Implementing Systems with Two Point Boundary Conditions for A CACSD Package of Sampled-Data Systems

Hisaya Fujioka
Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
fujioka@i.kyoto-u.ac.jp

***Abstract*— A library for systems with two point boundary conditions is implemented on MATLAB for development of a CACSD package for sampled-data systems. It is shown that we can handle various setups of sampled-data systems including multirate control and input/output time-delay in a unified way by using the systems with two point boundary conditions. Based on the developed library with object-oriented programming feature, it is easy to implement analysis and design algorithms of sampled-data systems with high readability.**

I. INTRODUCTION

Theory for analysis and design of sampled-data feedback control systems has been successfully developed in the last decade. Now we can take into account of intersample behavior of sampled-data systems in analysis and synthesis. See [2] and references therein. In spite of the development of the theory, however, there are less real applications. One of the main reasons would be that we do not have a widely spread CACSD package for sampled-data systems.

This paper concerns with a CACSD package for sampled-data systems. Actually we implement a library for systems with two point boundary conditions [8] on MATLAB[1] for easy developement of a CACSD package.

This work is motivated as follows: Consider a sampled-data system depicted in Fig. 1, where G_c, K_d, S, and H are a continuous-time and a discrete-time systems, a sampler, and a hold respectively. It has been shown that there is a discrete-time system G_s such that the following are equivalent:

(i) $\mathbf{L}_2$-induced norm from w_c to z_c is less than 1
(ii) $\mathbf{H}_\infty$ norm of a feedback connection of G_s and K_d is less than 1

under some reasonable assumptions [1], [6], [7]. Following [1], for example, G_s is given by

$$\begin{bmatrix} x_s[k+1] \\ z_s[k] \\ y[k] \end{bmatrix} = \begin{bmatrix} A_s & B_{s1} & B_{s2} \\ C_{s1} & 0 & D_{s12} \\ C_{s2} & 0 & 0 \end{bmatrix} \begin{bmatrix} x[k] \\ w_s[k] \\ u[k] \end{bmatrix}$$

$$\begin{bmatrix} B_{s1}B_{s1}' & \begin{bmatrix} A_s & B_{s2} \end{bmatrix} \\ \begin{bmatrix} A_s' \\ B_{s2}' \end{bmatrix} & \begin{bmatrix} C_{s1}' \\ D_{s12}' \end{bmatrix} \begin{bmatrix} C_{s1}' \\ D_{s12}' \end{bmatrix}' \end{bmatrix}$$

$$:= \begin{bmatrix} 0 & \begin{bmatrix} A & B_2 \end{bmatrix} \\ \begin{bmatrix} A' \\ B_2' \end{bmatrix} & 0 \end{bmatrix} + \begin{bmatrix} \acute{B}_1 & 0 \\ 0 & \begin{bmatrix} \grave{C}_1 & \grave{D}_{12} \end{bmatrix}^* \end{bmatrix} \begin{bmatrix} I & -\breve{D}_{11}^* \\ -\breve{D}_{11} & I \end{bmatrix}^{-1} \begin{bmatrix} \acute{B}_1^* & 0 \\ 0 & \begin{bmatrix} \grave{C}_1 & \grave{D}_{12} \end{bmatrix} \end{bmatrix} \quad (1)$$

[1]MATLAB is a trademark of MathWorks, Inc.

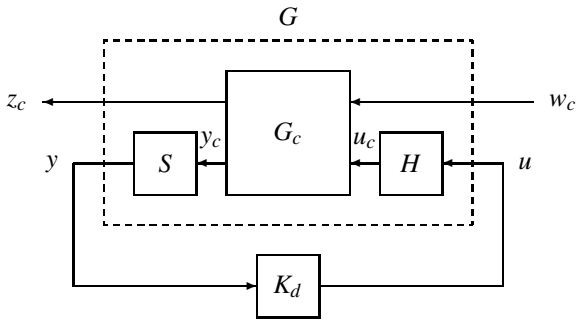

Fig. 1. Sampled-Data Feedback Control Systems

where matrices and operators in the RHS of (1) are from the state-space realization of the lifted system [1], [9] of G in Fig. 1 (See details below):

$$\begin{bmatrix} x[k+1] \\ z[k] \\ y[k] \end{bmatrix} = \begin{bmatrix} A & \acute{B}_1 & B_2 \\ \grave{C}_1 & \breve{D}_{11} & \grave{D}_{12} \\ C_2 & \acute{D}_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x[k] \\ w[k] \\ u[k] \end{bmatrix} \quad (2)$$

