

Development of MATLAB Tools and Case Study for Multirate Control Design ¹

Li Qiu², Tongwen Chen³, Andrew Potvin², Jun Gu⁴, Kan Tan², Yong Sun⁴, and Huang Shu³

²Dept. of EEE
HKUST
Clear Water Bay, Kowloon
Hong Kong

³Dept. of ECE
Univ. of Calgary
Calgary, Alberta
Canada T2N 1N4

⁴Dept. of CS
HKUST
Clear Water Bay, Kowloon
Hong Kong

Abstract

This paper reports on the development of MATLAB tools for analysis and design of multirate control systems. The routines developed are applied to a case study of designing power system stabilizers.

1 Introduction

Control system design for pure continuous or pure single-rate discrete systems is now a mature subject. The modern techniques based on \mathcal{H}_2 , \mathcal{H}_∞ , and \mathcal{L}_1 optimization have been thoroughly studied theoretically and applied to an increasingly wide range of domains. CAD software packages incorporating these design techniques, notably MATLAB Control System Toolbox, Robust Control Toolbox, and μ -Analysis and Synthesis Toolbox, are available commercially. However, these techniques and software cannot handle general computer control systems in which continuous and discrete systems coexist and discrete signals often have different sampling rates. A better setup reflecting these features is used in [6] and is shown in Figure 1. In this

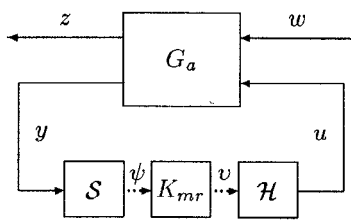


Figure 1: The general multirate sampled-data setup.

multirate sampled-data setup, G_a is an analog generalized plant with two (vector) inputs, the exogenous input w and the control input u , and two (vector) outputs, the signal z to be regulated and the measured

signal y . We assume that G_a is LTI with a state-space model

$$\hat{G}_a(s) = \left[\begin{array}{c|cc} A_a & B_{a1} & B_{a2} \\ \hline C_{a1} & 0 & D_{a12} \\ C_{a2} & 0 & 0 \end{array} \right].$$

S and H represent multirate sampling (A/D) and hold (D/A) operations and are defined as follows:

$$S = \begin{bmatrix} S_{m_1 h} & & \\ & \ddots & \\ & & S_{m_p h} \end{bmatrix}, H = \begin{bmatrix} H_{n_1 h} & & \\ & \ddots & \\ & & H_{n_q h} \end{bmatrix}.$$

These correspond to performing A/D conversions for the p channels of y periodically with periods $m_i h$ and D/A conversions for the q channels of v with periods $n_j h$. Here m_i and n_j are integers and h is a real number referred to as the *base period*. The linear multirate controller K_{mr} is assumed to satisfy three properties: periodicity, causality, and finite dimensionality [6]; then they admit description by some difference equations [6].

Figure 1 represents a hybrid and multirate system. Much progress has been made recently [16, 13, 6, 14] in faithfully capturing the two features in the \mathcal{H}_2 and \mathcal{H}_∞ optimal design frameworks. Specifically, a technique called lifting is used in both continuous and discrete time to convert the design problems into equivalent LTI ones. However, continuous lifting introduces infinite-dimensional operators in system representations and discrete lifting introduces a causality constraint in controllers. The software is able to handle the operators numerically and the causality constraint explicitly in \mathcal{H}_2 and \mathcal{H}_∞ optimizations.

The newly developed programs are capable of performing the following functions:

- Analysis of multirate control systems: This includes computation of various system norms, computation of stability and performance robustness, etc.
- Multirate model building: This includes the cascade, parallel, and feedback connections of mul-

¹The research has been supported by Sino Software Research Center, Hong Kong University of Science and Technology (HKUST). Corresponding author email: eeqiu@ee.ust.hk.

tirate systems, multirate discretization of analog systems, etc.

