Abstract*— This paper presents several useful extensions to the Multinotch Filter (MNF) [1], [2] and the Biquad State Space (BSS) structure [3], [4] that are useful for adding low pass filters (LPF) and rigid body models to those structures. The paper is in the form of a programmatic toolkit, which allows these structures to be including without distorting the original benefits of the MNF or the BSS. As such, they make those structures applicable to a larger set of dynamic models.**

I. Introduction

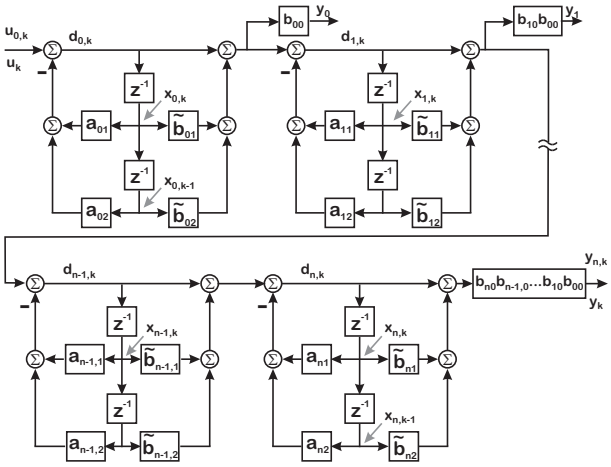

Fig. 1. Discrete biquad cascade, with factored out $b_{i,0}$ terms and scaling the output of each block.

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,

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

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

$$F(z^{-1}) = \frac{b_0 + b_1 z^{-1} + \cdots + b_{n-1}z^{-n+1} + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_{n-1}z^{-n+1} + a_n z^{-n}}, \quad (2)$$

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

$$\begin{aligned} yk) \quad = \quad & -a_1 y(k-1) + \cdots - a_{n-1}y(k-n+1) \\ & -a_n y(k-n) + b_0 u(k) + b_1 u(k-1) + \\ & \cdots + b_{n-1}u(k-n+1) + b_n u(k-n). \end{aligned} \quad (3)$$

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

The issues with these filters (or canonical state-space forms [5]) 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 [2].

The Multinotch Filter (MNF) [1], [2] was created to address these issues by breaking up polynomial form filters into a serial cascade of biquads (Fig. 1), while maintaining the ability to do precalculation for minimum latency control. The individual biquads are parameterized as:

$$\frac{Y_i(z^{-1})}{U_i(z^{-1})} = B_i(z^{-1}) = b_{i,0}\left(\frac{1 + \tilde{b}_{i,1}z^{-1} + \tilde{b}_{i,2}z^{-2}}{1 + a_{i,1}z^{-1} + a_{i,2}z^{-2}}\right), \text{ with} \quad (4)$$

$$\begin{aligned} U_{i+1}(z^{-1}) &= \tilde{Y}_i(z^{-1}), \quad \text{for } 0 \le i < n, \\ U_0(z^{-1}) &= U(z^{-1}), \quad \text{and} \\ \tilde{Y}_n(z^{-1}) &= \tilde{Y}(z^{-1}). \end{aligned} \quad (5)$$

Biquad State Space (BSS) mapped the multinotch's biquad structure to state space, first in discrete time (DT) [3] and then in continuous time (CT) [4]. 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.

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 [6] 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 [3], [4] this lack of direct feedthrough affects the propagation of states and the structure of the state space matrices.

The rest of this paper will be as follows. Section II will review the elements of the BSS and MNF needed for this paper. Options for continuous time low pass filters and rigid body models in the Multinotch Filter (MNF) and Biquad State Space (BSS) are discussed in Section III. In Section IV, we will review the discussion of lack of direct feedthrough from [4] as it is relevant to the sections that follow. In Section V, we introduce an unbundled Bilinear State Space (BLSS) form – an option when the rigid body poles and zeros are real and distinct – which can easily replace a biquad block without much pain, and allow convenient access to the internal state of a rigid body model, particularly in discrete time. Finally, Section VII provides a handful of discrete time rigid body models. Finally, Section VIII 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.