In a CACSD package for sampled-data systems, we should implement (1) in a numerically computable form. Actually we can reduce the computation in (1) to that of a matrix exponential:

$$\begin{bmatrix} B_{s1}B_{s1}' & \begin{bmatrix} A_s & B_{s2} \end{bmatrix} \\ \begin{bmatrix} A_s' \\ B_{s2}' \end{bmatrix} & \begin{bmatrix} C_{s1}' \\ D_{s12}' \end{bmatrix} \begin{bmatrix} C_{s1}' \\ D_{s12}' \end{bmatrix}' \end{bmatrix} \quad (3)$$

$$= \left[\begin{array}{ccc} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\ \hline 0 & 0 & -I \\ \Gamma_{41} & \Gamma_{42} & \Gamma_{43} \end{array}\right] \left[\begin{array}{c|cc} 0 & I & 0 \\ 0 & 0 & I \\ -\Gamma_{33}^{-1} & -\Gamma_{33}^{-1}\Gamma_{31} & -\Gamma_{33}^{-1}\Gamma_{32} \end{array}\right]$$

where Γ_{ij}'s are defined by

$$\left[\begin{array}{cc|cc} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} & 0 \\ 0 & I & 0 & 0 \\ \hline \Gamma_{31} & \Gamma_{32} & \Gamma_{33} & 0 \\ \Gamma_{41} & \Gamma_{42} & \Gamma_{43} & I \end{array}\right] := \exp\left(\begin{bmatrix} A_u & -B_uB_u' \\ C_u'C_u & -A_u' \end{bmatrix} h \right),$$

$$\left[\begin{array}{cc} A_u & B_u \\ C_u & * \end{array}\right] := \left[\begin{array}{cc|c} A_c & B_{c2} & B_{c1} \\ 0 & 0 & 0 \\ \hline C_{c1} & D_{c12} & * \end{array}\right]$$

if S is an ideal sampler of period $h > 0$, H is a ZOH, and G_c is given by

$$G_c : \left[\begin{array}{c} \dot{x}_c(t) \\ z_c(t) \\ y_c(t) \end{array}\right] = \left[\begin{array}{ccc} A_c & B_{c1} & B_{c2} \\ C_{c1} & D_{c11} & D_{c12} \\ C_{c2} & D_{c21} & 0 \end{array}\right] \left[\begin{array}{c} x_c(t) \\ w_c(t) \\ u_c(t) \end{array}\right]. \quad (4)$$

with $D_{c11} = 0$ and $D_{c21} = 0$. Hence we can implement (4) instead of (1). Implementations are found in [3], [5].

We, however, encounter cases of S is not an ideal sampler or H is not a ZOH in practical setups of sampled-data systems as we will see below. In the cases we can not use the matrix exponential formula (4) although the operator formula (1) is still true since matrices and operators in (2) are different (See details below).

In order to avoid to derive matrix exponential formulas for the cases individually, it is desirable to develop a software environment such that we can implement the operator formula (1) directly. It also enhances the readability of the program and makes development and maintenance easy. We can develop such software environment as a library on the object-oriented programming language. In this paper, we use the systems with two point boundary conditions [8] for unified representation of operators in (2) and MathScript on MATLAB as an object-oriented programming language.

This paper is organized as follows: Section II reviews the theory for systems with two point boundary conditions and derives a condition to check minimality of the systems. Section III shows that we can unify representation of operators related to lifted sampled-data systems in practical setups by using the systems with two point boundary conditions. Section IV describes the detail of the implementation. Section V concludes this paper.

II. SYSTEMS WITH TWO POINT BOUNDARY CONDITIONS

Mirkin and Palmor [8] proposed to use the state-space systems with two point boundary conditions, SSBC for short, as a unified representation of the operators related to the lifting [1], [9] of sampled-data systems, together with the impulse and the sample operators.

In this section, we review basic properties of SSBCs taken from [8]. We also derive conditions to check the minimality of SSBCs, those are easily implementable on a CACSD package.

Definition 1: We call the following linear differential equation with a boundary constraint as an SSBC:

$$\left[\begin{array}{c} \dot{x}(t) \\ y(t) \end{array}\right] = \left[\begin{array}{cc} A & B \\ C & D \end{array}\right] \left[\begin{array}{c} x(t) \\ u(t) \end{array}\right], \quad \Omega x(0) + \Upsilon x(h) = 0 \quad (5)$$

where $h > 0$. We also define the impulse and the sample operators by

$$(\mathscr{I}_\theta v)(t) := v\delta(t-\theta), \quad \mathscr{S}_\theta w := w(\theta) \quad (6)$$

where $\theta \in [0, h]$ and δ denotes the Dirac's delta function. Properties 1 and 2 below are found in [8]:

Property 1: G in (5) is well-posed if and only if

$$\Xi := \Omega + \Upsilon e^{Ah} \quad (7)$$

is nonsingular.