- Lifting and de-lifting of systems: These are for carrying out various equivalent conversions among multirate sampled-data systems, multirate discrete systems, single-rate sampled-data systems, and single-rate discrete systems.
- Nest matrix algebra: \mathcal{H}_2 and \mathcal{H}_∞ optimization of multirate systems depends on various factorizations and completion of nest matrices [6, 14]. Some, such as QR and Cholesky factorizations, exist in MATLAB; others, such as J-spectral factorization and contractive completion, need to be introduced.
- Multirate \mathcal{H}_2 and \mathcal{H}_∞ optimal control: The design procedures in [13, 6, 14] are implemented.
- Multirate controller model reduction: The purpose of controller model reduction is to seek lower-order controllers to replace high-order ones. With the nest matrix framework in [6, 14], the techniques for analog or single-rate model reduction are modified and implemented to handle multirate controllers.
- Multirate sampled-data simulation: This is to simulate general multirate sampled-data systems; the techniques implemented is based on fast sampling and lifting, which are extended from a single-rate procedure in [5]. This capability complements the method of simulation using SIMULINK, to be discussed later.

We hope that the tools will accelerate applications of the new multirate design methodologies and expand the domain of applications. In addition to the collection of M-files for design and analysis, a graphical user interface (GUI) is also designed. The GUI provides interactive data exploration of system responses and integrates the design and analysis functionality of the M-file routines with the modeling functionality of SIMULINK. Thus the tools provide a complete control design environment, from system modeling to problem formulation, and control synthesis, with both pre- and post-design analyses.

Multirate systems are abundant in industry; a small sample of applications of multirate control techniques includes aerospace control systems [7], chemical processes [17], power systems [9], robotic systems [12], and home electronic systems [15]. The tools developed can be used for practical multirate control design – in Section 4, we present a design example of multirate power system stabilizers.

This paper is organized as follows. Section 2 describes briefly the core \mathcal{H}_2 and \mathcal{H}_∞ control design routines of the toolbox. Section 3 discusses the multirate control interface and SIMULINK integration. Finally, Section 4 contains the power system example.

2 \mathcal{H}_2 and \mathcal{H}_∞ Control Algorithms

In view of Figure 1, multirate optimal control requires designing an internally stabilizing K_{mr} to minimize a certain performance spec J : In the \mathcal{H}_2 case, J is the average variance of z when w is a standard white noise; in the \mathcal{H}_∞ case, J is the \mathcal{L}_2 -induced norm from w to z .

The solutions to these problems are based on lifting. Let $\sigma = lh$ with l the least common multiple of m_i and n_j . It is a fact that the continuous system from w to z in Figure 1 is σ -periodic, if K_{mr} is periodic in a certain sense [6]. Let L_σ be the continuous lifting operator mapping a continuous signal to a discrete sequence taking values in $\mathcal{K} := \mathcal{L}_2[0, \sigma)$ (see, e.g., [2]), and L_m the m -fold discrete lifting operator [8]. Define $\bar{m}_i = l/m_i$, $\bar{n}_j = l/n_j$, and

$$\mathcal{L}_M = \begin{bmatrix} L_{m_1} & & \\ & \ddots & \\ & & L_{m_p} \end{bmatrix}, \mathcal{L}_N = \begin{bmatrix} L_{n_1} & & \\ & \ddots & \\ & & L_{n_q} \end{bmatrix}.$$

Then the multirate system of Figure 1 can be converted into a single-rate LTI discrete system in Figure 2, where

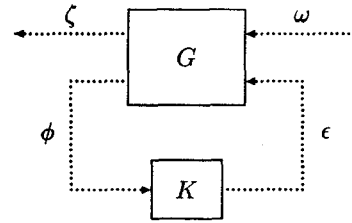


Figure 2: The lifted system

$$G = \begin{bmatrix} L_\sigma & \\ & \mathcal{L}_M S \end{bmatrix} G_a \begin{bmatrix} L_\sigma^{-1} & \\ & \mathcal{H} \mathcal{L}_N^{-1} \end{bmatrix},$$

$$K = \mathcal{L}_N K_{mr} \mathcal{L}_M^{-1}.$$

Note that w and ζ are \mathcal{K} -valued sequences. This implies that in the state-space model of G (with finite-dimensional state space),

$$\hat{G}(z) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right],$$

B_1 , C_1 , D_{11} , D_{12} , and D_{21} are all operators on appropriate spaces. Furthermore, due to causality of G_a and K_{mr} , the lifted system G and K are also causal and satisfy some causality constraint characterized by nest operators [6].