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 [7], [8], and [5]. 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 [4], and then does a biquad-by-biquad discretization to the discrete BSS [3], 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.

II. Brief Review of BSS

The fundamental blocks for the discrete time BSS were discussed in [3] while the blocks for the continuous time BSS were introduced in [4]. First, for the discrete time BSS:

$$\begin{bmatrix} x_{i,k+1} \\ x_{i,k} \end{bmatrix} = \begin{bmatrix} -a_{i1} & -a_{i2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{i,k} \\ x_{i,k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_{i,k}, \tag{6}$$

while the state output equation is given by:

$$\begin{bmatrix} \tilde{y}_{i,k} \end{bmatrix} = \begin{bmatrix} \tilde{b}_{i1} - a_{i1} & \tilde{b}_{i2} - a_{i2} \end{bmatrix} \begin{bmatrix} x_{i,k} \\ x_{i,k-1} \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} u_{i,k}. \tag{7}$$

The properly scaled output is generated via:

$$\begin{bmatrix} y_{i,k} \end{bmatrix} = \begin{bmatrix} b_{i0} \end{bmatrix} \begin{bmatrix} \tilde{y}_{i,k} \end{bmatrix}. \tag{8}$$

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

$$\begin{array}{lcll} u_{i+1,k} & = & \tilde{y}_{i,k}, & \text{for } 0 \leq i < n, \\ u_{0,k} & = & u_k, & \text{and} \\ \tilde{y}_{n,k} & = & \tilde{y}_k. & \end{array} \tag{9}$$

If one is willing to go through the algebraic pain and suffering of applying (9) to each biquad structure, the result is a very regular filter [1], [2] or state space structure [3], [4]. In the case of the continuous time BSS:

$$\begin{bmatrix} \ddot{x}_i \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} -a_{i1} & -a_{i2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_i, \tag{10}$$

while the state output equation is given by:

$$\begin{bmatrix} \tilde{y}_i \end{bmatrix} = \begin{bmatrix} \tilde{b}_{i1} - a_{i1} & \tilde{b}_{i2} - a_{i2} \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix} u_i. \tag{11}$$

The properly scaled output is generated via:

$$\begin{bmatrix} y_i \end{bmatrix} = \begin{bmatrix} b_{i0} \end{bmatrix} \begin{bmatrix} \tilde{y}_i \end{bmatrix}. \tag{12}$$

III. 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. 2 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

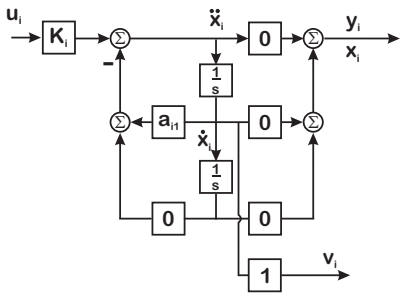

Fig. 2. Continuous time (CT) rigid body biquad. Setting $a_1 = 0$ turns it into a CT double integrator..

in an atomic force microscope (AFM) [9], [10] would require a more difficult access to velocity. The double integrator is modeled as:

$$D(s) = \frac{K}{s^2}, \tag{13}$$

while the single pole rigid body is modeled as

$$D(s) = \frac{Ka_1}{s(s+a_1)}. \tag{14}$$

where a_1 might be viscous friction applying velocity feedback.

