JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 15, No. 5, September-October 1992

New Multirate Sampled-Data Control Law Structure and Synthesis Algorithm

Martin C. Berg* and Gregory S. Mason†
University of Washington, Seattle, Washington 98195
and
Gen-Sheng Yang‡
Chung-Shan Institute of Science and Technology, Lung-Tan, Tao-Yuan, Taiwan, Republic of China

A new multirate sampled-data control law structure is defined and a new parameter-optimzation-based synthesis algorithm for that structure is introduced. The synthesis algorithm can be applied to multirate, multiple-input/multiple-output, sampled-data control laws having a prescribed dynamic order and structure, and a priori specified sampling/update rates for all sensors, processor states, and control inputs. The synthesis algorithm is applied to design two-input, two-output tip postion controllers of various dynamic orders for a sixth-order, two-link robot arm model.

I. Introduction

EVEN in this age of fast, low-cost microprocessors, there remain several important motivations for multirate sampling in sampled-data control systems. Multirate sampling provides the opportunity to allocate sampling rates, and thus real-time computing power, more efficiently. In two-time-scale control problems, for example, multirate sampling allows slow sampling in control loops associated with low-bandwidth control functions to be traded for fast sampling in control loops associated with high-bandwidth control functions.

The costs of analog-to-digital and digital-to-analog converters are also computation-rate dependent. Multirate sampling thus provides another opportunity to reduce hardware costs in that the compution rates required of analog-to-digital and digital-to-analog converters frequently depend on their sampling rates. Multirate sampling can occasionally even be used to reduce the total number analog-to-digital and/or digital-to-analog converters required by a system, by sample-dependent scheduling of multiple conversion tasks to a lesser number of conversion devices.

A third "motivation" for multirate sampling is becoming increasingly important: Sometimes multirate sampling is the *only* choice. This situation can arise when an a priori decision has been made to include in a system a sensor that provides only a discrete-time signal at a fixed sampling rate. A head position control system for a computer disk drive is a good example of such a system. The disk head, which is suspended on top of the rotating disk, includes a sensor that reads the head position directly from certain diametrically spaced segments on the disk. The sensor's sampling rate is thus *fixed* by the disk's rotation speed. To increase the control bandwidth beyond that dictated by that sampling rate, a second, faster-rate sensor must be added.

A key point often ignored by developers of multirate control law synthesis methods is that the motivations for multirate sampling also dictate certain flexibilities that any multirate control law synthesis method must have to meet the needs of engineering practice. Specifically, such a method must allow the sampling rates for all sensors, the update rates for all processor states, and the update rates for all actuators to be specified independently. The one generally accepted restriction regarding these rates is that the ratio of all combinations of sampling, update, and delay periods be rational, so that the complete sampling/update schedule is periodic. (We assume that all sampling and update events are synchronized to the same clock. The asynchronous case is treated elsewhere.[1])

Timelines representing such a sampling/update schedule are shown in Fig. 1. We define the basic time period (BTP) of such a schedule as the least common multiple of all of its sampling, update, and delay periods. The BTP is the period of repetition of the schedule. We define the shortest time period (STP) of such a schedule as the greatest common divisor of all of its sampling and update periods. We use the symbol P to represent the (integer) number of STPs per BTP and frequently use a double-indexing scheme for the independent variable so that, for example, $x(m,n)$ represents x at the start of the $(n+1)$th STP of the $(m+1)$th BTP, for $m=0,1,\ldots,$ and $n=0,\ldots,P-1$.

There are five well-recognized methods for synthesizing multirate control laws: 1) successive loop closures, 2) pole placement, 3) the singular perturbation method, 4) the linear quadratic Gaussian (LQG) method, and 5) parameter optimization methods. The successive loop closures method[2] is arguably the most important because it is the only one of the five that is widely used in industry. The advantages of the

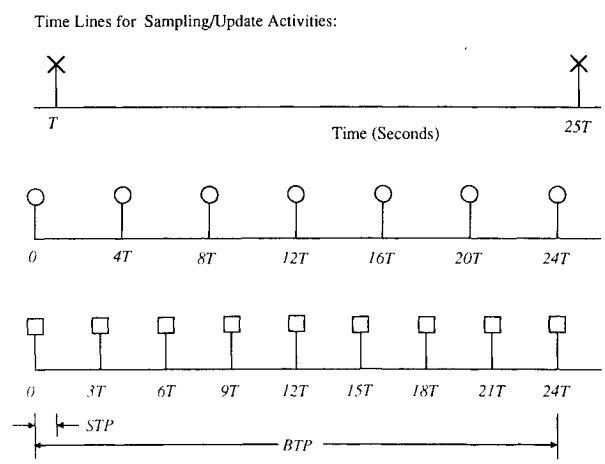


Fig. 1 Example of multirate sampling/update schedule.

Received May 3, 1991; revision received Sept. 24, 1991; accepted for publication Oct. 9, 1991.

*Assistant Professor of Mechanical Engineering, Department of Mechanical Engineering, FU-10. Member AIAA.
†Research Assistant, Department of Mechanical Engineering, FU-10.
‡Associate Scientist.

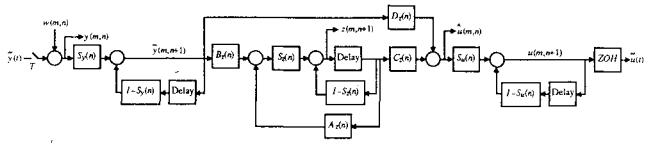


Fig. 2 Multirate sampled-data control law structure.

successive loop closures method are that it requires no fundamentally new multirate synthesis techniques and that the sampling/update rate for each control loop can be specified independently. The problem with successive loop closures is that its one-loop-at-a-time approach cannot fully account for all dynamic coupling between control loops.

Pole placement for multirate systems has received recent attention in the wake of reports on the capacities of periodically time-varying output feedback controllers to place closed-loop poles.[3-7] In Ref. 3, for example, it is shown that, given any controllable and observable continuous-time plant with m inputs, it is always possible to construct a periodically time-varying, pure-gain, output feedback control law that places the closed-loop poles arbitrarily, provided that the outputs are all sampled at a suitable chosen single sampling rate $1/T_0$ and that the inputs are updated at the rates $N_1/T_0, \ldots, N_m/T_0$, where N_i are certain positive integers.

The problem with pole placement for multirate systems is the same as that for pole placement for single-rate systems: determining where the closed-loop poles should be placed. It is an especially difficult problem in the multirate case because multirate systems are periodically time varying.[2,8] That periodicity implies that the eigenstructure of a multirate system can only be defined based on its BTP-to-BTP dynamics (which are time invariant). Desirable closed-loop poles for a multirate system are thus difficult to determine, in general, because the BTP of a multirate system is typically longer than the characteristic times of many of its faster dynamics.

Singular perturbation control law synthesis methods were first developed for continuous-time systems to take advantage of the multiple-time-scale dynamics that often occur in control systems.[9-15] It would seem that an extension to multirate sampled-data control systems should follow naturally, given that a principal motivation for multirate sampling has always been to take advantage of those same multiple time scales, but that has not been the case in practice.