Let \mathcal{U} and \mathcal{Y} be the spaces spanned by all v and ψ , respectively, occurred in the time interval $[k\sigma, (k+1)\sigma)$. For $0 \leq r \leq l$, let the subspace \mathcal{U}_r be spanned by all v occurred during the interval $[k\sigma + (l-r)h, (k+1)\sigma)$;

similarly for \mathcal{Y}_r and ψ . Then the causality of G_a implies that

$$D_{22}\mathcal{U}_r \subseteq \mathcal{Y}_{r+1}, \quad r = 0, 1, \dots, l-1,$$

and the the causality of K_{mr} requires that

$$\hat{K}(\infty)\mathcal{Y}_r \subseteq \mathcal{U}_r, \quad r = 0, 1, \dots, l,$$

where $\hat{K}(\infty)$ is the D -matrix in the lifted controller. Using nest operators [6], these can be rewritten as $D_{22} \in \mathcal{N}_s(\{\mathcal{U}_i\}, \{\mathcal{Y}_i\})$ and $\hat{K}(\infty) \in \mathcal{N}(\{\mathcal{Y}_i\}, \{\mathcal{U}_i\})$.

Now the multirate problems can be stated in the lifted spaces. The multirate design problems reduce to designing an internally stabilizing K satisfying $\hat{K}(\infty) \in \mathcal{N}(\{\mathcal{Y}_i\}, \{\mathcal{U}_i\})$ to minimize J : In the \mathcal{H}_2 case, J is now the \mathcal{H}_2 norm of the LTI system $\hat{T}_{\zeta\omega}$ mapping ω to ζ ; and in the \mathcal{H}_∞ case, J equals to the \mathcal{H}_∞ norm of $\hat{T}_{\zeta\omega}$.

Although the discrete plant G has a state-space model with operators, the formulas for \mathcal{H}_2 and \mathcal{H}_∞ control solutions in the matrix-valued case are still applicable in principle. Take the \mathcal{H}_2 case first. We can still put the operators $B_1, C_1, D_{11}, D_{12}, D_{21}$ into the formulas for the optimal controller in the matrix case: First, find the stabilizing solutions of the Riccati equations

$$\begin{aligned} X &= A^*XA + C_1^*C_1 - (A^*XB_2 + C_1^*D_{12}) \\ &\quad \times (D_{12}^*D_{12} + B_2^*XB_2)^{-1} (B_2^*XA + D_{12}^*C_1), \\ Y &= AYA^* + B_1B_1^* - (AYC_2^* + B_1D_{21}^*) \\ &\quad \times (D_{21}D_{21}^* + C_2YC_2^*)^{-1} (C_2YA^* + D_{21}B_1^*), \end{aligned}$$

and let

$$\begin{aligned} F &= -(D_{12}^*D_{12} + B_2^*XB_2)^{-1} (B_2^*XA + D_{12}^*C_1), \\ L &= -(AYC_2^* + B_1D_{21}^*)(D_{21}D_{12}^* + C_2YC_2^*)^{-1}; \end{aligned}$$

then carry out factorizations

$$\begin{aligned} D_{12}^*D_{12} + B_2^*XB_2 &= R_{12}^*R_{12}, \\ D_{21}D_{21}^* + C_2YC_2^* &= R_{21}R_{21}^*, \end{aligned}$$

such that $R_{12} \in \mathcal{N}(\{\mathcal{U}_r\})$ and $R_{21} \in \mathcal{N}(\{\mathcal{Y}_r\})$. Thus a state-space realization (A_K, B_K, C_K, D_K) for the optimal controller is given by

$$\begin{aligned} A_K &= A + B_2F + LC_2 - B_2HC_2 - (-L + B_2H) \\ &\quad \times D_{22}(I + HD_{22})^{-1}(F - HC_2), \\ B_K &= (-L + B_2H)(I + D_{22}H)^{-1}, \\ C_K &= (I + HD_{22})^{-1}(F - HC_2), \\ D_K &= (I + HD_{22})^{-1}H, \end{aligned}$$

and

$$H = -R_{12}^{-1}\Pi_{\mathcal{N}(\{\mathcal{Y}_r\}, \{\mathcal{U}_r\})} \{ R_{12}^{*-1}(B_2^*XAYC_2^* + D_{12}^*C_1YC_2^* + B_2^*XB_1D_{21}^* + D_{12}^*D_{11}D_{21}^*)R_{21}^{*-1} \} R_{21}^{-1}.$$