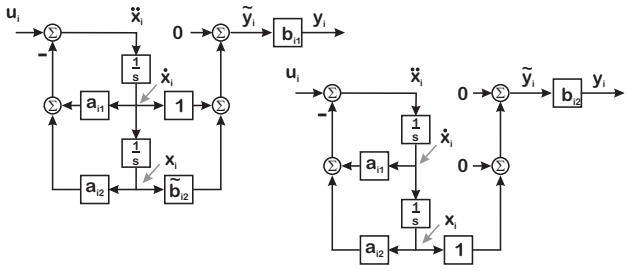

Fig. 3. Analog biquads without direct feedthrough. On the left, $b_{i0} = 0$. On the right, both b_{i0} and $b_{i1} = 0$. In either case, the leading gain is the gain of the highest order numerator term that has a non-zero coefficient. In either case, the lack of direct feedthrough means that the output is only determined by the state of the block. All downstream blocks from this one will not have direct feedthrough from the cascade input to the cascade output.

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:

$$L_{1,a}(s) = \frac{a}{s+a} \text{ and} \tag{15}$$

$$L_{1,b}(s) = \left(\frac{a}{b}\right)\frac{s+b}{s+a}. \tag{16}$$

In the case of (15), it is low pass because the "zero" is at infinite frequency. At some point, for positive a, it has to roll off. Equation (16) 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 (16). Likewise, there would be no problem with:

$$L_{2,b2}(s) = \left(\frac{a_2}{b_2}\right)\frac{s^2+b_1s+b_2}{s^2+a_1s+a_2}. \tag{17}$$

Equation (15) is a different matter as are:

$$L_{2,b0}(s) = \left(\frac{a_2}{b_2}\right)\frac{b_2}{s^2+a_1s+a_2}, \text{ and} \tag{18}$$

$$L_{2,b1}(s) = \left(\frac{a_2}{b_1}\right)\frac{s+b_1}{s^2+a_1s+a_2}. \tag{19}$$

Equations (15), (18), and (19) 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. 3 and will be discussed in Section IV.

IV. Handling Lack of Direct Feedthrough

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 VI. 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 3 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 (10) is unchanged. However, the state output equations change a lot. In the left hand case, (11) becomes

$$\begin{bmatrix} \tilde{y}_i \end{bmatrix} = \begin{bmatrix} 1 & \tilde{b}_{i2} \end{bmatrix}\begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u_i \tag{20}$$

where $\tilde{b}_{i2} = b_{i2}/b_{i1}$ and (12) becomes:

$$\begin{bmatrix} y_i \end{bmatrix} = \begin{bmatrix} b_{i1} \end{bmatrix}\begin{bmatrix} \tilde{y}_i \end{bmatrix}. \tag{21}$$

In the right hand case, (11) becomes

$$\begin{bmatrix} \tilde{y}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{x}_i \\ x_i \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u_i \tag{22}$$

and (12) becomes:

$$\begin{bmatrix} y_i \end{bmatrix} = \begin{bmatrix} b_{i2} \end{bmatrix}\begin{bmatrix} \tilde{y}_i \end{bmatrix}. \tag{23}$$

This may seem like an awful lot of bookkeeping for such a simple concept, but doing this bookkeeping allows us to maintain a the overall system structure, which allows us to

write scripts and programs to build up BSS matrices from individual biquad models. A four biquad system model that illustrates propagating the lack of direct feedthrough in the BSS appears in [4].

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.

V. Bilinear State Space Form

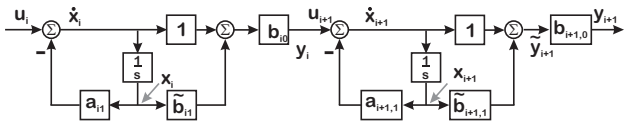

Fig. 4. Continuous time bilinear state space (CT-BLSS) form.

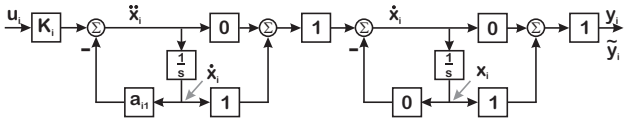

Fig. 5. Continuous time rigid body BLSS model. Note that rather than indexing the second stage as $i+1$, we stick with i but label the level of integration on the signals.