The problem with the singular perturbation method for multirate control law synthesis is the method's dependence on a coordinate transformation to separate the full control law synthesis problem into two (or more) dynamically decoupled control law synthesis problems of different time scales. The state coordinates of the plant model are easily decoupled because they only represent the plant model's *internal* dynamics. The input and output coordinates cannot be so manipulated, however, because they represent the plant model's *external* sensor and actuator signals, the elements of which are, in the multirate case, sampled and updated, respectively, at different rates. Consequently, every control input vector element and every sensor output vector element remains coupled to every state coordinate so that, just as with successive loop closures, when the control laws for the different time scale state vector components are synthesized, all dynamic coupling between the different control loops cannot be accounted for.

Various schemes have been developed to circumvent this difficulty. None have been completely successful. In Ref. 13, for example, a state feedback control law is synthesized by the singular perturbation method, and the lack of completely decoupling transformations gives rise to a requirement for the slow component of the plant state vector to be estimated between slow sampler updates and a requirement for every control input to be updated at every sampling/update instant.

The advantage of the LQG method for multirate control law synthesis is that the control laws for all control loops are synthesized simultaneously, taking into account all dynamic coupling between control loops.[2,16-18] The disadvantages are the same as those for the LQG method for continuous-time control law synthesis: Practical performance and stability robustness objectives are usually difficult to achieve via the minimization of a quadratic performance index, and the resulting control laws tend to be unnecessarily complex. In addition, LQG control laws are less desirable in the multirate compared to the single-rate case because the gains for the multirate Kalman filter and for the LQG optimal full state feedback control law are periodically time varying.[2] In short, LQG multirate control laws can provide useful benchmarks for performance comparisons, but they are generally not practical for applications.

Parameter optimization methods for multirate control law synthesis combine the principal advantages of the LQG and the successive loop closures synthesis methods.[2,19] Like the LQG method, parameter optimization methods simultaneously account for all dynamic coupling between control loops. Like the successive loop closures method, parameter optimization methods allow the synthesis of multirate control laws of arbitrary structure and dynamic order. The typical parameter optimization method requires that the control law structure and its parameters to be optimized be prescribed. A numerical search is used to optimize those parameters such that a quadratic performance index is minimized, possibly subject to constraints on those parameters. The disadvantage of parameter optimization methods is that they inevitably require a numerical search.

A new parameter optimization method for synthesizing multirate control laws is described in Sec. III. This method is the second generation of the method described in Refs. 2 and 20. Unlike its predecessor, which accommodates only partial state feedback control laws, this new method accommodates a general, dynamic, multiple-input/multiple-output control law structure. This new control law structure is described in Sec. II. Section IV describes an application of this new method to the design of a tip position control system for a two-link robot arm model. Conclusions are given in Sec. V.

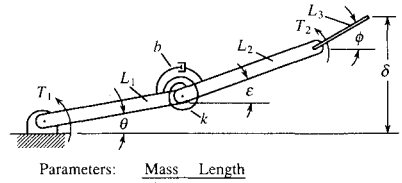
II. Control Law Structure

This section describes the multirate sampled-data control law structure in Fig. 2. In Fig. 2, $\bar{y}$ is the noise-free, continuous-time sensor signal, v is the discrete-time sensor noise signal, and $\bar{u}$ is the continuous-time control signal. The one sampler in Fig. 2 operates at the sampling rate $1/T$, where T is the STP of the system's complete sampling/update schedule. The delay blocks are one-STP delays. The ZOH block is a zero-order hold.

The sensor sample-and-hold dynamics are represented by

$$\bar{y}(m,n+1) = \left[I - S_y(n)\right]\bar{y}(m,n) + S_y(n)y(m,n) \qquad (1)$$

where $\bar{y}$ is the sensor signal hold state vector. The matrix $S_y(n)$ is the sensor switching matrix for the $(n+1)$th STP. We define a switching matrix as a diagonal matrix with 1 or 0 at every diagonal position. If the ith diagonal element of $S_y(n)$ is 1, the



Parameters:

	Mass	Length	
L_1	0.5 kg	0.5 m	
L_2	0.5 kg	0.5 m	$k = 37.33$ N/rad
L_3	0.04 kg	0.2 m	$b = 0.012$ N·s/m

The natural frequency of the vibration mode is 10 hz.
Inputs: Torques T_1 and T_2
Outputs: θ and δ

Fig. 3 Two-link robot arm.

continuous-time signal form the ith sensor is sampled at the start of the $(n+1)$th STP of every BTP, and that sampled value is immediately stored as the ith element of $\bar{y}$; otherwise, the same element of $\bar{y}$ is held at those instants. The key point is that $\bar{y}$ always contains the most recent sampled sensor data.

The processor dynamics are represented by

$$z(m,n+1) = [I - S_z(n)]z(m,n) + S_z(n)\Big\{A_z(n)z(m,n) + B_z(n)\Big\{[I - S_y(n)]\bar{y}(m,n) + S_y(n)y(m,n)\Big\}\Big\} \quad (2)$$

$$\hat{u}(m,n) = C_z(n)z(m,n) + D_z(n)\Big\{[I - S_y(n)]\bar{y}(m,n) + S_y(n)y(m,n)\Big\} \quad (3)$$

where z is the processor state vector, and $\hat{u}$ is the processor output vector. The matrix $S_z(n)$ is the processor state switching matrix. If the ith diagonal element of $S_z(n)$ is 1, the ith processor state is updated at the start of the $(n+1)$th STP of every BTP; otherwise, the same element of z is held at those instants. The matrices $A_z(n)$, $B_z(n)$, $C_z(n)$, and $D_z(n)$ are the processor state model matrices, whose determination constitutes the control law synthesis problem. Note that a nonzero $D_z(n)$ results in direct feedthrough of sensor data to $\hat{u}(m,n)$.

The control signal update-and-hold dynamics are represented by

$$\bar{u}(m,n+1) = [I - S_u(n)]\bar{u}(m,n) + S_u(n)\hat{u}(m,n) \quad (4)$$

where $\bar{u}$ is the control signal hold state vector. The matrix $S_u(n)$ is the control signal switching matrix. If the ith diagonal element of $S_u(n)$ is 1, the ith element of $\bar{u}$ is updated at the start of the $(n+1)$th STP of every BTP; otherwise, the same element of $\bar{u}$ is held at those instants.

Finally, the continous-time control signal $\tilde{u}$ is generated by

$$\tilde{u}(t) = [I - S_u(n)]\bar{u}(m,n) + S_u(n)\hat{u}(m,n) \quad (5)$$

for all $t \in [(mP+n)T, (mP+n+1)T)$.