Here, Π_S is the orthogonal projection to the subspace S . From these formulas, it is seen that the construction of the optimal \mathcal{H}_2 controller requires computing

operator compositions $B_1B_1^*, C_1^*C_1, D_{21}D_{21}^*, D_{12}^*D_{12}, D_{12}^*C_1, B_1D_{21}^*$, and $D_{12}^*D_{11}D_{21}^*$; these are all matrices and can be computed using matrix exponentials.

Define

$$P = \exp \left(h \begin{bmatrix} -A'_a & 0 & C'_{a1}C_{a1} & 0 \\ -B'_{a2} & 0 & D'_{a12}C_{a1} & 0 \\ 0 & 0 & A_a & B_{a1}B'_{a1} \\ 0 & 0 & 0 & -A'_a \end{bmatrix} \right), \quad (1)$$

$$Q = \exp \left(h \begin{bmatrix} -A'_a & 0 & C'_{a1}C_{a1} & C'_{a1}D_{a12} \\ -B'_{a2} & 0 & D'_{a12}C_{a1} & D'_{a12}D_{a12} \\ 0 & 0 & A_a & B_{a2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \quad (2)$$

and partition P and Q into block matrices $[P_{ij}]$ and $[Q_{ij}]$ ($i, j = 1, 2, 3, 4$) compatibly with the right-hand matrices in (1) and (2), respectively. Then

$$B_1B_1^* = \sum_{i=0}^{l-1} P_{33}^i P_{34} (P_{33}^{i+1})', \quad C_1^*C_1 = \sum_{i=0}^{l-1} (Q_{33}^{i+1})' Q_{13} Q_{33}^i.$$

Other operator compositions can also be obtained from the above matrix exponentials – details are omitted due to space limitation.

Similar lines can be followed for the \mathcal{H}_∞ problem but require considerably more efforts. Alternatively, one can reduce the \mathcal{H}_∞ problem step by step to an equivalent discrete, single-rate \mathcal{H}_∞ problem with a causality constraint and then solve the latter explicitly via contractive completion of nest matrices [6].

3 Integrated User Interface

In addition to the standard MATLAB command line functionality, the multirate system tools include an interactive graphical interface which may be invoked from the MATLAB command line or from a new SIMULINK block. It is widely believed that good GUI's for computer aided control system design are the key to bringing close the theory and application [10, 3].

The ideal user interface for control design would address the entire control design process including

1. System modeling
2. Control synthesis problem formulation
3. Actual controller synthesis
4. Pre- and post-design analysis

A prototype of our interface, developed in the MATLAB 5 language, appears in Figure 3. You see that it prominently displays the “big picture” for control design as the main tab of a tabbed dialog. On this tab, you can formulate and solve any (multirate) control design problem. Simply type the plant variable into the edit box, specify the sample rates and hold times, pick the

type of optimization you'd like to perform (\mathcal{H}_2 or \mathcal{H}_∞), and press the Synthesize button. Many of the other tasks you'd like to accomplish may be handled through other tabs of the interface. So let's summarize the rest of the interface in the context of addressing the control design steps listed above.

