Journal of Power Sources 206 (2012) 367–377

Contents lists available at SciVerse ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

Discrete-time realization of transcendental impedance models, with application to modeling spherical solid diffusion

James L. Lee [a], Andrew Chemistruck [b], Gregory L. Plett [a,*]

[a] *Department of Electrical and Computer Engineering, University of Colorado Colorado Springs, Colorado Springs, CO 80918, United States*
[b] *Texas Instruments, 2900 Semiconductor Dr., Santa Clara, CA 95051-0695, United States*

ARTICLE INFO

Article history:
Received 24 October 2011
Received in revised form
21 December 2011
Accepted 20 January 2012
Available online 31 January 2012

Keywords:
Spherical solid diffusion
Battery modeling
Model order reduction
Transcendental impedance model
Transcendental transfer function

ABSTRACT

This paper introduces the "discrete-time realization algorithm" (DRA) as a method to find a reduced-order, discrete-time realization of an infinite-order distributed-parameter system such as a transcendental impedance function. In contrast to other methods, the DRA is a bounded-time deterministic method that produces globally optimal reduced-order models. In the DRA we use the sample and hold framework along with the inverse discrete Fourier transform to closely approximate the discrete-time impulse response. Next, the Ho–Kalman algorithm is used to produce a state-space realization from this discrete-time impulse response. Two examples are presented to demonstrate the DRA using low-order rational-polynomial transfer functions, where the DRA solution can be compared to known solutions. A third example demonstrates the DRA with a transcendental impedance function model of lithium diffusion in the solid phase of a lithium-ion battery, showing that a third-order discrete-time model can closely approximate this infinite-order model behavior.

1. Introduction

Battery cell models are necessary to help gain insight into cell behavior and to be able to derive battery controls methods that are both effective and efficient. At the moment, available models tend to fall into one of two categories: theoretically developed physics-based models, and empirically justified equivalent-circuit-based models. The former are posed as coupled sets of partial-differential equations, which involve considerable computational resources to simulate, while the latter are generally computable as sets of continuous-time ordinary differential or discrete-time ordinary difference equations, which are relatively simple to simulate. The tradeoff is that the physics models provide much more predictive power and are able to apply even to unusual operating situations, while the empirical models should not be used to extrapolate beyond the original data used to generate their parameters.

The motivation for research into reduced order models is driven by a need for a model with predictive properties that closely match the partial differential equation (PDE) model but with computational efficiency similar to equivalent-circuit models. Some examples of reduced order models include Subramanian's single-particle model [1], Cai's proper orthogonal decomposition method [2], and Smith's 1D transcendental transfer function model [3]. Of these, we believe that Smith's approach has the greatest promise for applications that require frequent recalculation of the reduced order model as parameters in the cell change due to various aging mechanisms.

Smith derives Laplace (frequency) domain transfer functions (impedance models) between cell input current and solid surface concentration, overpotential, Butler–Volmer kinetics, and solid-electrolyte potential difference. He then uses two approaches to reduce these infinite order equations. In [4], high-order poles are truncated and the closely placed modes are combined, taking into account the residue at each pole (this method does not work in general, but does for this example). In [3], he developed a reduced order model by selecting poles and residues to minimize a cost function in the frequency domain via a nonlinear optimization. This process works well when it is done only once and can be supervised by a knowledgeable engineer. However, we envision the parameters of the physics models changing over a cell's life as it ages, and the need for unsupervised model order reduction. Nonlinear optimization is not ideal for this scenario: results are very sensitive to initial conditions chosen, a global optimum is not guaranteed, the final reduced-order model dimension is not at all clear except by trial and error, and the optimization process is unbounded in time and computation.

In this paper, we introduce a new method to automatically reduce a transfer function to an optimized reduced-order model. We call this method the "discrete-time realization algorithm" or

[*] Corresponding author. Tel.: +1 719 255 3468; fax: +1 719 255 3589.
E-mail addresses: jlee3@uccs.edu (J.L. Lee), andy.chemistruck@ti.com (A. Chemistruck), gplett@uccs.edu (G.L. Plett).

0378-7753/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2012.01.134

Nomenclature

a_s	specific surface area of porous electrode, $m^2\,m^{-3}$
c_s	concentration of Li in an electrode particle, $mol\,m^{-3}$
$c_{s,avg}$	average concentration of Li in an electrode particle, $mol\,m^{-3}$
$c_{s,e}$	surface concentration of Li in an electrode particle, $mol\,m^{-3}$
$\mathcal{C}_m$	extended controllability matrix with m block columns
D_s	diffusion coefficient, $m^2\,s^{-1}$
F	Faraday's constant, 96 487 $C\,mol^{-1}$
G_k	system Markov parameter k
$H(s)$	Laplace domain transfer function
$H^*(s)$	Laplace domain transfer function with a pole at the origin removed
$H(z)$	discrete domain transfer function
$\widehat{H}(z)$	discrete domain residual transfer function
$H_d[f]$	discrete Fourier transform
$h_d[n]$	approximation to the continuous time impulse response at sample n (sampling period of T_1)
h_s	discrete-time system step response with sampling period of T_s
h_{step}	approximation to the continuous-time step response (sampling period of T_1)
$\mathcal{H}_{l,m}$	block Hankel matrix with l block rows and m block columns
j^{Li}	reaction current density, $A\,m^{-3}$
$\mathcal{O}_l$	extended observability matrix
$\mathcal{O}_l^{\dagger}$	pseudo inverse of the matrix $\mathcal{O}_l$
$\mathcal{O}_l^{\uparrow}$	upward shift by one block row of the matrix $\mathcal{O}_l$
T_1	sampling period used to approximate the continuous time response, s
T_{len}	duration of sampling, s
T_s	sampling period of final discrete-time state space system, s
res_0	residue of the pole at the origin
R_s	particle radius, m
r	radial dimension, m
t	time, s
U	left singular vectors from the singular value decomposition
U_1	left singular vectors retained in the reduced order model
U_2	left singular vectors discarded from the reduced order model
V	right singular vectors from the singular value decomposition
V_1	right singular vectors retained in the reduced order model
V_2	right singular vectors discarded in the reduced order model
Σ	diagonal matrix of ordered singular values
Σ_1	diagonal matrix of ordered singular values retained in the reduced order model
Σ_2	diagonal matrix of ordered singular values discarded in the reduced order model
σ	singular values of the system

DRA. The DRA converts a nonlinear optimization problem into an equivalent linear optimization problem, and robustly solves the problem to give a globally optimal discrete-time reduced-order model. The method gives insight into the "best" reduced order model dimension and executes in deterministic time, suitable for unsupervised operation in an embedded battery management system.

The DRA consists of approximating the discrete-time impulse response of a Laplace-domain transfer function and then using this impulse response with the Ho–Kalman algorithm [5] to produce a discrete-time state space realization. We approximate the discrete-time impulse response assuming a sample and hold circuit on the input. The Ho–Kalman algorithm uses the discrete-time impulse response to produce a state space realization.

There are different methods to produce a discrete-time impulse response from a continuous time transfer function. If the transfer function can be written as a rational polynomial, well known techniques exist to convert to a discrete time system [6]. For transcendental transfer functions, however, these techniques do not work. It is possible to approximate the transcendental transfer function using the infinite product expansion and then to obtain a discrete time impulse response [7,8]. For some transcendental transfer functions, it is possible to combine the impact of multiple poles in a method known as residue grouping to derive a reduced order system [4]. This approach is limited to transcendental transfer functions where the poles can be grouped easily. Another approach presented in [9] combines the inverse discrete Fourier transform with $\mathcal{H}_{2;}$ approximations to produce the discrete-time impulse response. In this paper, we chose to estimate the discrete-time impulse response using a sample and hold framework presented in Section 2.1.

The Ho–Kalman algorithm gives a state-space minimal realization from the discrete-time impulse response. Most applications use measured input and output data to derive the impulse response [10]. The "Eigensystem Realization Algorithm" (ERA) [11] extends the Ho–Kalman algorithm to deal with noisy input/output data [12–14]. The Ho–Kalman algorithm and ERA have been used extensively for modal analysis to determine resonant frequencies in flexible structures in cases where it is difficult to apply an impulse input to the system [15–20]. Unlike these cases, we are starting with a known transcendental transfer function and then estimating the discrete-time impulse response. The Ho–Kalman algorithm is then used, not with measured system data, but with the approximation of the discrete-time impulse response to find a discrete-time state-space reduced-order model of the system described by the transcendental transfer function.