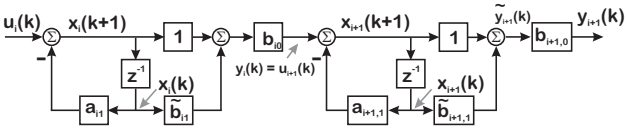

Fig. 6. Discrete time bilinear state space form (DT-BLSS).

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. In rigid body models, 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:

$$\dot{x} = F_C x + G_C u,\ y = H_C x + D_C u \text{ and} \tag{24}$$

$$x(k+1) = Fx(k) + Gu(k),\ y = Hx(k) + Du(k), \tag{25}$$

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

$$F = \begin{bmatrix} -a_{i+1,1} & b_{i,0}(\tilde{b}_{i1} - a_{i1}) \\ 0 & -a_{i,1} \end{bmatrix}, \tag{26}$$

$$H = \begin{bmatrix} b_{i+1,0}(\tilde{b}_{i+1,1} - a_{i+1,1}) & b_{i+1,0}b_{i,0}(\tilde{b}_{i,1} - a_{i,1}) \\ 0 & b_{i,0}(\tilde{b}_{i,1} - a_{i,1}) \end{bmatrix}, \tag{27}$$

$$G = \begin{bmatrix} b_{i,0} \\ 0 \end{bmatrix}, \text{ and } D = \begin{bmatrix} b_{i+1,0}b_{i,0} \\ b_{i,0} \end{bmatrix}, \text{ where} \tag{28}$$

$$u_i = y_{i-1} \text{ for } i = 1, \ldots, n. \tag{29}$$

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

$$F_C = \begin{bmatrix} -a_{i+1,1} & b_{i,0}(\tilde{b}_{i1} - a_{i1}) \\ 0 & -a_{i,1} \end{bmatrix} \tag{30}$$

$$H_C = \begin{bmatrix} b_{i+1,0}(\tilde{b}_{i+1,1} - a_{i+1,1}) & b_{i+1,0}b_{i,0}(\tilde{b}_{i,1} - a_{i,1}) \\ 0 & b_{i,0}(\tilde{b}_{i,1} - a_{i,1}) \end{bmatrix} \tag{31}$$

$$G_C = \begin{bmatrix} b_{i,0} \\ 0 \end{bmatrix}, \text{ and } D_C = \begin{bmatrix} b_{i+1,0}b_{i,0} \\ b_{i,0} \end{bmatrix}, \text{ where} \tag{32}$$

$$u_i = y_{i-1} \text{ for } i = 1, \ldots, n. \tag{33}$$

The general continuous time BLSS model is diagrammed in Fig. 4 which simplifies to Fig.5 for the rigid body models we have discussed. The general discrete time model is diagrammed in Fig. 6. 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.

VI. 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 [1], [2], [3], [4]. The Bode plot comparisons in [4] 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 ∞" [8]. 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 [11] and the conservatism of the zero at $z = 0$ helps stabilize the overall loop. However, they are not the most accurate match to the continuous time filter.
- Map the CT zeros at $s = -\infty$ to $z = -1$. This corresponds to the trapezoidal rule equivalent and is the most accurate match for the continuous time model.
- 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 [8] and results in no direct feedthrough. Assigning both DT zeros this way results in a discrete biquad that looks like

$$B_{i,FR}(z) = \frac{b_{i,2}z^{-2}}{1 + a_{i,1}z^{-1} + a_{i,2}z^{-2}}, \tag{34}$$

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 IV. Recent work suggests designing for a system with minimum delay and then backing off bandwidth to accommodate the phase due to measured delay [12], [13].