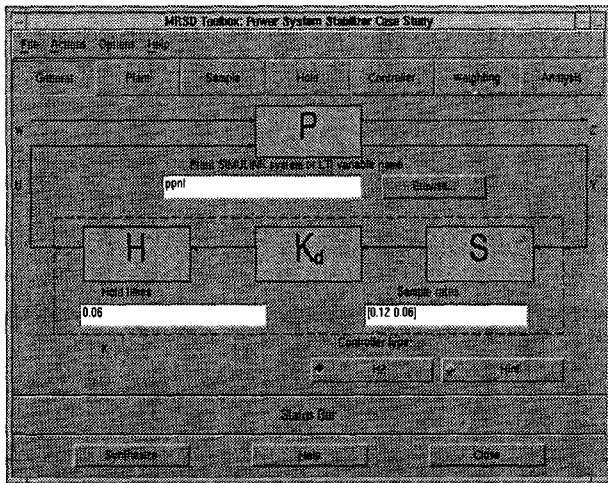


Figure 3: Prototype multirate control design interface

SIMULINK is no doubt an ideal tool to address Step 1, and our interface leverages the strength of SIMULINK to do system modelling. If the interface is created from the MATLAB command line, simply type the name of the SIMULINK model into the plant edit box; the plant is calculated as a linearization of that model. If instead, you place the newly developed multirate controller block within your SIMULINK system, you can create the interface by double-clicking the block and the SIMULINK model name automatically appears in the plant edit box. Should you not have SIMULINK, you may instead specify your system as an LTI system variable taken from the MATLAB workspaces. MATLAB 5 introduces these unified LTI system variable types. Clicking on the Plant tab or icon provides more information about the size of plant inputs, outputs, and state. For nonlinear SIMULINK plants, you should also specify an operating point about which the plant should be linearized with in this tab. The resulting linearized system will be the one used for design purposes.

Just how to address Step 2 remains an area of open research, not only for multirate sampled-data control, but for all robust control techniques. Such control methods require designers to introduce various frequency weightings on different signals to trade off control objectives. However, very few engineers in industry understand the frequency weighting selection process well enough to be able to apply the elegant theory to solve their particular problem. Unfortunately, many control theory researchers either lack the application experience or the technical interest to explain

these weightings to practicing engineers. Our interface hopes to guide you in making informed decisions about choosing these weights. Designing the Weighting tab interface continues to be a challenging part of our ongoing design effort. Current prototypes provide two problem formulation options. The first requires you to specify an ideal closed loop system; appropriate weighting functions are then chosen for you. We think this model matching approach will be particularly helpful to non-research engineers. The second option allows you to directly specify weighting functions on the exogenous input, plant output, and control signals.

MATLAB in general is the ideal tool to address Step 3. The GUI for this step is merely a thin layer on top of the underlying MATLAB function call. Specifically, once you input the plant, sample rates, hold times, and synthesis technique, simply click the Synthesize button to generate the controller by using the routines discussed in the previous section.

To address Step 4, the interface plots various plant and closed loop system responses via clicking checkboxes found on the Analysis tab. The analyses plots exhibit interactive data exploration; you can choose which inputs/outputs to plot, zoom and unzoom, change time/frequency or response scale or units, add extra points to plots, change line colors, etc. Interactive data exploration using MATLAB was first introduced in the version 4 release of the System Identification Toolbox [11].

We have already discussed the General, Plant, Weighting, and Analysis tabs of the interface. Clicking on the Sample (Hold) tab or icon, the user finds a listbox of controller inputs (outputs) listed by signal name. Engineers in industry find this functionality to be particularly useful as they typically remember the names of various signals, not the order in which SIMULINK groups the signals. The Controller tab allows you to modify various options to the control synthesis routines. Within it, you can also choose to generate and simulate with a customized C-code S-function of the multirate controller instead of using the generic multirate controller S-function. This brings up an important point about the structure of the multirate controllers discussed here. The non-LTI nature of the multirate controller's structure requires special attention to implement in any simulation environment or any real world system. To this end, two options exist to implement multirate controllers in SIMULINK. First a new block has been created to accommodate any size multirate controller. The new block's dialog box requires that the user specify the lifted state-space system, sample rates, hold times, and initial states. The controller dynamics are implemented as a C-coded S-function. Alternately, a MATLAB function has been created to generate a customized C-code S-function of

a specific multirate controller. This option is most useful in the final stages of the design process when code speed is a primary concern.