The advantage of the control law structure of Eqs. (1-5) is that it can be used to represent most sampled-data control law structures of practical interest. However, its form is not standard. Straightforward algebra, applied to Eqs. (1-5), yields the following more standard form:

$$c(m,n+1) = A_c(n)c(m,n) + B_c(n)y(m,n) \quad (6)$$

$$u(m,n) = C_c(n)c(m,n) + D_c(n)y(m,n) \quad (7)$$

where

$$c(m,n) = [z^T(m,n)\,\bar{y}^T(m,n)\,\bar{u}^T(m,n)]^T \quad (8)$$

$$A_c(n) = \begin{bmatrix} [I - S_z(n)] + S_z(n)A_z(n) & S_z(n)B_z(n)[I - S_y(n)] & 0 \\ 0 & I - S_y(n) & 0 \\ S_u(n)C_z(n) & S_u(n)D_z(n)[I - S_y(n)] & I - S_u(n) \end{bmatrix} \quad (9)$$

$$B_c(n) = \begin{bmatrix} S_z(n) & B_z(n) & S_y(n) \\ & S_y(n) & \\ S_u(n) & D_z(n) & S_y(n) \end{bmatrix} \quad (10)$$

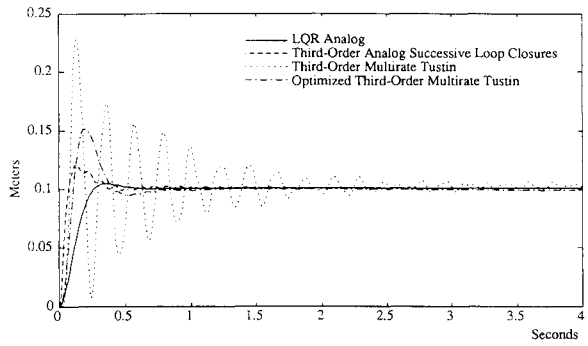
$$C_c(n) = \begin{bmatrix} S_u(n)C_z(n) & S_u(n)D_z(n)[I - S_y(n)] & I - S_u(n) \end{bmatrix} \quad (11)$$

$$D_c(n) = [S_u(n)D_z(n)S_y(n)] \quad (12)$$

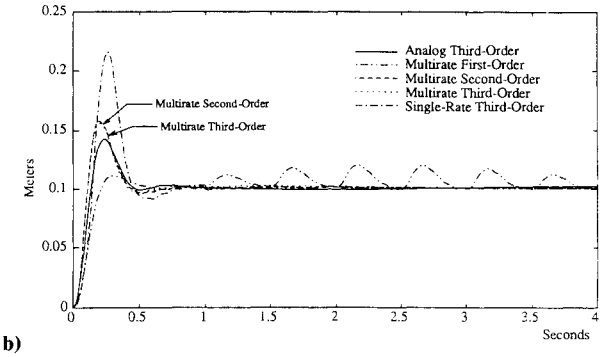
with

$$\tilde{u}(t) = u(m,n) \quad (13)$$

for all $t \in [(mP+n)T, (mP+n+1)T)$.



a)



b)

Fig. 4 Tip position δ responses to tip position step command.

III. Parameter Optimization Method

This section describes a parameter optimization control law synthesis method for the control law structure of Sec. II. It is a generalization of the similar method for state feedback control laws described in Refs. 2 and 20 and also incorporates the multiple plant condition design for robustness ideas of Ref. 21. The approach involves a numerical search to determine the processor matrices, $A_z(n)$, $B_z(n)$, $C_z(n)$, and $D_z z(n)$, for $n = 0, \ldots, P-1$, such that a quadratic performance index is minimized. That approach has been criticized in the past because of 1) the difficulties of achieving practical performance and stability robustness objectives via the minimization of a quadratic performance index and 2) difficulties related to the convergence of the numerical search.

The proposed method addresses those criticisms in several ways. First, to enable synthesis for robustness to plant parameter variations, the performance index is defined over multiple plant conditions. That simple idea has been a key to the success of the popular Sandy[21-26] algorithm for synthesizing robust continuous-time control laws. Second, to improve the convergence of the numerical search, the performance index and its gradients with respect to the control law parameters are calculated exactly, at every iteration, using closed-form expressions. Third, so that a stabilizing initial guess for the control law is not required, and to eliminate problems with destabilizing control laws encountered during the search, a finite-time performance index is used. Finally, to lessen the difficulties of achieving practical performance and stability robustness objectives via the minimization of a quadratic performance index, the method allows linear and nonlinear constraints to be imposed on the control law parameters.

The continuous-time plant dynamics at plant condition i are assumed to be represented by

$$\dot{\tilde{p}}^{(i)}(t) = \tilde{A}_p^{(i)}\tilde{p}^{(i)}(t) + \tilde{B}_{pu}^{(i)}\tilde{u}^{(i)}(t) + \tilde{B}_{pw}^{(i)}\tilde{w}^{(i)}(t) \tag{14}$$

$$\tilde{y}^{(i)}(t) = \tilde{C}_p^{(i)}\tilde{p}^{(i)}(t) \tag{15}$$

where $\tilde{p}^{(i)}$ is the plant state vector, $\tilde{u}^{(i)}$ is the control input vector, $\tilde{y}^{(i)}$ is the sensor output vector, and $\tilde{w}^{(i)}$ is a stationary, zero-mean, Gaussian white noise input vector of known power spectral density.

The performance index is assumed to be given by

$$J(t_f) = \sum_{i=1}^{N_p} E\left\{\frac{1}{2t_f}\int_0^{t_f}\begin{bmatrix}\tilde{p}^{(i)}(t)\\ \tilde{u}^{(i)}(t)\end{bmatrix}^T\begin{bmatrix}\tilde{Q}^{(i)} & 0\\ 0 & \tilde{R}^{(i)}\end{bmatrix}\begin{bmatrix}\tilde{p}^{(i)}(t)\\ \tilde{u}^{(i)}(t)\end{bmatrix}\mathrm{d}t\right\} \tag{16}$$

where N_p is the number of plant conditions; E is the expected value operator; t_f is the final time, which must be a multiple of the BTP of the closed-loop system's sampling/update schedule; and $\tilde{Q}^{(i)}$ and $\tilde{R}^{(i)}$ are the state and control weighting matrices, respectively, for the ith plant condition and must be non-negative-definite.

Based on the description of the continuous-time plant dynamics in Eqs. (14) and (15), a description of the closed-loop system's sampling/update schedule, the performance index in Eq. (16), and the control law structure in Eqs. (6–13), closed-form expressions for the performance index $J(t_f)$ and for its gradients with respect to the processor matrices $A_z(n)$, $B_z(n)$, $C_z(n)$, and $D_z(n)$, for $n = 0, \ldots, P-1$, are derived in Refs. 19 and 27. Those expressions are lengthy and will not be repeated here. The key points are that those performance index and gradient expressions are closed form, and that the number of computations required for their evaluation is independent of t_f. For those expressions to be valid, the single restriction is that the BTP-to-BTP state transition matrix for the closed-loop system must be diagonalizable.[19] That is not a serious restriction, however, because that matrix is rarely nondiagonalizable in practice.