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

$$B_{i,BR}(z) = \frac{b_{i,0}}{1 + a_{i,1}z^{-1} + a_{i,2}z^{-2}}, \tag{35}$$

where we see that the numerator delay from (34) 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

$$B_{i,TR}(z) = b_{i,0}\frac{1 + 2z^{-1} + z^{-2}}{1 + a_{i,1}z^{-1} + a_{i,2}z^{-2}}, \tag{36}$$

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

$$B_{i,TBR}(z) = b_{i,0}\frac{1 + z^{-1}}{1 + a_{i,1}z^{-1} + a_{i,2}z^{-2}}. \tag{37}$$

Again, 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 $-\infty$.

VII. Discrete Time Rigid Body Models

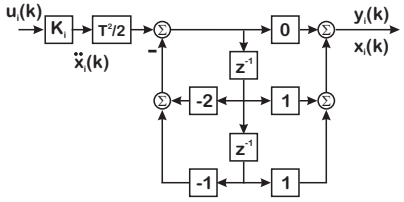

Fig. 7. Discrete double integrator biquad model (ZOH equivalent). Notice when viewed this way, the integrators are unbalanced in that the first one is implemented differently from the second.

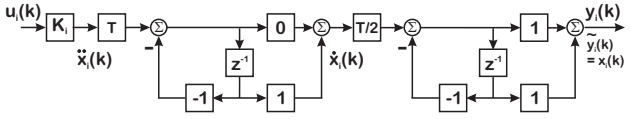

Fig. 8. Discrete double integrator BLSS model (ZOH equivalent). In this drawing, we've chosen to make the index, i, as with a biquad stage, but we are explicitly labeling the different integration levels.

In [3], we showed how using a trapezoidal rule equivalent on a double integrator preserved the feedthrough. With discrete equivalent forms of the model in Fig. 2, we run into the issue that we cannot readily access a reasonable velocity estimate from these models. Looking at the ZOH equivalent [8] model in Fig. 7 or the trapezoidal rule equivalent in Fig. 9, we can easily extract an acceleration estimate and/or a position estimate, but velocity would require some new combination of the states. In 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.

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. 5 exposes velocity. Discretizing this model with a ZOH equivalent leads to the model of Fig. 8, while a trapezoidal rule equivalent can be found in Fig. 10.

We break up the ZOH equivalent as

$$\begin{aligned} D_{ZOH}(z) &= K\frac{T^2}{2}\frac{(z+1)}{(z-1)^2} \\ &= KT\left(\frac{1}{z-1}\right)\left(\frac{T}{2}\right)\left(\frac{z+1}{z-1}\right). \end{aligned} \tag{38}$$

The blocks end up with the equations of:

$$x_{0,k+1} = KTu_{0,k} + x_{0,k} \text{ and} \tag{39}$$

$$y_{0,k} = x_{0,k}, \tag{40}$$

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

$$x_{1,k+1} = \frac{T}{2}u_{1,k} + x_{1,k} = x_{1,k} + \frac{T}{2}x_{0,k} \text{ and} \tag{41}$$

$$y_{1,k} = x_{1,k} + x_{1,k+1} = 2x_{1,k} + \tfrac{T}{2}x_{0,k}. \tag{42}$$

Together, these become:

$$\begin{bmatrix} x_{1,k+1} \\ x_{0,k+1} \end{bmatrix} = \begin{bmatrix} 1 & \frac{T}{2} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} 0 \\ KT \end{bmatrix} u_k. \tag{43}$$

The output is defined as:

$$\begin{bmatrix} y_{1,k} \\ y_{0,k} \end{bmatrix} = \begin{bmatrix} 2 & \frac{T}{2} \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u_k. \tag{44}$$

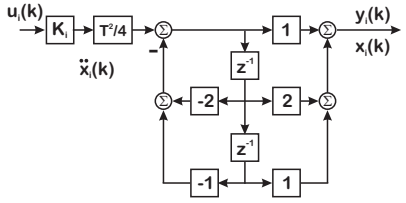

Fig. 9. Discrete double integrator biquad model (trapezoidal rule equivalent).