Property 2: Let G in (5) and H are well-posed SSBCs.

1) $G+H$, GH, G^* are all well-posed SSBCs.
2) G^{-1} is a well-posed SSBC.
   $\Leftrightarrow$ Both D and $\Omega + \Upsilon e^{(A-BD^{-1}C)h}$ are nonsingular.
3) $(\mathscr{S}_\theta G)^* = G^* \mathscr{I}_\theta$, $(G\mathscr{I}_\theta)^* = \mathscr{S}_\theta G^*$.
4) Let h_1, $h_2 \in [0, h]$ and $D = 0$, then

$$\mathscr{S}_{h_1} G \mathscr{I}_{h_2} = \begin{cases} Ce^{Ah_1}\Xi^{-1}\Omega e^{-Ah_2}B & (h_1 > h_2) \\ -Ce^{Ah_1}\Xi^{-1}\Upsilon e^{A(h-h_2)}B & (h_1 < h_2) \end{cases}$$

In the sequel, we use the following compact notation to represent an SSBC G in (5):

$$G = \left(\left[\begin{array}{cc} A & B \\ C & D \end{array}\right], \Omega, \Upsilon\right).$$

In order to reduce the size of the matrix exponential in Property 2 4), which is required when we perform an "equivalent" discretization of sampled-data systems, the following conditions are derived:

Property 3: The following statements are equivalent:

(i) (5) is a minimal realization.

(ii) The following state-space realization is minimal:

$$\left[\begin{array}{c} \dot{x}(t) \\ y(t) \end{array}\right] = \left[\begin{array}{c|cc} A & \Xi^{-1}\Omega M_C & \Xi^{-1}\Upsilon M_C \\ \hline M_O\Xi^{-1}\Omega & * & * \\ M_O\Xi^{-1}\Upsilon & * & * \end{array}\right] \left[\begin{array}{c} x(t) \\ u(t) \end{array}\right] \quad (8)$$

where

$$M_C := \left[\begin{array}{cccc} B & AB & \cdots & A^{n-1}B \end{array}\right],$$
$$M_O := \left[\begin{array}{cccc} C' & A'C' & \cdots & (A')^{n-1}C' \end{array}\right]'.$$

Moreover, let A_r be the 'A'-term of a minimal realization of (8) and defined by the related coordinate transformation T:

$$T^{-1}AT =: \left[\begin{array}{cc} A_r & * \\ * & * \end{array}\right].$$

Then a minimal realization of (5) is given by

$$\left(\left[\begin{array}{cc} A_r & B_r \\ C_r & D \end{array}\right], \Omega_r, \Upsilon_r\right)$$

where

$$T^{-1}B =: \left[\begin{array}{c} B_r \\ * \end{array}\right], \quad CT =: \left[\begin{array}{cc} C_r & * \end{array}\right],$$
$$T^{-1}\Xi^{-1}\Omega T =: \left[\begin{array}{cc} \Omega_r & * \\ * & * \end{array}\right], \quad T^{-1}\Xi^{-1}\Upsilon T =: \left[\begin{array}{cc} \Upsilon_r & * \\ * & * \end{array}\right].$$

The proof is found in appendix.

III. SAMPLED-DATA SYSTEMS

In order to develop a CACSD package, one of the important steps is to determine the class of systems which should be handled. In this paper we consider sampled-data systems such that the operators in the state space representation (2) of the related lifted system are given by SSBCs and linear combinations of sample and impulse operators:

$$\tilde{\mathscr{S}} = \sum_{i=1}^{n_S} \mathscr{S}_{\eta_{Si}} M_{Si}, \quad \tilde{\mathscr{I}} = \sum_{i=1}^{n_J} M_{Ji} \mathscr{I}_{\eta_{Ji}} \tag{9}$$

where η_{S_i}, $\eta_{J_i} \in [0, h)$ and M_{Si}'s and M_{Ji}'s are coefficient matrices.

Remark 1: A sum of SSBCs is again an SSBC (Property 2 1)). The same property holds for $\tilde{\mathscr{S}}$, $\tilde{\mathscr{I}}$, product of $\tilde{\mathscr{S}}$ and an SSBC, and that of an SSBC and $\tilde{\mathscr{I}}$. For example, let P_1 and P_2 are SSBCs, then we have

$$\tilde{\mathscr{S}}_1 P_1 + \tilde{\mathscr{S}}_2 P_2 = \begin{bmatrix} \tilde{\mathscr{S}}_1 & \tilde{\mathscr{S}}_2 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}.$$

In this section, we point out that a wide variety of sampled-data systems including multirate control and/or time-delay setups satisfy our assumption.