Thus far nothing has been said about synthesizing other than periodically time-varying control laws. To that end, the performance index and gradient derivations in Refs. 19 and 27 assume that the processor matrices are constrained to satisfy

$$\begin{bmatrix}D_z(n) & C_z(n)\\ B_z(n) & A_z(n)\end{bmatrix} = \sum_{r=0}^{M-1}\alpha(n,r)\begin{bmatrix}\bar{D}_z(r) & \bar{C}_z(r)\\ \bar{B}_z(r) & \bar{A}_z(r)\end{bmatrix} \tag{17}$$

with $M \in \{1, \ldots, P\}$ and with the α functions constrained to satisfy

$$\alpha(n,p)\alpha(n,q) = \begin{cases}1 & \text{if} \quad p = q\\ 0 & \text{if} \quad p \neq q\end{cases} \tag{18}$$

Equations (17) and (18) constrain the number of different sets of processor matrices to M. The function $\alpha(n,r)$ determines which set of processor matrices is active at the $(n+1)$th STP of every BTP. Equation (18) guarantees that only one set of processor matrices is active per STP.

Based on the description of the continuous-time plant dynamics in Eqs. (14) and (15), a description of the closed-loop system's sampling/update schedule, the performance index in Eq. (16), the control law structure in Eqs. (6–13), the constraint relations in Eqs. (17) and (18), and the closed-form expressions for the performance index $J(t_f)$, and for its gradients with respect to the processor matrices $\bar{A}_z(r)$, $\bar{B}_z(r)$, $\bar{C}_z(r)$, and $\bar{D}_z(r)$, for $r = 0, \ldots, M-1$, in Refs. 19 and 27, we have developed and programmed an algorithm to determine numerically a set of processor matrices that minimizes $J(t_f)$. A numerical search is used to determine the processor matrices $\bar{A}_z(r)$, $\bar{B}_z(r)$, $\bar{C}_z(r)$, and $\bar{D}_z(r)$, for $r = 0, \ldots, M-1$, given an initial guess for those matrices. The NPSOL nonlinear programming algorithm is used for the numerical search. NPSOL[28] is a powerful nonlinear programming package with good convergence properties as a result of its use of exact performance index and gradient evaluations at every iteration. In addition, NPSOL accommodates linear and/or nonlinear constraints on the independent variables. This means that linear and/or nonlinear constraints on the control law parameters can be combined with the usual performance index minimization objectives to achieve practical performance and stability robustness objectives.

Additional features of our synthesis algorithm include automatic discretization of the continuous-time plant model and automatic discretization of the continuous-time performance index.[27] Those features are important because they effectively decouple the sampling/update rates selection problem from the problem of determining a suitable performance index. Thus, a performance index can be determined first, based on a continuous-time design, and our algorithm can subsequently be used to determine a multirate sampled-data control law that minimizes the same performance index.

In summary, the following inputs are required to synthesize a multirate control law by this method:

1) a state model description of the continuous-time plant dynamics at each of the N_p plant conditions;

2) state and control weighting matrices for the performance index at each of the N_p plant conditions;

3) the t_f for the performance index;

4) the power spectral density of the continuous-time white process noise at each of the N_p plant conditions;

5) a complete description of the complete system's sampling/update schedule;

6) the integer M and the α functions that constrain the periodicity of the processor matrices via Eqs. (17) and (18);

7) the desired dynamic order and structure for the processor matrices;

8) the covariance matrix for the discrete-time sensor noise at each of the N_p plant conditions;

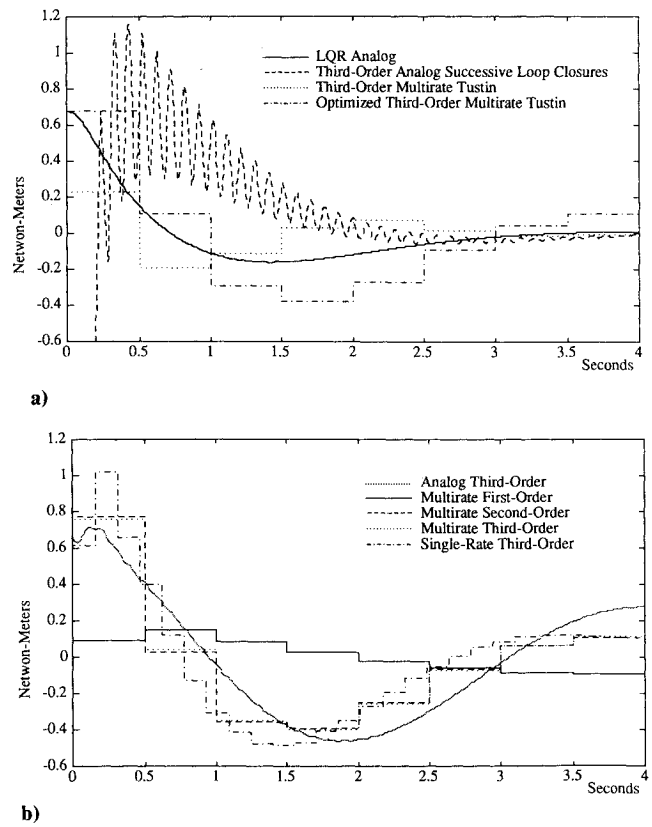


Fig. 5 Control torque T_1 responses to tip position step command.

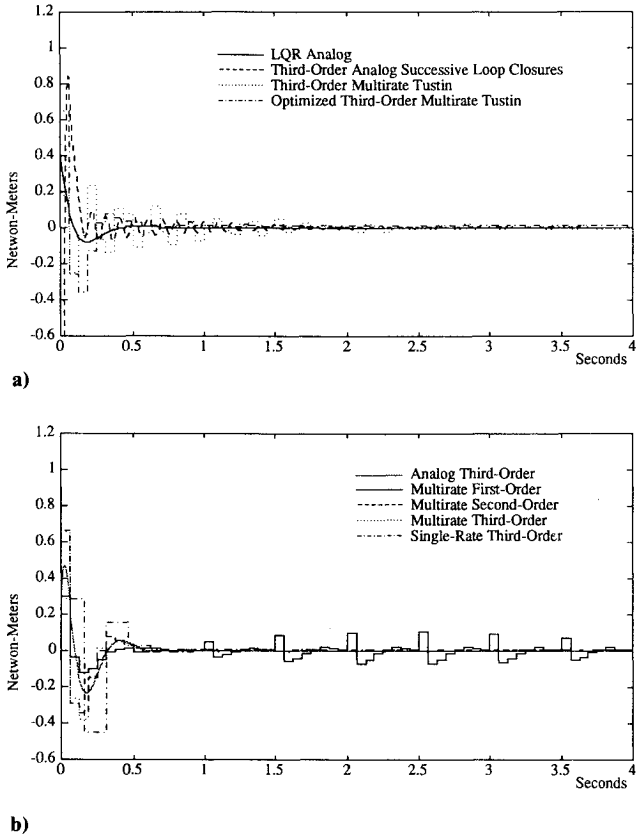


Fig. 6 Control torque T_2 responses to tip position step command.

9) a complete description of all additional linear and nonlinear constraints to be imposed on the elements of the processor matrices; and

10) an initial guess for the processor matrices.