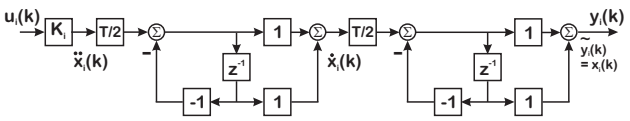

Fig. 10. Discrete double integrator BLSS model (trapezoidal rule equivalent). In this drawing, we've chosen to make the index, i, as with a biquad stage, but we are explicitly labeling the different integration levels.

This is the textbook model for a double integrator [8], but now we can access the velocity output, $y_{0,k}$ directly. This does not have direct feedthrough, unlike the trapezoidal rule model of (45). We break that up as follows:

$$\begin{aligned} D_{TR}(z) &= K\tfrac{T^2}{4}\tfrac{(z+1)^2}{(z-1)^2} \\ &= K\left(\tfrac{T}{2}\right)\left(\tfrac{z+1}{z-1}\right)\left(\tfrac{T}{2}\right)\left(\tfrac{z+1}{z-1}\right). \end{aligned} \quad (45)$$

The blocks end up with the equations of:

$$x_{0,k+1} = K\frac{T}{2}u_{0,k} + x_{0,k} \text{ and} \quad (46)$$

$$y_{0,k} = x_{0,k} + x_{0,k+1} = 2x_{0,k} + K\frac{T}{2}u_{0,k}, \quad (47)$$

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

$$\begin{aligned} x_{1,k+1} &= \tfrac{T}{2}u_{1,k} + x_{1,k} \\ &= 2x_{1,k} + Tx_{0,k} + K\tfrac{T^2}{4}u_k \text{ and} \end{aligned} \quad (48)$$

$$\begin{aligned} y_{1,k} &= x_{1,k} + x_{1,k+1} = 2x_{1,k} + K\tfrac{T}{2}u_{1,k} \\ &= 2x_{1,k} + Tx_{0,k} + K\tfrac{T^2}{4}u_k. \end{aligned} \quad (49)$$

Together, these become:

$$\begin{bmatrix} x_{1,k+1} \\ x_{0,k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} K\left(\frac{T}{2}\right)^2 \\ K\left(\frac{T}{2}\right) \end{bmatrix} u_k. \quad (50)$$

The output is defined as:

$$\begin{bmatrix} y_{1,k} \\ y_{0,k} \end{bmatrix} = \begin{bmatrix} 2 & T \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} K\left(\frac{T}{2}\right)^2 \\ K\left(\frac{T}{2}\right) \end{bmatrix} u_k. \quad (51)$$

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.

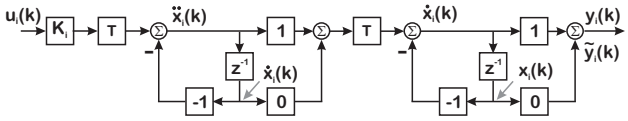

Fig. 11. Discrete double integrator BLSS model (backwards rectangular rule equivalent).

Finally, our discussion of discretization might make us consider using a backwards rule equivalent to model the double integrator. This would be broken up as

$$\begin{aligned} D_{BR}(z) &= KT^2\tfrac{(z)^2}{(z-1)^2} \\ &= KT\left(\tfrac{z}{z-1}\right)T\left(\tfrac{z}{z-1}\right). \end{aligned} \quad (52)$$

The blocks end up with the equations of:

$$x_{0,k+1} = KTu_{0,k} + x_{0,k} \text{ and} \quad (53)$$

$$y_{0,k} = x_{0,k+1} = x_{0,k} + KTu_{0,k}, \quad (54)$$

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

$$\begin{aligned} x_{1,k+1} &= Tu_{1,k} + x_{1,k} \\ &= x_{1,k} + Tx_{0,k} + KT^2u_k \text{ and} \end{aligned} \quad (55)$$

$$y_{1,k} = x_{1,k+1} = x_{1,k} + Tx_{0,k} + KTu_{1,k}. \quad (56)$$

Together, these become:

$$\begin{bmatrix} x_{1,k+1} \\ x_{0,k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} KT^2 \\ KT \end{bmatrix} u_k. \quad (57)$$