We note that the DRA has very general application to any problem that requires a reduced-order approximate model to a higher-order transfer function, and is ideal for reducing the order of distributed parameter (infinite order) systems. Modeling battery dynamics is only one possible application. In this paper, our concern is to introduce and derive the DRA, and ultimately to show how it can be applied to one equation that is part of a battery model. A future paper will show how a full coupled reduced-order battery model can be created with the aid of the DRA.

The paper is organized as follows. In the next section, we introduce the DRA and the procedural steps to derive the discrete-time realization from a Laplace-domain transfer function. The algorithm is illustrated on a simple second-order system with a continuous-time rational-polynomial transfer function and then on a third-order system with an integrator term. In both cases, the results are compared to the exact output of the continuous to discrete conversion using the zero-order hold method. The third example illustrates the performance of the algorithm on a transcendental transfer function that models the diffusion of lithium in a porous electrode [4]. We demonstrate two methods to deal with the pole at the origin of this transfer function. The discrete-time realization from the DRA is compared to numeric solution using a 1-D parabolic–elliptic partial differential equation solver.

2. Derivation of the discrete-time realization algorithm

Given a continuous-time transfer function in the Laplace domain, $H(s) = Y(s)/U(s)$, and a sampling period, T_s, we want to derive a reduced-order discrete-time state-space realization of the form

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ y[k] &= Cx[k] + Du[k], \end{aligned} \tag{1}$$

where the first equation is known as the “state equation” and describes the dynamics of the system, and the second equation is known as the “output equation” and describes how the output $y[k] \in \mathcal{R}^q$ is computed as linear combination of the states $x[k] \in \mathcal{R}^p$ and the input $u[k] \in \mathcal{R}^m$. The matrices $A \in \mathcal{R}^{p\times p}$, $B \in \mathcal{R}^{p\times m}$, $C \in \mathcal{R}^{q\times p}$, and $D \in \mathcal{R}^{q\times m}$ are constant in this work.[1] A sufficient condition for the DRA to operate is that $H(s)$ be an element of the Hardy space $\mathcal{H}_\infty$, which implies that it is a strictly stable and proper system. This is not a necessary condition, however, as we will later generalize the method to work with systems having isolated pole(s) on the imaginary axis. Note that we do not restrict $H(s)$ to be formulated as a quotient of polynomials in the Laplace variable “s” (for which well known methods exist to find the discrete-time system).

At the heart of the DRA is the Ho–Kalman algorithm [5], which takes a system's set of Markov parameters (which, for a single-input single-output system are equivalent to its discrete-time impulse response sequence) and computes from them a discrete-time state-space model of the system. To use this algorithm, we must then first compute the discrete-time impulse response from the continuous-time transfer function. The discrete-time impulse response itself is computed from the continuous-time impulse response, which is approximated as an inverse frequency transform of the initial transfer function. We describe the algorithm in four steps, which we preview here, and discuss in more detail in the following subsections.

1 Sample the continuous-time transfer function in the frequency domain at a high rate, and take the inverse discrete Fourier transform (IDFT) to get an approximation to the continuous time impulse response.
2 Compute the discrete-time impulse response values from the continuous-time response, assuming a sample and hold circuit connected to the system input.
3 Generate a discrete-time state-space realization using the deterministic Ho–Kalman algorithm. This algorithm returns the reduced order A, B, and C matrices from the discrete-time impulse response sequence in Step 2. The order of the system is determined from the ordered singular values of the Hankel matrix computed as part of the algorithm. The D matrix is found by the initial value theorem.
4 Transform the state space system into the desired final form using a similarity transformation, if required.

We note that a system having a pole at the origin does not meet the strictly stable requirement. However, we also show that this pole can be automatically accounted for.

We now proceed to discuss the sample-and-hold framework, the Ho–Kalman method, and accounting for a possible pole at $s = 0$ in more detail.

2.1. Sample and hold framework

Knowing that we require a discrete-time impulse response in order to invoke the Ho–Kalman method, we take advantage of a well known result from discrete-time system theory. Namely, the discrete-time transfer function $H(z)$ corresponding to a continuous time transfer function, $H(s)$, assuming that the input to $H(s)$ is piecewise constant with period T_s, is [6]:

$$\begin{aligned} H(z) &= \mathcal{Z}\left[\frac{1-e^{-sT_s}}{s}H(s)\right] \\ &= \mathcal{Z}\left[(1-e^{-sT_s})\frac{H(s)}{s}\right] \\ &= \mathcal{Z}\left[\frac{H(s)}{s}\right] - z^{-1}\mathcal{Z}\left[\frac{H(s)}{s}\right]. \end{aligned} \tag{2}$$

This equation is typically used for analytic computations, but we will use it numerically. Note that the very compact notation $\mathcal{Z}[\cdot]$ means: “find the continuous-time time-domain function corresponding to the Laplace-transform frequency-domain argument, then sample the continuous-time time-domain function, then take the z-transform of the samples.” Noting that $H(s)/s$ is the step response of $H(s)$, we see that Eq. (2) shows that the discrete-time impulse response corresponding to $H(z)$ is a one-step difference of the z-transform of the sampled continuous-time step response. This implies that

$$h[k] = h_s[k] - h_s[k-1], \tag{3}$$

where $h_s[k]$ is the discrete-time step response of the system with $H_s(z) = \mathcal{Z}[H(s)/s]$.

We are interested in computing $h[k]$ of Eq. (3), so we must first compute $h_s[k]$. This could be done by working with $H(s)/s$, but doing so has numeric issues as $s \to 0$. Instead, we recognize that $H(s)/s$ represents the step response of the system, which can also be computed in the time domain by integrating the impulse response. That is the approach we take here.

We approximate the continuous-time impulse response via a “discrete equivalent” or frequency-domain emulation approach [21]. This allows us to write

$$H_d(z) \approx H(s)\Big|_{s=\frac{2}{T_1}\frac{z-1}{z+1}}$$

where T_1 is an emulation sampling period (different from and generally significantly shorter than the final system sampling period T_s).[2]

We now recognize that the discrete Fourier transform (DFT) of a sequence is related to its z-transform via the relationship [23]

$$\begin{aligned} H_d[f] &= H_d(z)\Big|_{z=\exp(j2\pi f/N)} \\ &= H\left(\frac{2}{T_1}\left[\frac{e^{j2\pi f/N}-1}{e^{j2\pi f/N}+1}\right]\right), \quad 0 \le f < N, \end{aligned} \tag{4}$$

where N is the number of points chosen for the underlying sequence, and is usually chosen to be a power of 2 for efficient computations. Criteria for choosing N are given in Section 2.2. The inverse DFT of $H_d[f]$ gives $h_d[n]$, which is the approximation of

[1] Note that the true system being approximated has state dimension n, but we are deriving a reduced order approximation to the true system having state dimension $p \le n$.

[2] In order to arrive at an accurate estimation of the continuous time transfer function, the sampling frequency, $F_1 = 1/T_1$, must be high enough to capture the system dynamics. As a rule of thumb, the sampling frequency must be at least 20 times the as great as the bandwidth of the system to get an rough approximation in the frequency domain [22]. A higher emulation sampling frequency gives more accurate results.

the continuous-time impulse response at the emulation sampling period, T_1

$$h_d[n] = \frac{1}{N}\sum_{f=0}^{N-1} H_d[f]e^{j2\pi fn/N}. \tag{5}$$

The cumulative summation of this impulse response yields the approximation to the continuous-time step response

$$h_{\text{step}}[k] = T_1\sum_{i=0}^{k-1} h_d[i]. \tag{6}$$

This result is interpolated with sample period T_s to give $h_s[k]$. The difference between $h_s[k]$ and $h_s[k-1]$ gives the impulse response of the system via Eq. (3).

2.2. The Ho–Kalman method

The following discussion of the Ho–Kalman algorithm closely follows Ref. [24]. Given the impulse response of the system we find the A, B, and C state space matrices of Eq. (1) and an approximation p to the system order n.

The Markov parameters, G_k, of the system are given by,

$$G_k = \begin{cases} D, & k = 0 \\ CA^{k-1}B, & k = 1, 2, \ldots \end{cases}$$

For a single-input single-output system, these are simply equal to the discrete-time impulse-response sequence of the system (i.e., $G_k = h[k]$ from Eq. (3)). Multi-input multi-output systems can be accommodated in a straightforward way by both the Ho–Kalman method and therefore the DRA, but we omit the extra complexity from our discussion by focusing on the simpler case.