A particular feature of this new algorithm is that it does not require a stabilizing initial guess for the control law. This is a consequence of its use of a finite-time performance index. The finite time ensures that the performance index and its gradients will be finite whether or not the closed-loop system is stable. A disadvantage of the finite time is that a steady-state solution, i.e., for

$$J_{ss} \triangleq \lim_{t_f \to \infty} J(t_f) \tag{19}$$

cannot be obtained directly. A steady-state solution is easily obtained in practice, however, by choosing a t_f that is large compared to all characteristic times of the closed-loop system. Because the number of computations required to evaluate the performance index and gradients with respect to the control law parameters does not depend upon t_f, this can be done without penalty in terms of the computation time required for the numerical search.

In practice, because digital computers cannot accommodate arbitrarily large finite numbers, a steady-state solution usually cannot be obtained immediately by simply setting t_f to a large value. Instead, it is usually necessary to complete first (when the guess for the control law parameters is poor) an optimization for a small t_f and to then reoptimize, for larger and larger t_f, until t_f becomes large compared to the characteristic times of all closed-loop poles.

The final key issue regarding this new synthesis method concerns the specification of the structure of the processor matrices. The key point is that, for an efficient numerical search, there should be imposed, on the processor matrices, a structure that defines an independent set of free parameters with respect to the control law's input/output dynamics. With regard to the processor dynamics in Eqs. (2) and (3), with the constraints in Eqs. (17) and (18) in effect, it is straightforward to see that the complete set of the elements of $\bar{A}_z(r)$, $\bar{B}_z(r)$, $\bar{C}_z(r)$, and $\bar{D}_z(r)$, for $r = 0, \ldots, M-1$, does not constitute such an independent set. For example, it is straightforward to see that an arbitrary change in one element of $\bar{B}_z(0)$ can be compensated for by changes to the elements of $\bar{C}_z(0)$ and to the other elements of $\bar{B}_z(0)$ such that the processor's input/output dynamics are unchanged.

Thus, additional constraints must be imposed on the elements of the processor matrices to guarantee an independent set of free parameters for the numerical search. In practice, a suitable set of such constraints can frequently be determined based on "classical" control law structures (see, e.g., the optimized third-order multirate Tustin design example of the following section). More general control law structures can, of course, also be accommodated. What constitutes an optimal general structure is a topic of current research. For the present we have successfully applied the following structure for the particular case where the constraints in Eqs. (17) and (18) are applied with $M = 1$ (the time-invariant case, shown for the n-is-even case):

$$\bar{A}_z(0) = \text{block diag}\left\{\begin{bmatrix} 0 & 1 \\ -\sigma_i^2 - \omega_i^2 & -2\sigma_i \end{bmatrix}\right\} \qquad i = 1, \ldots, n/2 \tag{20}$$

$$\bar{B}_z(0) = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix} \tag{21}$$

$$\bar{C}_z(0) = \begin{bmatrix} 1 & \cdots & 1 \\ c_{21} & \cdots & c_{2n} \\ \vdots & \ddots & \vdots \\ c_{p1} & \cdots & c_{pm} \end{bmatrix} \tag{22}$$

$$\bar{D}_z(0) = \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \ddots & \vdots \\ d_{p1} & \cdots & d_{pm} \end{bmatrix} \tag{23}$$

The appendix shows that the $(m+p)n + mp$ σ_i, ω_i, b_{ij}, c_{ij}, and d_{ij} parameters of this structure constitute an independent set with respect to the control law's input/output dynamics (for any nontrivial sampling/update schedule), provided that no eigenvalues, $\sigma_i \pm j\omega_i$, of the processor dynamics are repeated.

IV. Examples

This section describes the design of a tip position control system for the mathematical model of the planar two-link robot arm system represented in Fig. 3. The first link is long and massive, for large-scale slewing motions. The second link is relatively short and lightweight, so that high-bandwidth control of the arm's tip position can be achieved with a relatively small motor at the second joint. The pin joint, rotational spring, and rotational damper at the midpoint of the first link models flexibility in the first link. The motor torques T_1 and T_2 are the control inputs, and only the joint angle θ and the

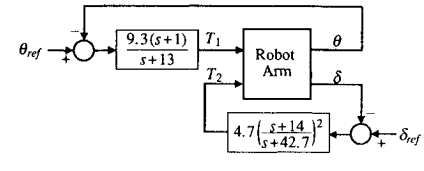


Fig. 7 Third-order analog successive loop closures design.

tip position δ are measured. The linearized dynamic equations for this system for small $\epsilon - \theta$ and small $\phi - \epsilon$ are easily derived. The spring constant k and damping coefficient b values (in Fig. 3) were chosen based on those equations to achieve 1% damping and a 10-Hz natural frequency for the open-loop vibration mode.

Figures 4–6 show the closed-loop arm responses, based on the linearized arm dynamics, to a step change in the commanded tip postion with nine different control laws. The tip postion (δ) responses are shown in Fig. 4. The simultaneous control torque (T_1 and T_2) responses are shown in Figs. 5 and 6. The nine control laws are briefly described as follows:

LQR analog: The continuous-time LQR (full state feedback) control law that minimizes

$$J = \lim_{t_f \to \infty} \frac{1}{2t_f} \int_0^{t_f} \beta^2 \left[\theta^2 + (\rho\delta)^2\right] + \left(\frac{T_1}{T_{1\,\max}}\right)^2 + \left(\frac{T_2}{T_{2\,\max}}\right)^2 dt \quad (24)$$

with

$$\beta = 32 \text{ rad}^{-1} \quad (25)$$

$$\rho = 14 \text{ rad/m} \quad (26)$$

$$T_{2\,\max} = 0.00335 \text{ N}\cdot\text{m} \quad (27)$$

$$T_{1\,\max}/T_{2\,\max} = 8 \quad (28)$$

The $T_{2\,\max}$ value is the T_2 torque that achieves $\dot{\phi} = 2\pi$ rad/s in 1 s with $\dot{\phi}(0) = 0$ and $\theta(t) \equiv \epsilon(t) \equiv 0$. The $T_{1\,\max}/T_{2\,\max}$ value is a typical ratio of peak motor torques at the respective joints. The β and ρ values were chosen by trial and error to achieve the closed-loop poles in Table 1. Note that the ratio of the characteristic frequencies of the rigid-body closed-loop pole pairs is 8 and that the characteristic frequency of the faster rigid-body closed-loop pole pair is a factor of 5 less than the characteristic frequency of the closed-loop vibration mode.

Third-Order Analog Successive Loop Closures

This is the third-order, continuous-time, successive loop closures control law in Fig. 7, which consists of a single lead compensator in the θ to T_1 loop and twin, cascaded lead compensators in the δ to T_2 loop. The closed-loop poles for this design are in Table 2. Note that the rigid-body and vibration mode closed-loop poles match those of the LQR analog design.