The output is defined as:

$$\begin{bmatrix} y_{1,k} \\ y_{0,k} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_{1,k} \\ x_{0,k} \end{bmatrix} + \begin{bmatrix} KT^2 \\ KT \end{bmatrix} u_k. \quad (58)$$

While this is not as true to the continuous time model as the trapezoidal rule, it does have the nice property of minimizing latency with the two zeros at $z = 0$. This property makes the model more robust to time delay, much in the same way that the backwards rule equivalent discretization used in most digital PID controllers makes the controller more robust.

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 [1], [4], but should present no problem with the rigid body modes.

VIII. Examples

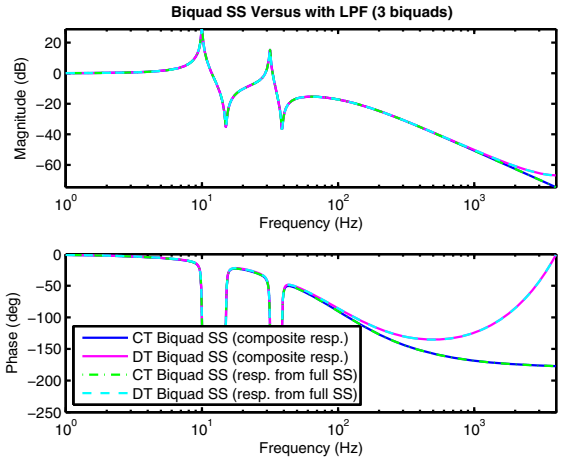

Fig. 12. Biquad State Space with three biquads including a low pass filter. The CT biquad plots include a composite of the individual CT biquad Bode plots (blue) and a Bode of the complete CT BSS structure (green). The DT biquad plots also include a composite of individual DT biquad Bode plots (magenta) 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 $z = -1$ due to the use of a Trapezoidal rule equivalent discretization.

Fig. 12 demonstrates the Bode plot of a BSS model with a low pass filter (LPF) as part of a 3-biquad model. Note the close match between the composite responses (individual biquad frequency responses combined) versus responses extracted directly from the full BSS structure. 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. 13 shows that the CT and DT match is across all the

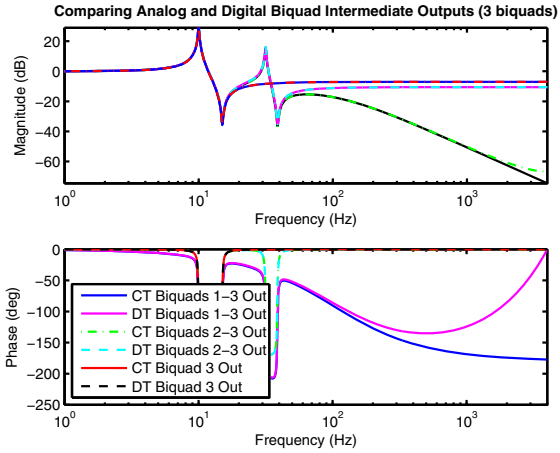

Fig. 13. BSS with three biquads including a low pass filter in Biquad 1. This plot compares the Bode responses of the individual CT and DT biquad sections. The outputs of biquad 3 and biquad 2 show the magnitude and phase flattening out at high frequency (due to the matched number of poles and zeros). Once the response of biquad 1 is added in, we see the low pass rolloff of Figure 12. At each biquad output, the match between continuous and discrete responses is incredibly close, a unique and useful feature of this structure.