Given the Markov parameters, we can form the system block Hankel matrix

$$\mathcal{H}_{l,m} = \begin{bmatrix} G_1 & G_2 & G_3 & \cdots & G_m \\ G_2 & G_3 & G_4 & \cdots & G_{m+1} \\ G_3 & G_4 & G_5 & \cdots & G_{m+2} \\ \vdots & \vdots & \vdots & \ddots & \\ G_l & G_{l+1} & G_{l+2} & & G_{l+m-1} \end{bmatrix}.$$

We see now that the required length of the discrete-time impulse response is $N = l + m$. Further, a requirement of the Ho–Kalman method is that both m and n be at least as large as the reduced-order-model's order p. We usually choose $l = m$, so $N \geq 2n$.[3] When executing the Ho–Kalman method the first time, when the actual system order is uncertain, an assumed upper bound of the system order is chosen *a priori*. This bound can later be verified by examining the singular values of the Hankel matrix, as we will demonstrate.

An interesting feature of the block Hankel matrix, which is critical for this development, is that it can be written as $\mathcal{H}_{l,m} = \mathcal{O}_l\mathcal{C}_m$, where the extended observability matrix, $\mathcal{O}_l$, and the extended controllability matrices, $\mathcal{C}_m$, are defined as,

$$\mathcal{O}_l = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{l-1} \end{bmatrix}$$

$$\mathcal{C}_m = \begin{bmatrix} B & AB & A^2B & \cdots & A^{m-1}B \end{bmatrix}.$$

If we are able to factor $\mathcal{H}_{l,m}$ into $\mathcal{O}_l$ and $\mathcal{C}_m$, then we can directly extract the desired C matrix as the top block row of $\mathcal{O}_l$ and the desired B matrix as the left block column of $\mathcal{C}_m$. We can do further processing to find the A matrix.

To do the factoring, we rely on the singular value decomposition (SVD) [25], which allows us to write $\mathcal{H}_{l,m} = U\Sigma V^T$, where U and V are orthogonal matrices, and Σ is a diagonal matrix with non-negative entries. The diagonal of Σ holds the so-called singular values of $\mathcal{H}_{l,m}$, which are ordered such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_m$. The magnitude of the singular values is a measure of the relative importance of its corresponding state to the overall system behavior. By counting the "large" singular values, we arrive at an estimate p of the system order. We then rethink the singular value decomposition as

$$\begin{aligned} \mathcal{H}_{l,m} &= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &= \underbrace{U_1\Sigma_1V_1^T}_{\text{retained}} + \underbrace{U_2\Sigma_2V_2^T}_{\text{discarded}}, \end{aligned} \tag{7}$$

where $\Sigma_1 \in \mathcal{R}^{p\times p}$ and we retain only $\mathcal{H}_{l,m} \approx U_1\Sigma_1V_1^T$. The approximation is exact if $\Sigma_2 = 0$. We know from the Schmidt–Mirsky theorem that this approach yields a globally optimum p-rank approximation [26].

It is possible to show that the original system's extended observability matrix and the extended controllability matrix are related to the singular value decomposition via

$$\mathcal{O}_l = U\Sigma^{1/2}T, \quad \text{and} \quad \mathcal{C}_m = T^{-1}\Sigma^{1/2}V^T, \tag{8}$$

where T is an invertible similarity transformation matrix that determines the basis in which the A, B, C, and D matrices are defined. This basis is not important for the function of the reduced-order model, so we simply set $T = I$ and compute $\mathcal{O}_l = U_1\Sigma_1^{1/2}$ and $\mathcal{C}_m = \Sigma_1^{1/2}V_1^T$.

At this point the order of the reduced-order model, p, has been determined from the singular values. The B matrix has been found by taking the first block row of $\mathcal{C}_m$. Likewise, the C matrix has been found by taking the first block column of $\mathcal{O}_l$. It remains to find A and D.

The A matrix can be found via the relationship,

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \\ CA^{l-2} \end{bmatrix} A = \begin{bmatrix} CA \\ CA^2 \\ \\ \\ CA^{l-1} \end{bmatrix}.$$

The matrix on the left side of the equation is $\mathcal{O}_{l-1}$ and the matrix on the right hand side of the equation is $\mathcal{O}_l^{\uparrow}$ where the ↑ represents an upward shift of the matrix. The A matrix may be solved from this relationship by matrix manipulations and back substitution, and the solution may be written as

$$A = \mathcal{O}_{l-1}^{\dagger}\mathcal{O}_l^{\uparrow},$$

[3] For transcendental transfer functions having an infinite number of poles, we cannot choose N to be twice the number of actual poles. We find that best accuracy is achieved when N is chosen such that the approximate discrete-time impulse used as input to the Ho–Kalman algorithm response settles to near its steady-state value.

where the symbol † represents the matrix pseudo inverse [25]. Finally, the D matrix is the first Markov parameter but can often be better found numerically by using the initial value theorem for a discrete time system,

$$D = G_0 = h[0] = \lim_{z\to\infty} H(z). \tag{9}$$

The singular values of the system Hankel order provide clear insight into the relative importance of each state. Instead of merely truncating all higher order states balanced residualization [27] could be used to include some of the behavior of these less important states in the final realization. For example if a final third order system is desired, the Ho–Kalman method could be used to produce a sixth-order system. The balanced residualization technique could then be used to reduced the system to third order, incorporating behavior from the least important three states.

2.3. Dealing with one or more poles in H(s) at the origin

If the original system, $H(s)$, has a pole at the origin, it is not strictly stable, so violates the necessary conditions that make the DRA work. However, it is quite simple to deal with this case. We first subtract the pole at the origin from the transfer function, then execute the DRA on the residual system, then compute a final discrete-time state-space model that augments the DRA result with additional dynamics to implement the function of the s-domain pole at the origin (we apply the same general approach to remove multiple poles at the origin, or poles elsewhere on the imaginary axis of the s-plane).

A pole at the origin is removed by first calculating the residue of this pole and then subtracting it from the original transfer function.

$$H^*(s) = H(s) - \frac{\mathrm{res}_0}{s} \tag{10}$$

where res_0 is calculated as

$$\mathrm{res}_0 = \lim_{s\to 0} sH(s). \tag{11}$$

The remainder of the DRA is executed using $H^*(s)$ instead of $H(s)$.

To see how to re-incorporate the effect of the s-plane pole at the origin into the final reduced-order model, recall that a pole at the origin of the s-plane corresponds to an integrator. The discrete-time equivalent of an integrator with gain res_0 can be implemented as

$$\begin{aligned} x_i[k+1] &= x_i[k] + T_s u[k]. \\ y_i[k] &= \mathrm{res}_0 x_i[k] \end{aligned}$$

Therefore, we can implement a reduced-order approximation to the original system via the augmented discrete-time state-space model:

$$\underbrace{\begin{bmatrix} x_i[k+1] \\ \hdashline x[k+1] \end{bmatrix}}_{x_{\mathrm{aug}}[k+1]} = \underbrace{\left[\begin{array}{c:c} 1 & 0 \\ \hdashline 0 & A \end{array}\right]}_{A_{\mathrm{aug}}} \underbrace{\begin{bmatrix} x_i[k] \\ \hdashline x[k] \end{bmatrix}}_{x_{\mathrm{aug}}[k]} + \underbrace{\begin{bmatrix} T_s \\ \hdashline B \end{bmatrix}}_{B_{\mathrm{aug}}} u[k] \tag{12}$$

$$y[k] = \underbrace{\left[\begin{array}{c:c} \mathrm{res}_0 & C \end{array}\right]}_{C_{\mathrm{aug}}} \underbrace{\begin{bmatrix} x_i[k] \\ \hdashline x[k] \end{bmatrix}}_{x_{\mathrm{aug}}[k]} + Du[k]. \tag{13}$$

The A, B, and C matrices in the above equations are those generated by Step 3 of the DRA.

2.4. Summary of the DRA

Here, we put together all the steps of the DRA in one place, summarizing the method.