Consider first a standard sampled-data feedback system depicted in Fig. 1, where G_c is a continuous-time generalized plant given by (4). K_d is a discrete-time system. The sampler S mapping continuous-time signal y_c to a discrete-time signal y and the hold H mapping a discrete-time signal u to a continuous-time signal u_c are supposed to have the form:

$$S : y[k] = \tilde{\mathscr{S}}_S P_S (\mathscr{L} y_c)[k-1] + E_S y_c(kh), \tag{10}$$

$$H : (\mathscr{L} u_c)[k](\theta) = \left(P_H \tilde{\mathscr{I}}_H\right)(\theta) u[k], \quad \theta \in [0, h) \tag{11}$$

respectively, where E_S is a matrix of compatible size. $\tilde{\mathscr{S}}_S$ and $\tilde{\mathscr{I}}_H$ are linear combinations of sample and impulse operators as in (9). P_S and P_H are SSBCs defined by

$$P_S := \left(\begin{bmatrix} A_S & B_S \\ C_S & D_S \end{bmatrix}, I, 0 \right), \quad P_H := \left(\begin{bmatrix} A_H & B_H \\ C_H & 0 \end{bmatrix}, I, 0 \right).$$

The symbol $\mathscr{L}$ denotes the lifting [1], [9] mapping a continuous-time signal f_c to a discrete-time signal f_d taking values in a function space:

$$\mathscr{L} : f_c \mapsto f_d, \quad f_d[k](\theta) := f_c(kh + \theta), \quad \theta \in [0, h).$$

Then the lifted system of G in Fig. 1, namely,

$$\begin{bmatrix} \mathscr{L} & 0 \\ 0 & S \end{bmatrix} G_c \begin{bmatrix} \mathscr{L}^{-1} & 0 \\ 0 & H \end{bmatrix}$$

has a state-space representation (2) by defining $z := \mathscr{L} z_c$ and $w := \mathscr{L} w_c$, where matrices and operators are defined in (12) if $n_S = 0$, and in (13) if $n_S \geq 1$, respectively, where F_S is a column full rank matrix obtained by eliminating i-th column of the identity for all i's such that i-th standard basis satisfies $e_i' \tilde{\mathscr{S}}_S = 0$. We also use the following definition:

$$\begin{bmatrix} \breve{A} & \breve{B}_1 & \breve{B}_2 \\ \breve{C}_1 & \breve{D}_{11} & \breve{D}_{12} \\ \breve{C}_2 & \breve{D}_{21} & \breve{D}_{22} \end{bmatrix} := \left(\left[\begin{array}{c|ccc} A_c & I & B_{c1} & B_{c2} \\ \hline I & 0 & 0 & 0 \\ C_{c1} & 0 & D_{c11} & D_{c12} \\ C_{c2} & 0 & D_{c21} & 0 \end{array}\right], I, 0 \right).$$

Remark 2: We need to assume

(i) $E_S D_{c21} = 0$.

(ii) $M_{Si} D_S D_{c21} = 0$ for $i = 1, 2, \ldots, n_S$.

for the $\mathbf{L}_2$-stability of the system.

In the rest of this section, we show three practical setups of sampled-data systems those we can handle.

1) Multirate Control: Let us consider the situation where the sampling period is large because of the slow sensor or other constraints, while the processor to implement the discrete-time controller has sufficient computational power. We can find such situation in, e.g., hard disk drive control. Under this condition, it would be useful to switch control input at some points in the intersample, as well as at sampling instants, in order to enhance the performance of the system.

To be more concrete, we consider the following multirate control to determine the continuous-time control input u_c from the discrete-time measured output y:

$$u_c(kh + \theta) = \begin{cases} u_1[k], & \theta \in [0, h_1) \\ u_2[k], & \theta \in [h_1, h) \end{cases},$$

$$\begin{bmatrix} x_K[k+1] \\ u_1[k] \\ u_2[k] \end{bmatrix} = \begin{bmatrix} A_K & B_K \\ C_{K1} & D_{K1} \\ C_{K2} & D_{K2} \end{bmatrix} \begin{bmatrix} x_K[k+1] \\ y[k] \end{bmatrix},$$

while G_c and S are the same in the standard setup.