4 Multirate Power System Stabilizer

Using the software described above, we now design a multirate digital stabilizer for a single-machine, infinite-bus power system. A nonlinear model of the power system consists of a synchronous generator, a governor, an exciter and automatic voltage regulator (AVR), and transmission lines [1]. A power system stabilizer (PSS) provides additional damping to low frequency disturbances; it measures the rotor speed and generates a voltage control signal which is fed back to the exciter and AVR.

Under normal operating conditions [power $P = 0.9$ p.u. and power factor $P_f = 0.95$ (lag)], we obtain a linearized model [1, 4] for the power system as shown in Figure 4. Here Δp , $\Delta\omega$, $\Delta\delta$, and Δv_t are the deviations

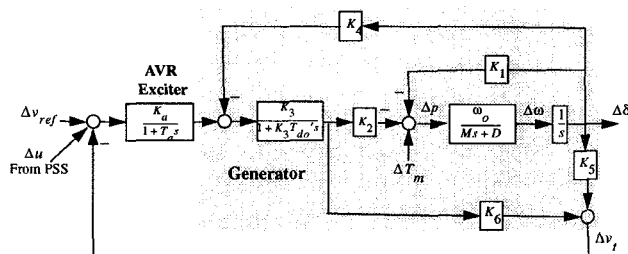


Figure 4: Linearized power system model

in power, speed, power angle, and terminal voltage respectively, while ΔV_{ref} is a reference voltage disturbance and ΔT_m is a mechanical torque disturbance. For ease of reference, we drop the prefix Δ in all variables from now on. By linearizing the nonlinear model about the given operating point, formulas for the parameters of the model in Figure 4 can be determined; the values for the parameters given in Table 1 are based on an experimental setup at the University of Calgary [4]. Let's now compare PSS designs for the fourth order system of Figure 4.

K_1	1.5495	M	8
K_2	1.2255	D	0
K_3	0.3822	ω_o	377
K_4	0.8910	T'_{do}	3.9640
K_5	-0.0138	K_a	190
K_6	0.4942	T_a	0.01

Table 1: Linearized power system model parameters

Our first design is an IEEE standard analog conventional power system stabilizer (CPSS) type PSS1A

shown in Figure 5. In our model, the control limit

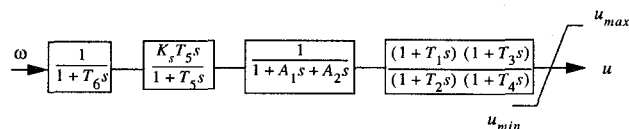


Figure 5: IEEE standard CPSS configuration

constraints are $u_{max} = -u_{min} = 0.1$ p.u. The parameters of the CPSS are tuned carefully for good response in δ (power angle) when a torque disturbance (step) is applied to the power system under normal operating conditions. The parameters are given in Table 2; since A_1 and A_2 are zero, the CPSS design is fourth order.

T_1	0.1	T_4	0.02	A_1	0
T_2	0.02	T_5	1.65	A_2	0
T_3	0.1	T_6	0.005	K_s	0.05

Table 2: Tuned analog CPSS parameters

In our second design, we discretize the analog CPSS above via the step-invariant approximation [5] to get a single-rate digital stabilizer with sampling period $h = 30$ ms.

We now discuss our multirate \mathcal{H}_2 -optimal design for the power system. We use the same control setup as the CPSS [one input (ω) and one output (u)], but ω is sampled with period mh ($h = 30$ ms) while u is updated with period nh . Initially we build a system with two disturbance inputs (T_m and V_{ref}) and three regulated signals, p , δ , and u , and then select weighting functions. We introduce a single weighting function $\hat{W}(s) = 1/(0.2s + 1)$ on both T_m and V_{ref} to get

$$\begin{bmatrix} T_m \\ V_{ref} \end{bmatrix} = \begin{bmatrix} W w_1 \\ W w_2 \end{bmatrix}, \quad w := \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

The theory requires the lowpass nature of W to guarantee the well-posedness of the \mathcal{H}_2 problem; practically, W defines a desired bandwidth for disturbance rejection. We define the regulated (weighted) output z as

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} p \\ C_\delta \delta \\ W_u u \end{bmatrix},$$

and after a few design iterations, decide on using

$$C_\delta = 0.25, \quad \hat{W}_u = \frac{2s + 3.5}{s + 10^{-7}}.$$

It can be shown that such weightings guarantee that, for a step disturbance, both ω and u are zero at steady state. This desirable property means that the PSS is "on" only during the transient process.