- If the original system $H(s)$ has a pole at the origin, determine res_0 using Eq. (11) and compute $H^*(s)$ using Eq. (10). Computer code can automatically determine whether $H(s)$ has a pole at the origin by substituting small values of s. If $|H(s)| \to \infty$, then this "zeroth" step must be performed. Perform the following four steps of the DRA using $H^*(s)$ instead of $H(s)$.
- Step 1: Select an emulation sampling period T_1 such that $1/T_1$ is significantly greater than the bandwidth of $H(s)$, and N is at least twice as large as the assumed reduced-order model dimension. Compute $H_d[f]$ using Eq. (4), and then the approximate continuous-time impulse response via Eq. (5).
- Step 2: Find the approximate continuous-time step response of $H(s)$ via Eq. (6). Then, interpolate at the final desired sample rate T_s to give $h_s[k]$. Difference $h_s[k]$ according to Eq. (3) to yield $h[k]$, the discrete-time impulse response for $H(s)$.
- Step 3: Populate the Hankel matrix $\mathcal{H}_{l,m}$ with the discrete-time impulse response values $h[k]$. Perform the singular value decomposition of $\mathcal{H}_{l,m}$ and determine from the singular values the system order p. Partition the singular value decomposition to give the matrices U_1, Σ_1, and V_1^T. From these, compute $\mathcal{O}_l$ and $\mathcal{C}_m$ via Eq. (8). Compute the reduced-order-model matrices A, B, and C from $\mathcal{O}_l$ and $\mathcal{C}_m$. Compute D via Eq. (9).
- Step 4: Depending on the application it may be beneficial to do a similarity transformation of the state space system. For example, diagonalizing the A matrix could speed up the calculations, which could be important if this is used in a real time control application [28].
- Finally, if the original system $H(s)$ has a pole at the origin, find the final state-space form via Eqs. (12) and (13).

Note that the steps of the DRA require only standard linear-algebra and signal-processing routines. The method finds the globally optimal reduced-order discrete-time approximation to the original continuous-time system. It does not require nonlinear optimization, and does not require iteration.

3. Examples to illustrate the method

We now look at three different examples to illustrate the operation of the DRA. The first two examples are rational-polynomial transfer functions, which we use because we can calculate the exact solution using other methods [29]. We can then compare the exact solutions to the approximate solutions obtained by the DRA. The third example does not have a closed-form exact solution, but we can use a 1-D parabolic–elliptic partial differential equation solver to find an accurate near-exact solution against which to compare the DRA solution. We find excellent agreement between the exact solutions and DRA solutions in all cases.

3.1. Example 1: rational polynomial transfer function

The DRA method is first applied to a simple second-order system. We require a discrete-time realization with the a sampling period of $T_s = 0.1$ s from the continuous-time transfer function

$$H_1(s) = \frac{s^2 + 20s + 100}{s^2 + 2s + 8}. \tag{14}$$

This system has complex poles at $-1 \pm j2.65\,\mathrm{rad\,s^{-1}}$ and two zeros at $10\,\mathrm{rad\,s^{-1}}$. The magnitude response of $H_1(s)$ is shown in Fig. 1.

Step1: The sampling frequency is selected as 256 Hz which is significantly greater than the system bandwidth. The sampling length

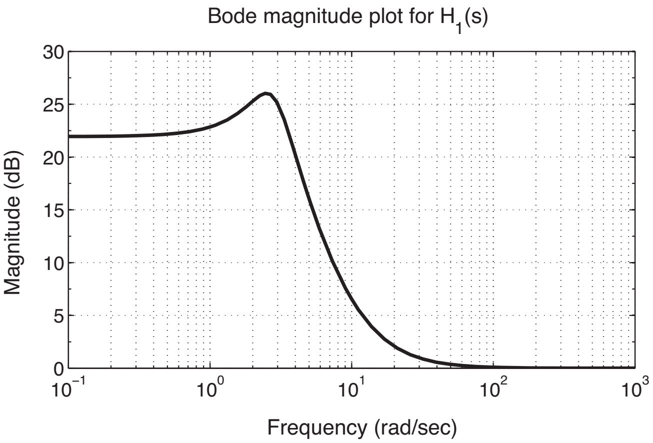

Fig. 1. Bode magnitude plot for Example 1.

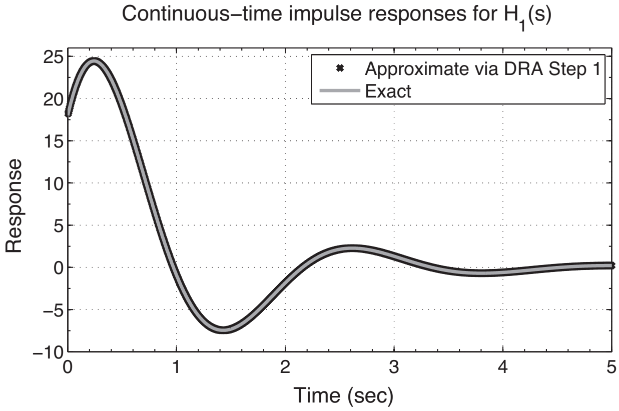

Fig. 2. Computed continuous-time impulse responses for Example 1.

N is set to 64 which allows up to a 32 × 32 system Hankel matrix in Step 3. The transfer function is sampled at discrete frequencies according to Eq. (4). The inverse DFT yields an approximation to the continuous time impulse response. Fig. 2 compares the approximate continuous-time impulse response computed via the inverse DFT to the exact continuous-time impulse response of Eq. (14). We see that the two solutions are coincident.

Step 2: The approximation to the continuous-time step response is found by doing a cumulative summation of the impulse response. The results for this example are shown in Fig. 3 and show excellent agreement with the exact step response of the continuous time system. We linearly interpolate the step response to give values at integer multiples of 0.1 s. The down-sampled step response is differenced to yield the discrete-time impulse response of Fig. 4. Again, there is excellent agreement between the approximate impulse response and the exact solution, with the exception of the single point at $t = 0$. This is often the case because of some properties of the inverse DFT, but it causes no problems since the impulse response value at $t = 0$ is not used by the Ho–Kalman algorithm in Step 3, and the *D* matrix (which is equal to the impulse response value at $t = 0$) is computed differently, using Eq. (9).

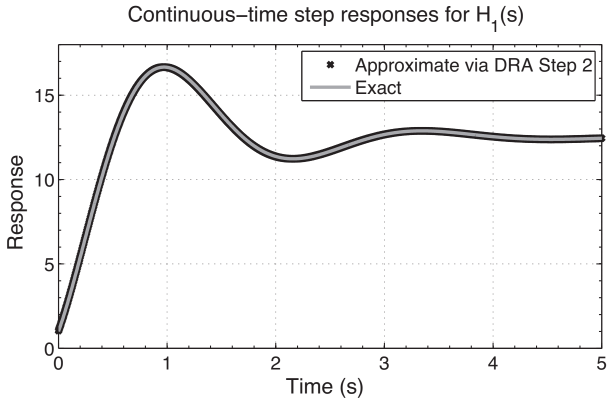

Fig. 3. Step responses for Example 1.

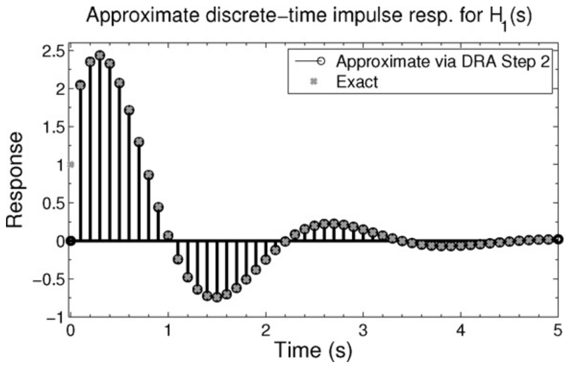

Fig. 4. Discrete-time impulse response at a sampling period of 0.1 s for Example 1.

Step 3. The deterministic Ho–Kalman algorithm is used to find a state-space realization from the approximate discrete-time impulse response from Step 2. The approximation of the continuous-time impulse response was truncated in Step 1 to 64 points, which allows a maximum Hankel matrix of 32 × 32. The SVD of the Hankel matrix gives insight into the order of the system. A log plot of the singular values is shown in Fig. 5. The first two singular values are almost three orders of magnitude greater than the third singular value, so we select a reduced-order model dimension $p = 2$.

Truncating to the first two states only, the Ho–Kalman algorithm gives a state-space realization with the following *A*, *B*, and *C* matrices

$$\begin{aligned} A &= \begin{bmatrix} 0.8656 & -0.2367 \\ 0.2367 & 0.8811 \end{bmatrix} \\ B &= \begin{bmatrix} -1.624 \\ 0.7694 \end{bmatrix} \\ C &= \begin{bmatrix} -1.624 & -0.7694 \end{bmatrix}. \end{aligned}$$

The *D* matrix is found from the initial value theorem and, for this example, is $D = [1]$.