Although the discrete-time dynamics above is periodic, we can cast this multirate control setup into the standard setup by setting

$$\begin{bmatrix} A & \acute{B}_1 & B_2 \\ \grave{C}_1 & \breve{D}_{11} & \grave{D}_{12} \\ C_2 & \acute{D}_{21} & D_{22} \end{bmatrix} := \begin{bmatrix} \mathscr{S}_h & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \mathscr{S}_0 E_S \end{bmatrix} \begin{bmatrix} \breve{A} & \breve{B}_1 & \breve{B}_2 \\ \breve{C}_1 & \breve{D}_{11} & \breve{D}_{12} \\ \breve{C}_2 & \breve{D}_{21} & \breve{D}_{22} \end{bmatrix} \begin{bmatrix} \mathscr{I}_0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & P_H \tilde{\mathscr{I}}_H \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} A & \acute{B}_1 & B_2 \\ \grave{C}_1 & \breve{D}_{11} & \grave{D}_{12} \\ C_2 & \acute{D}_{21} & D_{22} \end{bmatrix} := \left[\begin{array}{cc|c} \mathscr{S}_h & 0 & 0 \\ 0 & \tilde{\mathscr{S}}_S P_S & 0 \\ \hline 0 & 0 & I \end{array}\right] \left[\begin{array}{ccc} \breve{A} & 0 & \breve{B}_2 \\ \breve{C}_2 & 0 & \breve{D}_{22} \\ \breve{C}_1 & 0 & \breve{D}_{12} \\ \mathscr{S}_0 E_S \breve{C}_2 & F_S & \mathscr{S}_0 E_S \breve{D}_{22} \end{array}\right] \left[\begin{array}{cc|c} \mathscr{I}_0 & 0 & 0 \\ 0 & I & 0 \\ \hline 0 & 0 & I \end{array}\right] \left[\begin{array}{c} 0 \\ 0 \\ \hline 0 \end{array}\right] \left[\begin{array}{c} 0 \\ 0 \\ \hline P_H \tilde{\mathscr{I}}_H \end{array}\right] \tag{13}$$

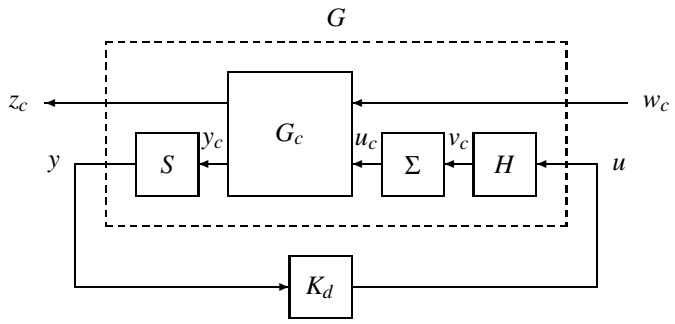

Fig. 2. Sampled-Data Systems with Input Time-Delays

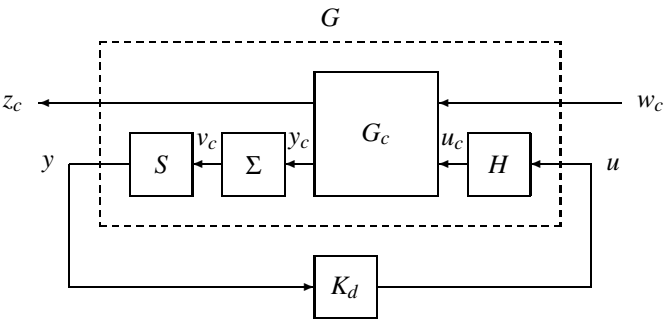

Fig. 3. Sampled-Data Systems with Output Time-Delays

$$K_d : \begin{bmatrix} x_K[k+1] \\ u[k] \end{bmatrix} = \left[\begin{array}{cc} A_K & B_K \\ \hline C_{K1} & D_{K1} \\ C_{K2} & D_{K2} \end{array}\right] \begin{bmatrix} x_K[k+1] \\ y[k] \end{bmatrix}$$

and define H by

$$P_H = \left(\left[\begin{array}{cc|cc} 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ \hline I & I & 0 & 0 \end{array}\right], I, 0 \right),$$

and

$$\tilde{\mathscr{I}}_H = \begin{bmatrix} \mathscr{I}_0 - \mathscr{I}_{h_1} & 0 \\ 0 & \mathscr{I}_{h_1} \end{bmatrix}.$$

2) Input Delays: Consider a sampled-data system with time-delays Σ at control input as depicted in Fig. 2. The following property is essentially the same that in [4]:

Property 4: Given a hold H: $u \mapsto v_c$:

$$(\mathscr{L}u_c)[k](\theta) = (P_H \mathscr{I}_\eta)(\theta) u[k], \quad \theta \in [0, h)$$

and a time-delay Σ: $v_c \mapsto u_c$:

$$u_c(t) = v_c(t - \sigma)$$

where $\eta,\ \sigma \in [0, h)$. Then $u_c = \Sigma H u$ satisfies

$$(\mathscr{L}u_c)[k](\theta) = \left(P_{H1} \tilde{\mathscr{I}}_{H1}\right)(\theta) \begin{bmatrix} u[k-1] \\ u[k] \end{bmatrix}$$

if $\eta + \sigma \le h$, where

$$P_{H1} := \begin{bmatrix} P_{H11} & P_{H12} & P_H \end{bmatrix},$$

$$P_{H11} := \left(\begin{bmatrix} A_H & B_H \\ C_H e^{A_H(h-\eta-\sigma)} & 0 \end{bmatrix}, I, 0 \right),$$

$$P_{H12} := \left(\begin{bmatrix} A_H & B_H \\ C_H e^{A_H(h-\eta)} & 0 \end{bmatrix}, I, 0 \right)$$

$$\tilde{\mathscr{I}}_{H1} := \begin{bmatrix} \mathscr{I}_0 & 0 \\ -\mathscr{I}_\sigma & 0 \\ 0 & \mathscr{I}_{\eta+\sigma} \end{bmatrix},$$

and

$$(\mathscr{L}u_c)[k](\theta) = \left(\begin{bmatrix} P_H & P_{H12} \end{bmatrix} \begin{bmatrix} \mathscr{I}_{\eta+\sigma-h} \\ -\mathscr{I}_\sigma \end{bmatrix} \right)(\theta) u[k-1]$$

if $\eta + \sigma > h$, respectively.

Hence we can obtain a state-space representation of the lifted system of G in Fig. 2 by

(i) $P_H \leftarrow \begin{bmatrix} P_H & P_{H12} \end{bmatrix}$ and $\tilde{\mathscr{I}} \leftarrow \begin{bmatrix} \mathscr{I}_{\eta+\sigma-h} \\ -\mathscr{I}_\sigma \end{bmatrix}$.

(ii) Obtain $\left[\begin{array}{ccc} A & \acute{B}_1 & B_2 \\ \grave{C}_1 & \breve{D}_{11} & \grave{D}_{12} \\ C_2 & \acute{D}_{21} & D_{22} \end{array}\right]$ for the setup in Fig. 1 with G_c, S, and H defined by P_H and $\tilde{\mathscr{I}}$.

(iii) A state-space representation of the lifted system of G is given by

$$\begin{bmatrix} x[k+1] \\ z[k] \\ y[k] \end{bmatrix} = \left[\begin{array}{cc|c|c} A & B_2 & \acute{B}_1 & 0 \\ 0 & 0 & 0 & I \\ \hline \grave{C}_1 & \grave{D}_{12} & \breve{D}_{11} & 0 \\ \hline C_2 & D_{22} & \acute{D}_{21} & 0 \end{array}\right] \begin{bmatrix} x[k] \\ w[k] \\ u[k] \end{bmatrix}$$

if $\eta + \sigma > h$. We can obtain it for the case of $\eta + \sigma \le h$ similarly.

Note also that we can apply Property 4 for general hold H in (11) and/or general time-delay Σ of the form

$$u_c(t) = \sum_{i=1}^{n_L} M_{Li} v_c(t - \sigma_i)$$

by using the linearity (See Remark 1 also).

3) Output Delays: Consider a sampled-data system with time-delays Σ at measurement output as depicted in Fig. 3. The next property is a key to treat sampled-data systems with output time-delay:

Property 5: Given a sampler S: $v_c \mapsto y$:

$$y[k] = \mathscr{S}_\eta P_S (\mathscr{L}v_c)[k-1] + E_S v_c(kh)$$

and a time-delay L: $y_c \mapsto v_c$:

$$v_c(t) = y_c(t - \sigma)$$

where $\eta \in [0, h)$, $\sigma \in (0, h)$. Then $y = S\Sigma y_c$ satisfies

$$\begin{aligned} y[k] \quad = \quad & \begin{bmatrix} \mathscr{S}_h & -\mathscr{S}_{h-\sigma} \end{bmatrix} \begin{bmatrix} P_{S1} \\ P_{S2} \end{bmatrix} (\mathscr{L}y_c)[k-2] \\ & + \begin{bmatrix} \mathscr{S}_{\eta-\sigma} & \mathscr{S}_{h-\sigma} \end{bmatrix} \begin{bmatrix} P_S \\ E_S \end{bmatrix} (\mathscr{L}y_c)[k-1] \end{aligned}$$