Third-Order Multirate Tustin

This is a multirate sampled-data approximation to the third-order analog successive loop closures design obtained via Tustin's approximations of the continuous-time transfer functions in Fig. 7. The sampling/update rates in the θ to T_1 and δ to T_2 loops are 2 and 16 samples/updates per second, respectively, which are eight times the characteristic frequencies (in cycles per second) of the slow and fast rigid-body closed-loop pole pairs, respectively, from the third-order analog successive loop closures design.

Optimized Third-Order Multirate Tustin

This is the same as the third-order multirate Tustin design, but with the lead compensator gain, zero, and pole locations optimized, by the parameter optimization control law synthesis method of Sec. III, to minimize the same performance index as in the LQR analog design. To synthesize this control law, continuous-time process noise and discrete-time sensor noise inputs were added to the robot arm model. The former inputs were taken to be white noise disturbance torques w_1 and w_2, coincident with the respective control torques, with

$$E\left\{\begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \begin{bmatrix} w_1(\tau) & w_2(\tau) \end{bmatrix}\right\} = \begin{bmatrix} 4.9\times10^{-5} & 0 \\ 0 & 1.6\times10^{-5} \end{bmatrix} \delta(t-\tau) \quad (29)$$

Table 1 LQR analog design closed-loop poles

	Closed-loop pole, s^{-1}	Damping ratio	Characteristic frequency, Hz
Rigid-body mode	$-1.10 \pm j\,1.10$	0.71	0.25
Rigid-body mode	$-8.81 \pm j\,8.83$	0.71	2.0
Vibration mode	$-0.649 \pm j\,62.8$	0.01	10

Table 2 Third-order successive loop closures design closed-loop poles

	Closed-loop pole, s^{-1}	Damping ratio	Characteristic frequency, Hz
Rigid-body mode	$-1.10 \pm j\,1.11$	0.71	0.25
Rigid-body mode	$-8.88 \pm j\,8.84$	0.71	2.0
Vibration mode	$-1.35 \pm j\,63.9$	0.02	10
Compensator mode	-10.5	—	1.7
Compensator mode	$-33.2 \pm j\,34.0$	0.70	7.6

where δ is the Dirac delta function. The latter inputs were taken to be stationary, purely random sequences, v_1 and v_2, for the θ and δ measurements, respectively, with

$$E\left\{\begin{bmatrix} v_1(m,n) \\ v_2(m,n) \end{bmatrix} \begin{bmatrix} v_1(m,n) & v_2(m,n) \end{bmatrix}\right\} = \begin{bmatrix} 8.1\times10^{-5} & 0 \\ 0 & 1\times10^{-4} \end{bmatrix} \quad (30)$$

Multirate Third Order

This is the same as the optimized third-order multirate Tustin design, but uses the third-order, generalized, time-invariant structure in Eqs. (20–23) for the processor matrices. Like the optimized third-order multirate Tustin design, two of the processor states are updated at the faster sampling/update rate, and the third is updated at the slower sampling/update rate.

Multirate Second-Order

This is the same as the multirate third-order design, but uses the second-order, generalized, time-invariant structure in Eqs. (20–23) for the processor matrices. One of the processor states is updated at the faster sampling/update rate, and the other is updated at the slower sampling/update rate.

Multirate First-Order

This is the same as the multirate third-order design, but uses the first-order, generalized, time-invariant structure of Eqs. (20–23) for the processor matrices. The one processor state is updated at the faster sampling/update rate.

Single-Rate Third Order

This is the same as the multirate third-order design, but single rate, with the single sampling/update rate chosen to yield the same number of real-time computations per unit time as the multirate third-order design.

Analog Third Order

This is continuous-time equivalent to the multirate and single-rate third-order designs. The processor matrices have the same structure as in the multirate and single-rate third-order designs. The control law was synthesized using the Sandy algorithm[21] to minimize the same performance index as in the multirate and single-rate third-order designs.

The LQR analog responses in Figs. 4a, 5a, and 6a constitute the optimal responses for the performance index in Eq. (24), assuming full state feedback, no process or sensor noise, and

infinitely fast sampling. The third-order analog successive loop closures responses in the same figures have low tip position overshoot, but also include a relatively large contribution from the vibration mode (see especially Figs. 5a and 6a).

The third-order multirate Tustin responses in Figs. 4a, 5a, and 6 are unacceptable. This is a result of the Tustin discretization with the 2 and 16 samples/updates per second rates in the θ to T_1 and δ to T_2 control loops, respectively; these rates are very slow compared to the 10-Hz vibration mode frequency. Analog prefilters could be used to improve the performance of this design by filtering out the 10-Hz components from the θ and δ sensor signals.

The optimized third-order multirate Tustin responses in Figs. 4a, 5a, and 6a suggest, however, that analog prefiltering of the θ and δ sensor signals is unnecessary. The optimized third-order multirate Tustin responses in those figures are acceptable and demonstrate that the parameter optimization control law synthesis algorithm of Sec. III can be effectively used to optimize the parameters of classically structured control laws.

The multirate third-order, second-order, and first-order responses in Figs. 4b, 5b, and 6b demonstrate that the same parameter optimization control law synthesis algorithm can be used to synthesize multirate control laws having a prescribed dynamic order and a prescribed, but general, structure, with a priori specified sampling/update rates for all sensors, processor states, and control inputs.

The single-rate third-order and analog third-order responses in the same figures put the multirate responses in perspective. The single-rate third-order control law is the single-rate equivalent to the multirate third-order control law because 1) it was synthesized to minimize the same performance index, using the same process and sensor noise characteristics; and 2) it requires the same number of computations per unit time for real-time operation. The analog third-order responses in the same figures are the responses that would have been obtained with either the multirate third-order control law or the single-rate third-order control laws if sampling and update rates were not an issue and very fast sampling and update rates were used everywhere.

V. Conclusions

With successive loop closures being a possible exception, the multirate sampled-data control law synthesis methods available today fail to provide the designer with sufficient flexibility to prescribe sensor sampling rates and processor state and control input update rates. A new parameter-optimization-based method for synthesizing multirate control laws of arbitrary dynamic order that provides that flexibility is described in Sec. III. This new method determines, by numerical optimization, the parameters of the new general purpose multiple-input/multiple-output, sampled-data control law structure described in Sec. II to minimize a quadratic performance index, possibly subject to linear and/or nonlinear constraints on those parameters. A stabilizing initial guess for the control law is not required because the performance index is finite time. To enable the synthesis of robust control laws, the performance index can be defined over multiple plant conditions.

An application of this new synthesis method to the design of a tip position control system for a sixth-order, two-link robot arm, described in Sec. IV, confirms that this new synthesis method can be used to synthesize multirate control laws having a prescribed dynamic order and structure, with a priori specified sampling/update rates for all sensors, processor states, and control inputs.