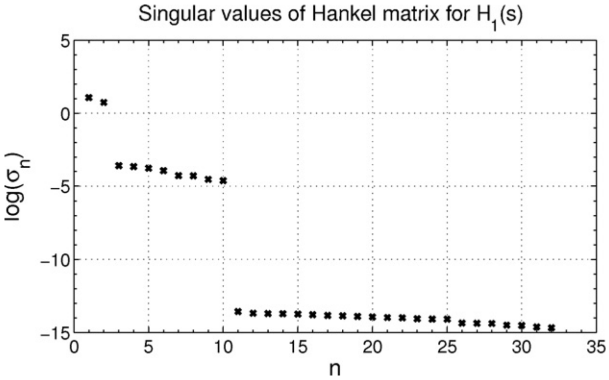

Fig. 5. Singular values indicate only two significant states in Example 1.

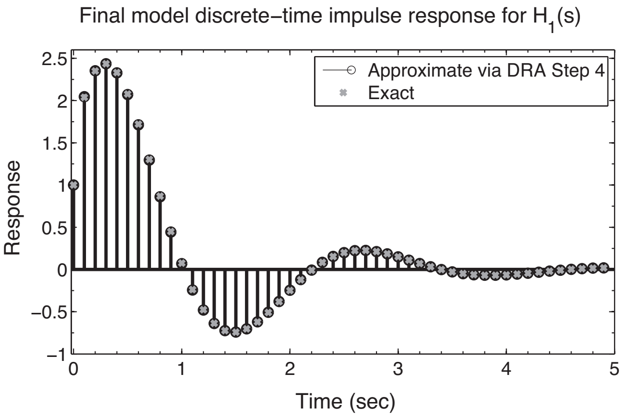

Fig. 6. Comparison of final results from the DRA to the true discrete-time solution for Example 1.

Step 4: We have chosen not to transform the system representation in this example.

A comparison of the impulse response of the DRA discrete-time state-space realization to the exact impulse response is shown in Fig. 6. The results agree very well (note that the impulse response value at $t = 0$ has been corrected by the correct calculation of the D matrix in Step 3). Because the impulse responses agree very well, the response of the reduced-order model will also agree well with the exact response for any input signal $u[k]$.

The residual error between the exact solution and the DRA is seen by comparing the pole-residue representation of the discrete time systems. The transfer function of the true discrete-time system can be written as,

$$H_1(z) = \frac{a}{z - p_1} + \frac{b}{z - p_2}$$

where p_1 and p_2 are poles and a and b are the corresponding residues. In this example the poles, p_1 and p_2, can be solved for analytically, and are $0.87335 \pm j0.2366$ and the residues likewise can be found to be $1.019 \mp j1.205$. For the DRA realization, the poles are computed to be at the same location (up to the fifth significant figure) but the residues are computed to be $1.023 \mp j1.197$. The residuals transfer function is,

$$\widehat{H}_1(z) = \frac{a - a_{\text{dra}}}{z - p_1} + \frac{b - b_{\text{dra}}}{z - p_2}$$

where a_{dra} and b_{dra} are the residues of the DRA realization. At the 256 Hz sampling frequency, the residue errors for $\widehat{H}_1(z)$ are $-0.00418 \mp j0.00762$. The residual transfer function is deterministic, but small because the residues of the true system and the DRA are close. The error can be decreased further by increasing the sampling frequency. For example, sampling at 512 Hz decreases the residue errors for $\widehat{H}_1(z)$ to $-0.00211 \mp j0.00379$.

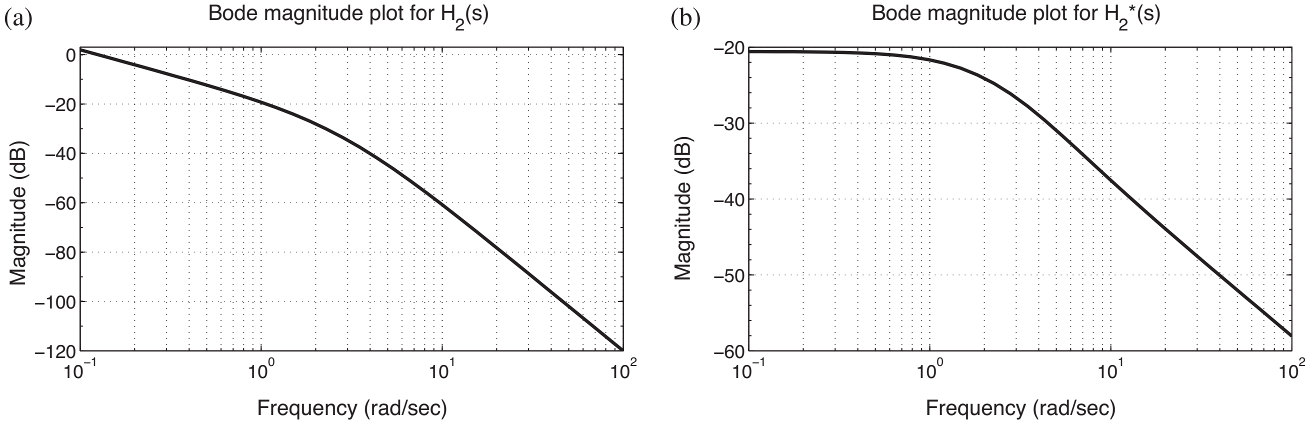

Fig. 7. Bode magnitude plots for Example 2: (a) original system, and (b) the system after removing the pole at the origin.

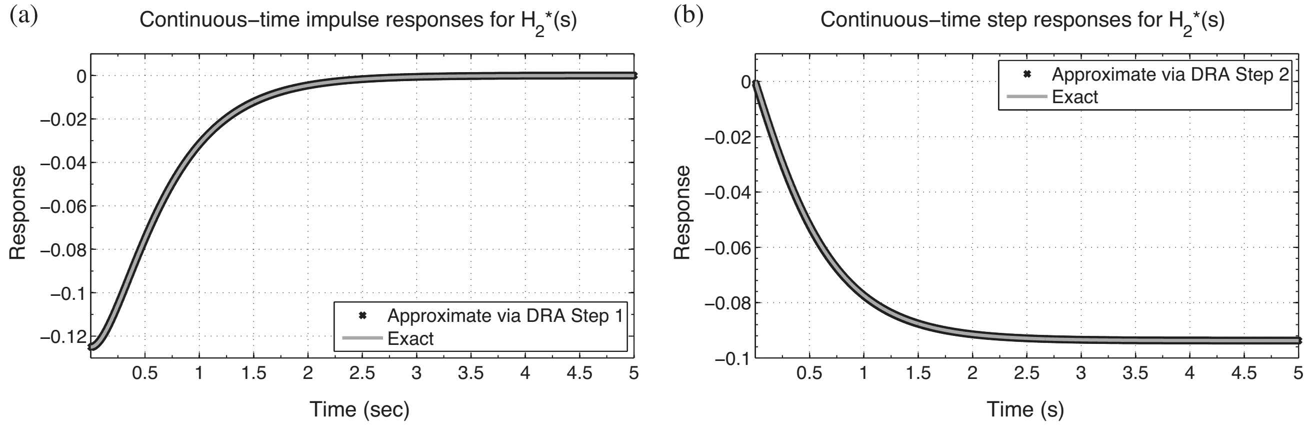

Fig. 8. Approximation of the continuous time responses for Example 2 using the DRA.

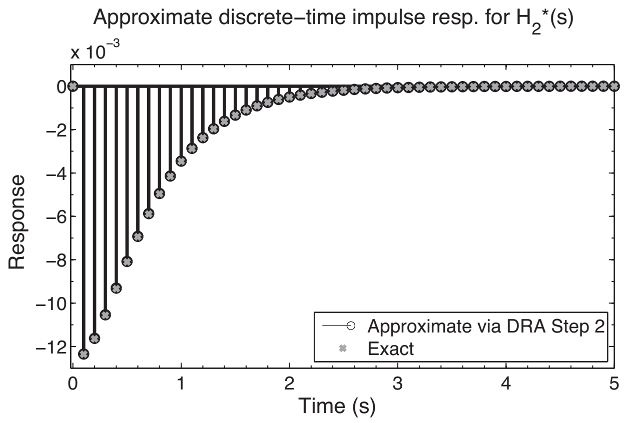

Fig. 9. System impulse response at a sampling period of 0.1 s for Example 2.