TABLE I
OVERLOADING MULTIPLICATION OPERATOR (*)

	ssbc	sampler	impgen	gsampler	ghold	matrix
ssbc	ssbc	–	ghold	–	ghold	ssbc
sampler	gsampler	–	–	–	matrix	sampler
impgen	–	–	–	–	–	impgen
gsampler	gsampler	–	matrix	–	matrix	gsampler
ghold	–	–	–	–	–	ghold
matrix	ssbc	sampler	impgen	gsampler	ghold	matrix

if $\eta \geq \sigma$, where

$$P_{S1} := \left(\left[\begin{array}{cc} A_S & B_S \\ C_S e^{A_S(\eta-\sigma)} & 0 \end{array} \right], I, 0 \right),$$

$$P_{S2} := \left(\left[\begin{array}{cc} A_S & B_S \\ C_S e^{A_S \eta} & 0 \end{array} \right], I, 0 \right),$$

and

$$\begin{aligned} y[k] &= \left[\begin{array}{cc} \mathscr{S}_{h+\eta-\sigma} & -\mathscr{S}_{h-\sigma} \end{array} \right] \left[\begin{array}{c} P_S \\ P_{S2} \end{array} \right] (\mathscr{L} y_c)[k-2] \\ &\quad + \mathscr{S}_{h-\sigma} E_S (\mathscr{L} y_c)[k-1] \end{aligned}$$

if $\sigma > \eta$, respectively.

Hence we can obtain a state-space representation of the lifted system of G in Fig. 3 by

(i) $\tilde{\mathscr{S}}_S \leftarrow \left[\begin{array}{cc} \mathscr{S}_{h+\eta-\sigma} & -\mathscr{S}_{h-\sigma} \end{array} \right]$ and $P_S \leftarrow \left[\begin{array}{c} P_S \\ P_{S2} \end{array} \right]$.

(ii) Obtain $\left[\begin{array}{ccc} A & \acute{B}_1 & B_2 \\ \grave{C}_1 & \breve{D}_{11} & \grave{D}_{12} \\ C_2 & \acute{D}_{21} & D_{22} \end{array} \right]$ for the setup in Fig. 1 with G_c, H, and S defined by P_S and $\tilde{\mathscr{S}}_S$.

(iii) A state-space representation of the lifted system of G is given by

$$\left[\begin{array}{c} x[k+1] \\ z[k] \\ y[k] \end{array} \right] = \left[\begin{array}{cc|c|c} A & 0 & \acute{B}_1 & B_2 \\ C_2 & 0 & \acute{D}_{21} & D_{22} \\ \hline \grave{C}_1 & 0 & \breve{D}_{11} & \grave{D}_{12} \\ \hline 0 & I & 0 & 0 \end{array} \right] \left[\begin{array}{c} x[k] \\ w[k] \\ u[k] \end{array} \right]$$

if $\sigma > \eta$. We can obtain it for the case of $\sigma \leq \eta$ similarly.

We can apply Property 5 for general sampler (10) and/or general time-delay Σ of the form:

$$v_c(t) = \sum_{i=1}^{n_L} M_{Li} y_c(t - \sigma_i)$$

by using the linearity (See Remark 1 also).

IV. IMPLEMENTATION ON MATLAB

A. Object-Oriented Programming

The main motivation of this paper is to implement formulas in operator form as in (1) directly. For the purpose, we need a programming language having the following features:

(i) One variable can represent an SSBC and linear combination of sample or impulse operators.

(ii) We can define operators such as +, *, and so on for operation among SSBCs and linear combinations of sample or impulse operators.

These requirements are satisfied by object-oriented programming languages. Among them, we choose MathScript on MATLAB to utilize the existing class definitions for CACSD.

1) Class Definitions: The following classes are defined in the developed library:

(i) `ssbc`: An object of the `ssbc` class represents an SSBC. This class is implemented as a subclass of `ss` in Control System Toolbox.

(ii) `sampler`: An object of the `sampler` class represents a linear combination of sample operators in (9).

(iii) `impgen`: An object of the `impgen` class represents a linear combination of impulse operators in (9).

(iv) `gsampler`: An object of the `gsampler` class represents a product of a linear combination of sample operators in (9) and an SSBC. For example, an object of the `gsampler` class can represent the operator $\acute{B}_1$ in (2).

(v) `ghold`: An object of the `ghold` class represents a product of an SSBC and a linear combination of impulse operators in (9). For example, an object of the `ghold` class can represent the operator $\grave{C}_1$ in (2).

2) Operator Overloading: The operator overloading feature enhances the readability, and it is a great help for easy development and maintenance.

In the developed library, operators for horizontal concatenation (`[a, b]`), vertical concatenation (`[a; b]`), addition, subtraction, multiplication, and transpose (`a'`) are overloaded. Operands for operators other than multiplication and transpose must be objects of the same class, and the results is also an object of the same class.

Table I describes order of operands and the result of the overloaded multiplication operator. We can see that, for example, the result of multiplication of an `ssbc` and an `impgen` objects is a `ghold` object. The symbol – means that the operation is undefined for the related order of the operands.