Appendix: Supporting Mathematics

Consider the control law in Eqs. (1-5). Suppose that the constraints in Eqs. (17) and (18), with $M = 1$, are in effect, so that the processor matrices are constrained to be time invariant. Suppose that the processor matrices, $\bar{A}_z(0)$, $\bar{B}_z(0)$, $\bar{C}_z(0)$, and $\bar{D}_z(0)$ are further constrained to have the forms in Eqs. (20-23). Finally, to guarantee a nontrivial sampling/update schedule, suppose that the sensor, processor state, and actuator switching matrices satisfy

$$\det\left[\sum_{n=0}^{P-1} S_y(n)\right] \neq 0 \tag{A1}$$

$$\det\left[\sum_{n=0}^{P-1} S_z(n)\right] \neq 0 \tag{A2}$$

$$\det\left[\sum_{n=0}^{P-1} S_u(n)\right] \neq 0 \tag{A3}$$

We will show that the $(m+p)n+pm$ σ_i, ω, b_{ij}, c_{ij}, and d_{ij} elements of the control law then constitute an independent set with respect to that control law's input/output dynamics if and only if $\bar{A}_z(0)$ has no repeated eigenvalues.

We begin by noting that, with Eqs. (31-33) in effect, it is straightforward to see that the independence in question does not depend whatsoever on the sensor, processor state, or actuator switching matrices. Therefore, we consider only the special case where $S_y(n)$, $S_z(n)$, and $S_u(n)$ $(n = 0, \ldots, P-1)$ are identity matrices. The control law then reduces to

$$z(m,n+1) = \bar{A}_z(0)z(m,n) + \bar{B}_z(0)y(m,n) \tag{A4}$$

$$u(m,n) = \bar{C}_z(0)z(m,n) + \bar{D}_z(0)y(m,n) \tag{A5}$$

where $u(m,n)$ is defined in Eq. (13).

Consider first the control law

$$z(k+1) = Az(k) + By(k) \tag{A6}$$

$$u(k) = Cz(k) + Dy(k) \tag{A7}$$

where

$$z = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} \tag{A8}$$

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \tag{A9}$$

and B, C, and D have the forms in Eqs. (21-23). So that the control law's impulse response will be purely real, the λ_i and the corresponding columns of C and rows of B must be either real or must occur in complex-conjugate pairs, and D must be purely real.

Lemma. The $(m+p)n+pm$ λ_i, b_{ij}, c_{ij}, and d_{ij} parameters (counting a real element as one parameter and a complex conjugate pair of elements as two parameters) of the control law in Eqs. (A6) and (A7) constitute an independent set with respect to that control law's input/output dynamics if and only if $\lambda_i \neq \lambda_j$ for $i \neq j$.

Proof. Consider the related control law

$$z(k+1) = Az(k) + By(k) \tag{A10}$$

$$u(k) = \bar{C}z(k) \tag{A11}$$

with

$$\bar{C} = \begin{bmatrix} \bar{c}_{11} & \cdots & \bar{c}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{c}_{p1} & \cdots & \bar{c}_{pn} \end{bmatrix} \tag{A12}$$

Its input/output dynamics are represented by

$$\bar{H}(z) = \bar{C}(zI - A)^{-1}B = \sum_{i=1}^{n} \frac{\bar{C}_i B_i}{z - \lambda_i} \tag{A13}$$

where $\bar{C}_i$ is the ith column of $\bar{C}$, and B_i is the ith row of B. The λ_i, b_{ij}, and $\bar{c}_{ij}$ parameters of this control law are dependent with respect the control law's input/output dynamics if and only if, for an arbitrary change in one, the others can be changed so that $\bar{H}(z)$ is unchanged.

Case 1: No repeated λ_i. From Eq. (A11), for the case of no repeated λ_i, it is straightforward to see that when one of the λ_i is changed by an arbitrary amount, it will not be possible to change the remaining λ_i, b_{ij}, and $\bar{c}_{ij}$ elements so that $\bar{H}(z)$ is unchanged. However, when b_{ij} is multiplied by a nonzero but otherwise arbitrary α, we can multiply the remaining elements of B_i by α and divide $\bar{C}_j$ by α, so that $\bar{H}(z)$ is unchanged. Thus, the λ_i, b_{ij}, and $\bar{c}_{ij}$ parameters of the control law in Eqs. (A10) and (A11) are dependent with respect to that control law's input/output dynamics.

If, however, one element of every column of $\bar{C}$ is fixed, as is the case in the C matrix of Eq. (22), it is similarly straightforward to see that it will not be possible to compensate for an arbitrary change in any λ_i, b_{ij}, or $\bar{c}_{ij}$ element by changing the remaining λ_i, b_{ij}, and $\bar{c}_{ij}$ elements so that $\bar{H}(z)$ is unchanged. Thus, for the case of no repeated λ_i, with one element of every column of $\bar{C}$ fixed, the λ_i, b_{ij}, and $\bar{c}_{ij}$ parameters of the control law in Eqs. (A10) and (A11) constitute an independent set with respect to that control law's input/output dynamics.

Case 2: Repeated λ_i. Consider again the control law in Eqs. (A10) and (A11), but this time suppose that $\lambda_1 = \lambda_2$. That control law's input/output dynamics are represented by

$$\bar{H}(z) = \bar{C}(zI - A)^{-1}B = \frac{\bar{C}_1 B_1 + \bar{C}_2 B_2}{z - \lambda_1} + \sum_{i=3}^{n} \frac{\bar{C}_i B_i}{z - \lambda_i} \tag{A14}$$

From Eq. (A12), it is straightforward to see that, with or without one element of every column of $\bar{C}$ fixed, the remaining b_{ij} and $\bar{c}_{ij}$ elements of $\bar{C}_1$, B_1, $\bar{C}_2$, and B_2 can be changed to compensate for an arbitrary change in any one element of $\bar{C}_1$, B_1, $\bar{C}_2$, or B_2 so that $\bar{H}(z)$ is unchanged. Thus, for the case of repeated λ_i with or without one element of every column of $\bar{C}$ fixed, the λ_i, b_{ij}, and $\bar{c}_{ij}$ parameters of the control law of Eqs. (A10) and (A11) are dependent with respect to that control law's input/output dynamics.

General case: We conclude that the $(m+p)n$ λ_i, b_{ij}, and c_{ij} parameters of the control law in Eqs. (A6) and (A7), with $D = 0$, constitute an independent set with respect to that control law's input/output dynamics if and only if $\lambda_i \neq \lambda_j$, for $i \neq j$. A nonzero D matrix simply adds pm parameters to that set.

Theorem. For the control law in Eqs. (1–5), with the constraints in Eqs. (17) and (18), with $M = 1$, in effect, so that the processor matrices are constrained to be time invariant; with $\bar{A}_z(0)$, $\bar{B}_z(0)$, $\bar{C}_z(0)$, and $\bar{D}_z(0)$ further constrained to have the forms in Eqs. (20–23); and assuming that the sensor, processor state, and actuator switching matrices satisfy Eqs. (A1–A3); the $(m+p)n + pm$ σ_i, ω_i, b_{ij}, c_{ij}, and d_{ij} elements of that control law constitute an independent set with respect to that control law's input/output dynamics if and only if $\bar{A}_z(0)$ has no repeated eigenvalues.