3.2. Example 2: rational polynomial transfer function with a pole at the origin

In this example, we demonstrate how to handle a single pole at the origin. The continuous-time transfer function is given by

$$H_2(s) = \frac{1}{s}\left(\frac{1}{s^2+6s+8}\right). \tag{15}$$

This system has real poles at 0, 2 and 4 rad s^{-1}. We require a discrete-time transfer function with a sampling period of $T_s = 0.1$ s. Prior to Step 1 we remove the pole at the origin. This is accomplished by first calculating the residue for this pole. In this polynomial example, the residue can be computed analytically as

$$\text{res}_0 = \lim_{s\to 0} sH(s) = 0.125.$$

In general, we find this residue by selecting a very small value for s and numerically computing res$_0$. The reduced transfer function, $H_2^*(s)$ with the pole at the origin removed is

$$H_2^*(s) = \frac{1}{s}\left(\frac{1}{s^2+6s+8}\right) - \frac{0.125}{s}.$$

Fig. 7 shows the magnitude plot of the original system and the system with the pole at the origin removed.

Step 1: $H^*(s)$ is sampled at 256 Hz which is significantly greater than the system bandwidth. The approximate continuous-time impulse response is computed and plotted in Fig. 8(a). The length of the impulse response is truncated to 64 samples which allows for a Hankel matrix up to 32 × 32.

Step 2: the approximation to the continuous-time step response of $H_2^*(s)$ is calculated as in the first example and plotted in Fig. 8(b). This step response is sampled at $T_s = 0.1$ s, and differenced to yield the discrete-time impulse response, plotted in Fig. 9.

Step 3: The system Hankel matrix is generated from the discrete-time impulse response found in Step 2. Fig. 10 depicts the 32 singular values of the system Hankel matrix. The first two singular values are two orders of magnitude greater than the third, indicating that $H_2^*(s)$ is a second order system. The Ho–Kalman algorithm generates the A, B, and C matrices after truncating all but the first two states. The value of D in this example is 0.

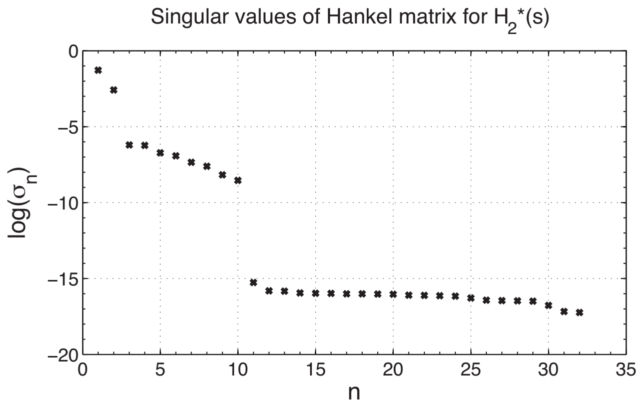

Fig. 10. Singular values of Hankel matrix in Example 2.

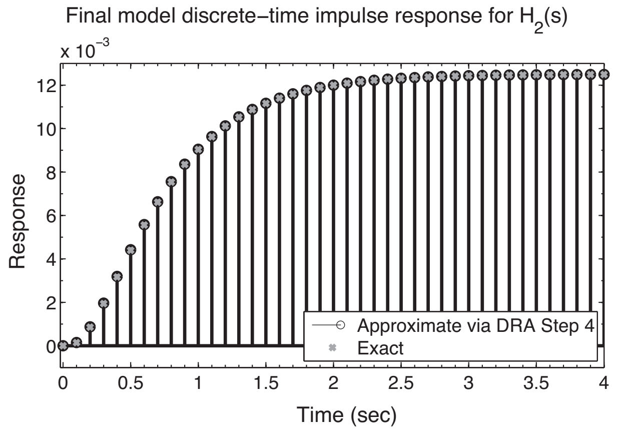

Fig. 11. Output comparison of the DRA realization to the exact answer for Example 2.

Step 4: The state-space representation found in Step 3 is augmented to include the pole at the origin according to Eqs. (12) and (13). The discrete-time realization of $H_2(s)$ is

$$A_{\text{aug}} = \left[\begin{array}{c:cc} 1 & 0 & 0 \\ \hdashline 0 & 0.8616 & -0.09064 \\ 0 & 0.09064 & 0.6276 \end{array}\right],$$

$$B_{\text{aug}} = \left[\begin{array}{c} 0.1 \\ \hdashline 0.1162 \\ -0.03401 \end{array}\right],$$

$$C_{\text{aug}} = \left[\begin{array}{c:cc} 0.125 & -0.1162 & -0.03401 \end{array}\right], \qquad D = [0].$$

Fig. 11 shows close comparison of the impulse response found from the DRA and the exact solution. The poles of the exact discrete-time system are found at 1, 0.819 and 0.670 with corresponding residues of 0.125, −0.0227 and 0.0103. The difference in residues between the exact solution and the DRA realization are 0, -8.88×10^{-5} and 8.03×10^{-5}.

3.3. Example 3: transcendental transfer function

In the first two examples, we used rational polynomials to illustrate the DRA method where order of the system is known *a priori*, and the exact answer could be calculated analytically. We will now

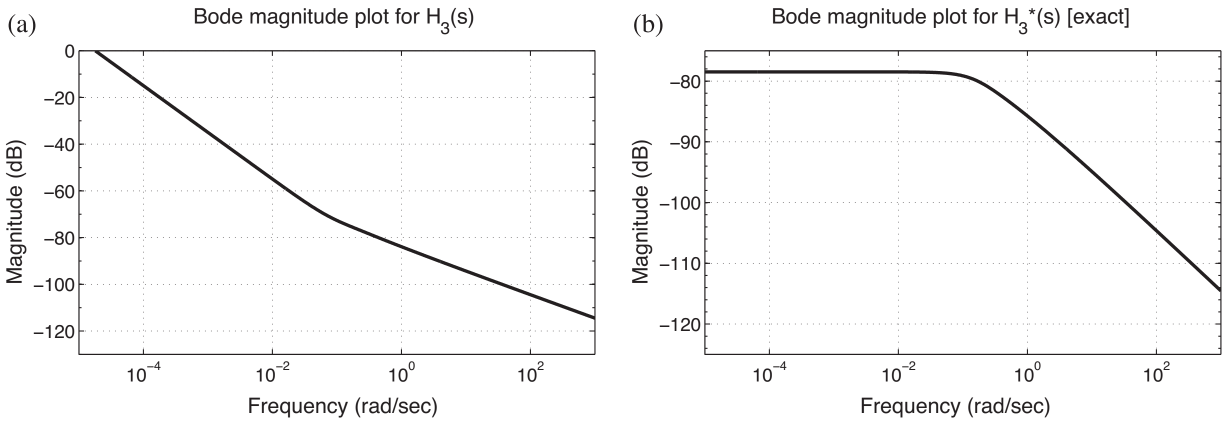

Fig. 12. Bode magnitude responses Example 3a: (a) original equation; (b) with pole at origin removed.

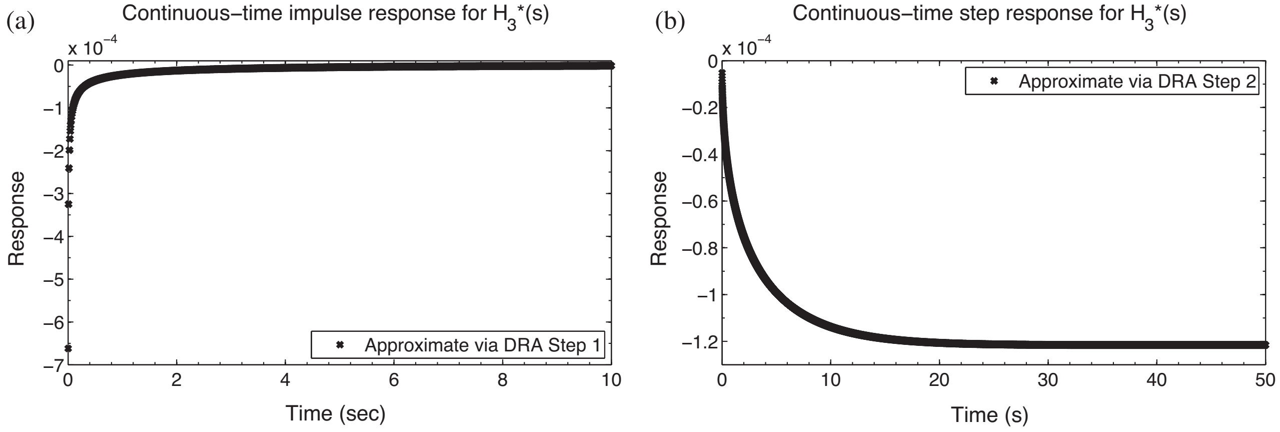

Fig. 13. Continuous time responses for Example 3a.