Classes of the operand and the result of the overloaded transpose operator are summarized as follows:

operand	result
ssbc	ssbc
sampler	impgen
impgen	sampler
gsampler	ghold
ghold	gsampler

B. Examples

In this subsection, we demonstrate how the readability is enhanced by using the developed library.

Consider the sampled-data system in Fig. 1 and assume the ideal sampler and the ZOH. In the case we can implement (4) instead of (1). However, we can implement (1) by using the developed library with enhancement of readability.

We can represent $\check{D}_{11}$ in (2) operator by an `ssbc` operator `D11` by the following syntax:

```
D11tmp = ss(Ac, Bc1, Cc1, Dc11);
D11 = ssbc(D11tmp, h);
```

Similarly $\acute{D}_{12}$ operator is represented by an `ghold` object `D12` defined as

```
D12tmp = ss(Ac, Bc2, Cc1, Dc12);
D12op = ssbc(D12tmp, h);
Sint = ss(0, 1, 1, 0); % integrater
Sint = ssbc(Sint, h);
H = Sint*impgen(0);
D12 = D12op*H;
```

where `H` represents the ZOH. Finally the formula (1) is implemented as follows:

```
P = append(B1, [C1, D12]');
Q = [eye(mc1), -D11'; ...
    -D11, eye(pc1)];
R = [zeros(n), A, B2; ...
    [A, B2]', zeros(n+m2)] + P/Q*P';
```

where `append()` is the function for block diagonal augmentation, and `R` is a matrix and the result of the RHS of (1). We confirm that readability of the program with the developed library. We also emphasize that the last syntax is applicable even if S is not an ideal sampler and/or H is not a ZOH just by changing the definitions of variables.

V. CONCLUDING REMARKS

We have implemented a library for SSBCs on MATLAB for development of a CACSD package for sampled-data systems. We have shown that we can handle a wide variety of setups of sampled-data systems including multirate control and input/output time-delay in a unified way by using the SSBC representation. Based on the developed library with object-oriented programming feature, it is easy to implement analysis and design algorithms of sampled-data systems with high readability, and a CACSD package for analysis and design of sampled-data system is under development.

ACKNOWLEDGMENT

The author would like to appreciate to Profs. Mirkin and Hara for valuable comments.

APPENDIX

PROOF OF PROPERTY 3

Suppose that (5) is well-posed. From [8], y is given by

$$y(t) = \int_0^h K(t,s)u(s)\,\mathrm{d}s + Du(t)$$

where

$$K(t,s) := \begin{cases} Ce^{At}\Xi^{-1}\Omega e^{-As}B & 0 \le s < t \le h \\ -Ce^{At}\Xi^{-1}\Upsilon e^{A(h-s)}B & 0 \le t < s \le h \end{cases}.$$

Hence Property 3 follows.

REFERENCES

[1] B. Bamieh and J. B. Pearson, "A general framework for linear periodic systems with application to $\mathscr{H}^\infty$ sampled-data control," IEEE Trans. on Automat. Contr., vol. 37, no. 4, pp. 418–435, 1992.

[2] T. Chen and B. A. Francis: Optimal Sampled-Data Control Systems, Springer, 1995.

[3] H. Fujioka, Y. Yamamoto, and S. Hara, "Sampled-Data Control Toolbox: A software package via object-oriented programming," in Proc. of 1999 IEEE Int. Symp. on CACSD, pp. 404–409, 1999.

[4] S. Hara, H. Fujioka, and P. T. Kabamba, "A hybrid state-space approach to sampled-data feedback control," Linear Algebra and Its Applications, vol. 205–206, pp. 675–712, 1994.

[5] S. Hara, Y. Yamamoto, H. Fujioka, and A. Takeda, "$\mathscr{H}$SYS Module Manual," Sumisho Electronics, Co., 1994.

[6] Y. Hayakawa, S. Hara, and Y. Yamamoto: "H_∞ type problem for sampled-data control systems — A solution via minimum energy characterization," IEEE Trans. Automat. Contr., vol. 39, pp. 2278–2284, 1994.

[7] P. T. Kabamba and S. Hara: "Worst case analysis and design of sampled data control systems," IEEE Trans. Automat. Contr., vol. 38, pp. 1337–1357, 1993.

[8] L. Mirkin and Z. J. Palmor, "A new representation of the parameters of lifted systems," IEEE Trans. Automat. Contr., vol. 44, pp. 833–840, 1999.

[9] Y. Yamamoto: "A function space approach to sampled-data control systems and tracking problems," IEEE Trans. on Automat. Contr., vol. 39, pp. 703–712, 1994.