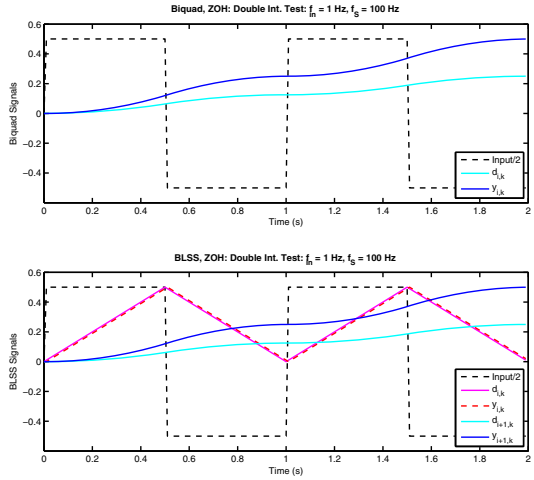

Fig. 14. Double integrator with square wave input. Implemented using a ZOH equivalent biquad (top) and BLSS (bottom).

biquad outputs, as previously demonstrated without the LPF in [4].

Fig. 14 shows at double integrator, discretized using a ZOH equivalent, implemented as a biquad (top) and a BLSS (bottom). The input-output behavior is consistent, but we now have access to the internal intermediate state with the BLSS. Fig.15, 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.

IX. Conclusions

This paper has provided some options for incorporating useful low pass filter and rigid body elements into a Bi-

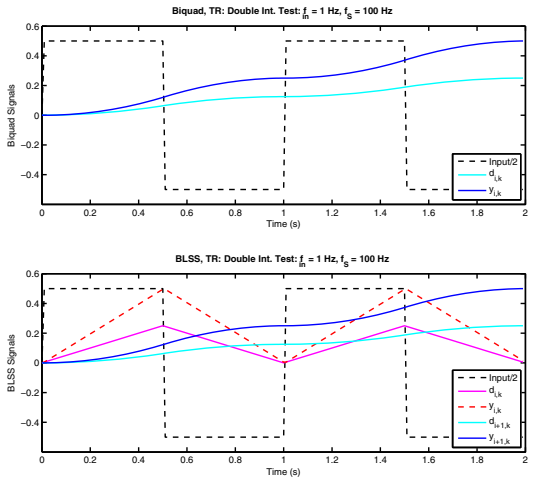

Fig. 15. Double integrator with square wave input. Implemented using a trapezoidal rule equivalent biquad (top) and BLSS (bottom).

quad State Space (BSS) structure or a Multinotch Filter (MNF). Most important was reconciling of the lack of direct feedthrough in the CT models with their DT structures, while preserving the physical state correspondence between the two. The tradeoffs with respect to the structure of several discretization methods was discussed. Furthermore, an alternate structure allowing access to the internal states of rigid body models, the Bilinear State Space Structure (BLSS) was introduced. The paper presents a toolbox to allow the reader to add these elements to their BSS and MNF models without disturbing the overall structure.

References

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

[2] D. Y. Abramovitch, "The Multinotch, Part II: Extra precision via Δ coefficients," in *Proc. Amer. Ctrl. Conf.*, (Chicago), IEEE, 2015.

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

[4] D. Y. Abramovitch, "The continuous time biquad state space structure," in *Proc. Amer. Ctrl. Conf.*, (Chicago), IEEE, 2015.

[5] T. Kailath, *Linear Systems*. Prentice-Hall, 1980.

[6] K. J. Åström and R. M. Murray, *Feedback Systems*. Princeton Univ. Press, 2nd ed., 2016.

[7] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Prent. Hall, 5th ed., 2006.

[8] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*. Add. Wesl. Long., 3rd ed., 1998.

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

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

[11] D. Y. Abramovitch, "A unified framework for analog and digital PID controllers," in *Proc. Multi-Conf. Sys. & Ctrl.*, (Sydney), IEEE, 2015.

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

[13] D. Y. Abramovitch, "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.*, (Sydney), IEEE, 2015.