demonstrate the DRA with an infinite-order distributed-parameter system. Specifically the algorithm is used on a model of lithium diffusion in a spherical solid particle of an electrochemical cell which is described by,

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c_s}{\partial r}\right),$$

with boundary conditions,

$$\left.\frac{\partial c_s}{\partial r}\right|_{r=0} = 0$$

$$-D_s\left.\frac{\partial c_s}{\partial r}\right|_{r=R_s} = \frac{j^{Li}(t)}{a_sF}.$$

In these equations, c_s is the concentration of lithium in the electrode particle and r is the radial distance from the center of the particle. The range of r is from 0, the center of the particle, to R_s, the radius of solid electrode particle. D_s is the solid diffusivity and F is Faraday's constant. The specific interfacial surface area is a_s and j^{Li} is the reaction current. The continuous-time transfer function for this system was derived by Jacobsen and West [30]

$$H_3(s) = \frac{c_{s,e}(s)}{j^{Li}(s)} = \frac{R_s}{a_sD_sF}\left[\frac{\tanh(\beta)}{\tanh(\beta)-\beta}\right], \tag{16}$$

where $\beta = R_s\sqrt{s/D_s}$ and $c_{s,e}(t) = c_s(R_s, t)$ is the surface concentration of lithium. The transfer function has a pole at the origin. In Example 3a, this pole is removed in the same manner as used by Smith, which is to analytically subtract out the average concentration [4]. In Example 3b, the pole is removed by the numerical residue method. Ultimately, both examples produce a reduced-order, discrete-time, state-space realization of Eq. (16). Table 1 lists the parameters and values used in this example.

3.4. Example 3a: diffusion equation with pole at origin removed exactly, analytically

The average concentration of lithium in the solid phase is given by

$$\frac{c_{s,avg}(s)}{j^{Li}(s)} = \frac{-3}{R_sa_sF}\frac{1}{s}. \tag{17}$$

Smith showed that subtracting Eq. (17) from Eq. (16) gives the following transfer function, where $\Delta c_{s,e}(s) = c_{s,e}(s) - c_{s,avg}(s)$:

$$H_3^*(s) = \frac{\Delta c_{s,e}(s)}{j^{Li}(s)} = \frac{R_s}{a_sD_sF}\left[\frac{(\beta^2+3)\tanh(\beta)-3\beta}{\beta^2(\tanh(\beta)-\beta)}\right].$$

Table 1
Parameters used in Examples 3a and 3b.

Parameter name	Interpretation	Value
T_s	Sampling period	1 s
R_s	Particle radius	10^{-5} m
D_s	Diffusivity	10^{-12} m^2 s^{-1}
a_s	Specific interfacial surface area	1.74×10^5 m^{-1}
$c_s(r, 0)$	Initial lithium concentration	10,000 mol m^{-3}

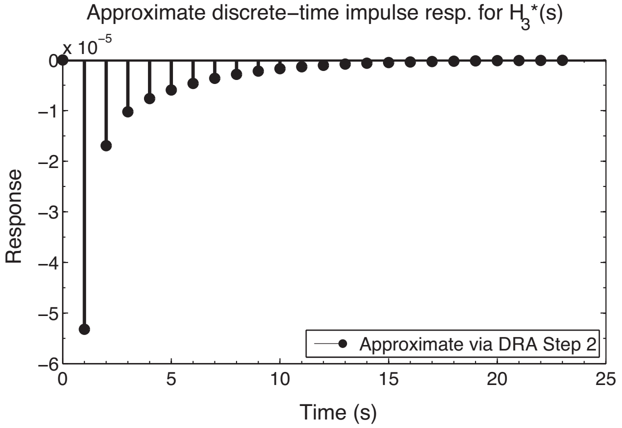

Fig. 14. Impulse response of the $H_3^*(s)$ for Example 3a.

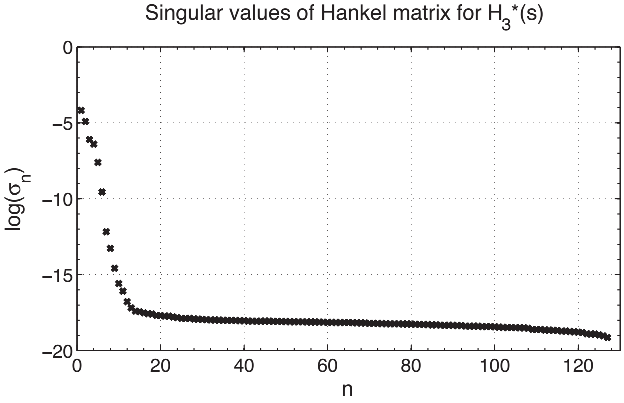

Fig. 15. Singular values of Hankel matrix for Example 3a.

We find a reduced-order model for $H_3(s)$ by first using the DRA on $H_3^*(s)$. The pole at the origin, given by Eq. (17), is then added as the final step of the procedure, and the final result is a reduced-order state-space realization of $H_3(s)$.

Step 1: The magnitude responses of $H_3(s)$ and $H_3^*(s)$ are shown in Fig. 12. $H_3^*(s)$ is sampled at 256 Hz for a total of 256 s. The approximate continuous-time impulse response is shown in Fig. 13(a). (There is no known exact solution to the continuous-time impulse response for this system against which to compare this result.)

Step 2: The approximate continuous-time step response is calculated by performing a cumulative sum of the impulse response of Step 1. Fig. 13(b) shows the approximation of the continuous time impulse and step response. (Again, there is no known exact solution to the continuous-time step response for this system against which to compare this result.) The approximate continuous-time step response is sampled at $T_s = 1$ second, and differenced to produce the discrete time impulse response, shown in Fig. 14.

Step 3: The Hankel matrix is formed, and the singular values are plotted in Fig. 15. $H_3^*(s)$ represents a distributed-parameter system that actually has an infinite number of poles. However, we see from this plot that only a few of these poles are significant to the solution. In particular, we choose to use a reduced-order model dimension $p = 2$ in the results we present here. This demonstrates a tradeoff between the complexity and accuracy of the solution. In this example, the realization is formed by truncating all but the first two states. Another approach would be to reduced the order to a value higher than two and do a balanced residualization [27] in Step 4.

Step 4: The state-space realization derived in Step 3 is augmented with the integrator state to give the final third-order state-space model of the diffusion equation. The final realization is given by,

$$A_{\text{aug}} = \left[\begin{array}{c:cc} 1 & 0 & 0 \\ \hdashline 0 & 0.4695 & 0.3296 \\ 0 & 0.3296 & 0.4355 \end{array}\right],$$

$$B_{\text{aug}} = \left[\begin{array}{c} 1 \\ \hdashline 7.094 \times 10^{-3} \\ -1.698 \times 10^{-3} \end{array}\right],$$

$$C_{\text{aug}} = \left[\begin{array}{c:cc} -1.787 \times 10^{-5} & -7.094 \times 10^{-3} & 1.698 \times 10^{-3} \end{array}\right],\ D = [0].$$

The output of this discrete-time realization to a 10 s discharge followed by a 10 s rest is shown in Fig. 16(a). This is validated against a solution computed by a 1-D parabolic–elliptic partial differential equation solver. Errors between the PDE solution and DRA solution are shown in Fig. 16(b). The DRA third-order model accurately models the system behavior.

3.5. *Example 3b: diffusion equation with pole at origin removed numerically*

In Example 3a, the pole at the origin is removed exactly by subtracting out the average concentration. In general, it may be difficult or impossible to find a closed-form algebraic function that allows the pole at the origin to be removed exactly. For those situations,

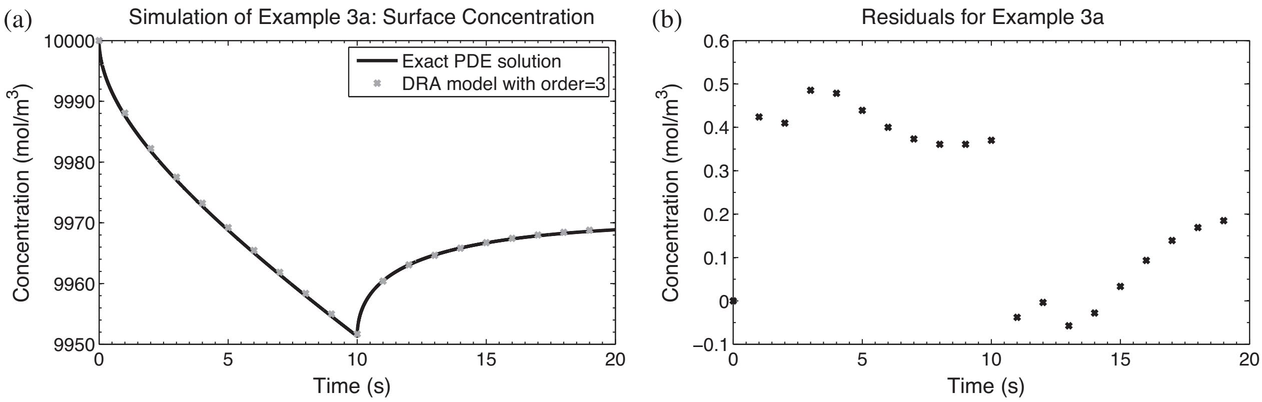

Fig. 16. Simulation results of a 10 s discharge followed by a 10 s rest for Example 3a.