Proof. The control law in Eqs. (A4) and (A5) has the same number of free parameters as the control law in Eqs. (A6) and (A7), and the two are related by the similarity transformation $x = Mz$, where

$$M = \text{block diag}\left\{ \frac{1}{2j\omega_i} \begin{bmatrix} \sigma_i + j\omega_i & 1 \\ -\sigma_i + j\omega_i & -1 \end{bmatrix}, \quad i = 1, \ldots, n/2 \right\} \tag{A15}$$

with $\sigma_i = \text{Re}(\lambda_i)$ and $\omega_i = \text{Im}(\lambda_i)$.

Acknowledgments

This research was supported by NASA Langley Research Grant NAG-1-1055 and by the government of Taiwan. The authors particularly wish to thank Jarrel R. Elliott and Aaron J. Ostroff of the NASA Langley Research Center for their support.

References

[1]Ritchey, V. S., and Franklin, G. F., "A Stability Criterion for Asynchronous Multirate Linear Systems," *IEEE Transactions on Automatic Control,* Vol. 34, No. 5, 1989, pp. 529–535.

[2]Berg, M. C., Amit, N., and Powell, J. D., "Multirate Digital Control System Design," *IEEE Transactions on Automatic Control,* Vol. 33, No. 12, 1988, pp. 1139–1150.

[3]Araki, M., and Hagiwara, T., "Pole Assignment by Multirate Sampled-Data Output Feedback," *International Journal of Control,* Vol. 44, No. 6, 1986, pp. 1661–1673.

[4]Chammas, A., and Leondes, C., "Pole Assignment by Piecewise Constant Output Feedback," *International Journal of Control,* Vol. 29, No. 1, 1979, pp. 31–38.

[5]Hagiwara, T., and Araki, M., "Design of a Stable State Feedback Controller Based on the Multirate Sampling of the Plant Output," *IEEE Transactions on Automatic Control,* Vol. 33, No. 9, 1988, pp. 812–819.

[6]Hernandez, V., and Urbano, A., "Pole-Assignment for Discrete-Time Linear Periodic Systems," *International Journal of Control,* Vol. 46, No. 2, 1987, pp. 687–697.

[7]Patel, Y., and Patton, R. J., "A Robust Approach to Multirate Controller Design Using Eigenstructure Assignment," *Proceedings of the American Control Conference,* IEEE, Piscataway, NJ, 1990, pp. 945–951.

[8]Araki, M., and Yamamoto, K., "Multivariable Multirate Sampled-Data Systems: State-Space Description, Transfer Characteristics, and Nyquist Criterion," *IEEE Transactions on Automatic Control,* Vol. 31, No. 2, 1986, pp. 145–154.

[9]Saksena, V. R., O'Reilly, J., and Kokotovic, P. V., "Singular Perturbations and Time-Scale Methods in Control Theory: Survey 1976–1983," *Automatica,* Vol. 20, No. 3, 1984, pp. 273–293.

[10]Calise, A. J., Prasad, J. V. R., and Sicilano, B., "Design of Optimal Output Feedback Compensators in Two-Time Scale Systems," *IEEE Transactions on Automatic Control,* Vol. 35, No. 4, 1990, pp. 488–492.

[11]Kando, H., and Iwazumi, T., "Multirate Digital Control Design of an Optimal Regulator via Singular Perturbation Theory," *International Journal of Control,* Vol. 44, No. 6, 1986, pp. 1555–1578.

[12]Lennartson, B., "Multirate Sampled-Data Control of Two-Time-Scale Systems," *IEEE Transactions on Automatic Control,* Vol. 34, No. 6, 1989, pp. 642–644.

[13]Litkouhi, B., and Khalil, H., "Multirate and Composite Control of Two-Time-Scale Discrete-Time Systems," *IEEE Transactions on Automatic Control,* Vol. 30, No. 7, 1985, pp. 645–651.

[14]Litkouhi, B., and Khalil, H., "Infinite-Time Regulators for Singularly Perturbed Difference Equations," *International Journal of Control,* Vol. 39, No. 3, 1984, pp. 587–598.

[15]Naidu, D. S., and Price, D. B., "Time-Scale Synthesis of a Closed-Loop Discrete Optimal Control System," *Journal of Guidance, Control, and Dynamics,* Vol. 10, No. 5, 1987, pp. 417–421.

[16]Amit, N., "Optimal Control of Multirate Digital Control Systems," Ph.D. Dissertation, Stanford Univ., Stanford, CA, 1980.

[17]Glasson, D. P., "A New Technique for Multirate Digital Control Design and Sample Rate Selection," *Journal of Guidance, Control, and Dynamics,* Vol. 5, No. 4, 1982, pp. 379–382.

[18]Al-Rahmani, H. M., and Franklin, G. F., "A New Optimal Multirate Control of Linear Periodic and Time-Invariant Systems," *IEEE Transactions on Automatic Control,* Vol. 35, No. 4, 1990, pp. 406–415.

[19]Berg, M. C., and Yang, G. S., "A New Algorithm for Multirate Digital Control Law Synthesis," *Proceedings of the IEEE Conference on Decision Control,* IEEE, Piscataway, NJ, 1988, pp. 1685–1690.

[20]Berg, M. C., "Design of Multirate Digital Control Systems," Ph.D. Dissertation, Stanford Univ., Stanford, CA, 1986.

[21]Ly, U. L., "A Design Algorithm for Robust Low-Order Controllers," Ph.D. Dissertation, Stanford Univ., Stanford, CA, 1982.

[22]Ly, U. L., "Robust Control Design Using Nonlinear Constrained Optimzation," *Proceedings of the American Control Conference,* IEEE, Piscataway, NJ, 1990, pp. 968–969.

[23]Voth, C. T., and Ly, U. L., "Total Energy Control System Autopilot Design with Constrained Optimzation," *Proceedings of the*

American Control Conference, IEEE, Piscataway, NJ, 1990, pp. 1332–1337.

[24]Gangsaas, D., Bruce, K. R., Blight, J. D., and Ly, U. L., "Application of Modern Synthesis to Aircraft Control: Three Case Studies," *IEEE Transactions on Automatic Control,* Vol. 31, No. 11, 1986, pp. 995–1014.

[25]Bossi, J. A., Jones, R. D., and Ly, U. L., "Multivariable Regulator Design for Robustness and Performance: A Realistic Example," *Proceedings of the American Control Conference,* IEEE, Piscataway, NJ, 1986, pp. 285–288.

[26]Ly, U. L., "Optimal Low-Order Flight Critical Pitch Augmentation Control Law for a Transport Airplane," *Proceedings of the AIAA Guidance and Control Conference,* AIAA, New York, 1984, pp. 743–757.

[27]Yang, G. S., "A Generalized Synthesis Method for Multirate Feedback Control Systems," Ph.D. Dissertation, Univ. of Washington, Seattle, WA, 1988.

[28]Gill, P. E., Saunders, M. A., and Wright, M. H., "User's Guide for NPSOL," Dept. of Operations Research, Stanford Univ., TR SOL 86-2, Stanford, CA, 1986.