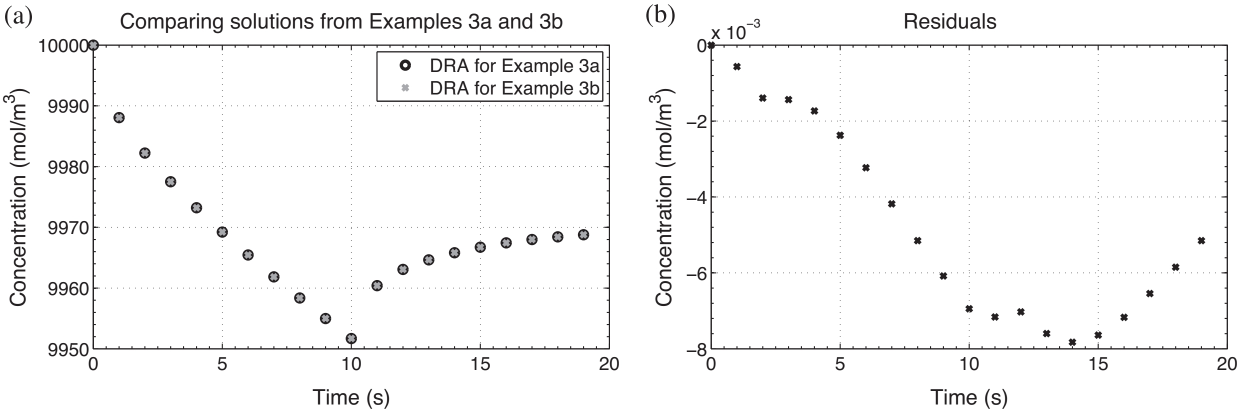

Fig. 17. Comparison of final results by removing the pole at the origin by analytic method and by the residue method for Example 3b.

the pole is numerically removed prior to Step 1. The residue for the pole at the origin is found with

$$\mathrm{res}_0 = \lim_{s\to 0} s\frac{c_{s,e}(s)}{j^{Li}(s)} = \frac{-3R_s}{a_s D_s F}$$

which we find using Mathematica. We also achieve similar results using a very small value for s (e.g., $s = 10^{-10}$). For this example $\mathrm{res}_0 = -1.787 \times 10^{-5}$. This result is subtracted from the original transfer function to give

$$\begin{aligned} H_3^*(s) &= \frac{\Delta c_{s,e}(s)}{j^{Li}(s)} \\ &= \frac{R_s}{a_s D_s F}\left[\frac{\tanh(\beta)}{\tanh(\beta)-\beta}\right] - \frac{-1.787\times 10^{-5}}{s}. \end{aligned}$$

The same sampling frequency (256 Hz) and the same impulse-response length (256 s) are used. The singular values are very similar to Example 3a and again a third-order state space model is generated. Simulation results in Fig. 17(a) show excellent agreement between the analytic method and the residue method for accounting for a pole at the origin. Figure 17(b) shows the difference between the Example 3a and Example 3b solutions, which are on the order of 10^{-3} mol m^{-3}.

4. Conclusions

In this paper, we introduced the discrete-time realization algorithm (DRA) which is used to arrive at a discrete-time, reduced-order state-space model of a Laplace domain transfer function. This method is based on deriving an approximation to the impulse response of the system and then using this impulse response with the Ho–Kalman algorithm. Although the examples were for single-input–single-output systems this, method works with multiple-input multiple-output systems as well. Note that the steps of the DRA require only standard linear-algebra and signal-processing routines. The method finds the globally optimal reduced-order discrete-time approximation to the original continuous-time system. It does not require nonlinear optimization, and does not require iteration. In a future work we will use the DRA to derive a reduced-order single-input multiple-output lithium-ion cell model.

Acknowledgement

Financial support for the research reported in this paper has been received from National Semiconductor Corporation (now Texas Instruments).

References

[1] V.R. Subramanian, V.D. Diwakar, D. Tapriyal, Journal of The Electrochemical Society 152 (2005) A2002–A2008.
[2] L. Cai, R.E. White, Journal of The Electrochemical Society 156 (2009) A154–A161.
[3] K.A. Smith, C.D. Rahn, C.-Y. Wang, Energy Conversion and Management 48 (2007) 2565–2578.
[4] K.A. Smith, C.D. Rahn, C.-Y. Wang, Journal of Dynamic Systems, Measurement, and Control 130 (2008) 011012.
[5] B. Ho, R. Kalman, Regelungstechnik 14 (1966) 545–548.
[6] G.F. Franklin, J.D. Powell, M. Workman, Digital Control of Dynamic Systems, 3rd ed., Addison-Wesley, 1998, pp. 73–153.
[7] D.C. Garvey, A.F. D'Souza, Journal of Basic Engineering 89 (1967) 327–333.
[8] R. Goodson, Simulation 15 (1970) 255–263.
[9] B. Kouvaritakis, J.A. Rossiter, M.S. Trimboli, International Journal of Control 51 (1990) 1015–1049.
[10] A. Tether, IEEE Transactions on Automatic Control 15 (1970) 427–436.
[11] J. Juang, R. Pappa, Journal of Guidance Control and Dynamics 8 (1985) 620–627.
[12] J.-N. Juang, R.S. Pappa, Journal of Guidance, Control, and Dynamics 9 (1986) 294–303.
[13] B. Peeters, G.D. Roeck, Journal of Dynamic Systems, Measurement, and Control 123 (2001) 659–667.
[14] P. Li, S.L.J. Hu, H.J. Li, Procedia Engineering, vol. 14, 2011, pp. 1681–1689.
[15] R.D. Nayeri, F. Tasbihgoo, M. Wahbeh, J.P. Caffrey, S.F. Masri, J.P. Conte, A. Elgamal, Journal of Engineering Mechanics 135 (2009) 669–683.
[16] C. Lin, C. Huang, C. Lee, M. Chao, IEEE Transactions on Industrial Electronics 58 (2011) 5206–5219.
[17] J.M. Caicedo, Experimental Techniques 35 (2011) 52–58.
[18] D. Chiang, C. Lin, Journal of Mechanical Science and Technology 24 (2010) 2377–2382.
[19] H. Moncayo, J. Marulanda, P. Thomson, Journal of Aerospace Engineering 23 (2010) 99–104.
[20] R. Spadavecchia, A. De Stefano, D. Sabia, On the applicability of the Ho–Kalman minimal realization theory, volume 347 of Key Engineering Materials, 2007.
[21] G.F. Franklin, J.D. Powell, M. Workman, Digital Control of Dynamic Systems, 3rd ed., Addison-Wesley, 1998, pp. 187–210.
[22] G.F. Franklin, J.D. Powell, M. Workman, Digital Control of Dynamic Systems, 3rd ed., Addison-Wesley, 1998, pp. 449–478.
[23] A.V. Oppenheim, R.W. Schafer, Discrete-Time Signal Processing, 3rd ed., Prentice Hall, 2010.
[24] T. Katayama, Subspace Methods for System Identification, Springer, 2005.
[25] G. Strang, Introduction to Linear Algebra, 4th ed., Wellesley-Cambridge Press, 2009.
[26] Antoulas, Sorensen, Gugercin, Contemporary Mathematics (2001) 193–219.
[27] S. Skogestad, I. Postlethwaite, Multivariable Feedback Control, 2nd ed., Wiley, 2005.
[28] C.-T. Chen, Linear System Theory and Design, 3rd ed., Oxford University Press, 1998.
[29] G.F. Franklin, J.D. Powell, A. Emami-Naeini, Feedback Control of Dynamic Systems, 6th ed., Prentice Hall, 2009.
[30] T. Jacobsen, K. West, Electrochimica Acta 40 (1995) 255–262.