Combining the definitions of z and w with the system of Figure 4, we can put things into the standard framework of Figure 1 where P is sixth order. Using $m = 1$ and $n = 2$ (ω is sampled with period 30 ms and u is updated with period 60 ms), we use the software to compute the optimal \mathcal{H}_2 multirate stabilizer (which is also sixth order).

Figure 6 (7) shows the responses of δ and u for the three different designs due to step change of 0.1 p.u. in T_m (0.05 p.u. in V_{ref}). The dotted lines are for the analog CPSS design; the dash-dot lines are for the digital CPSS design; the solid lines are for the \mathcal{H}_2 multirate design. As expected, the digital design is noticeably

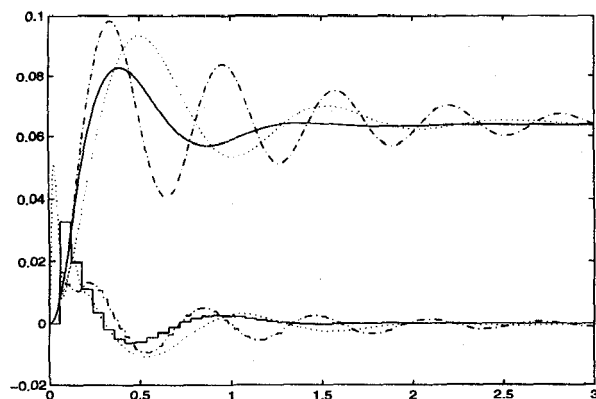


Figure 6: Step disturbance at T_m ; upper for δ vs. time (s) and lower for u vs. time: CPSS (dotted), discretized CPSS (dash-dot), multirate \mathcal{H}_2 stabilizer (solid).

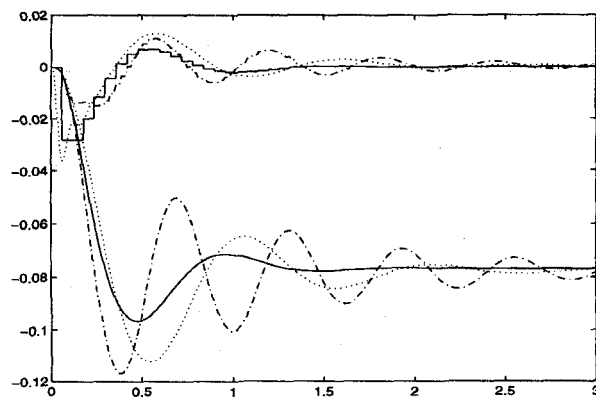


Figure 7: Step disturbance at V_{ref} ; upper for u vs. time (s) and lower for δ vs. time: CPSS (dotted), discretized CPSS (dash-dot), multirate \mathcal{H}_2 stabilizer (solid).

worse than the analog design. But note that the multirate stabilizer outperforms them both.

References

- [1] P.M. Anderson and A.A. Fouad, *Power System Control and Stability*, Iowa University Press, Iowa, 1977.
- [2] B. Bamieh and J.B. Pearson, "A general framework for linear periodic systems with application to \mathcal{H}_∞ sampled-data control," *IEEE Trans. Automat. Control*, vol. 37, pp. 418-435, 1992.
- [3] S. Boyd, "Robust control tools — graphical user-interfaces and LMI algorithms," *Systems, Control, and Information*, vol. 38, no. 2, pp. 111-117, 1994.
- [4] S. Chen and O.P. Malik, "An \mathcal{H}_∞ optimization based power system stabilizer design," *IEEE Proc. - Generation, Transmission and Distribution*, vol. 142(2), pp. 179-184, 1995.
- [5] T. Chen and B.A. Francis, *Optimal Sampled-Data Control Systems*, Springer, London, 1995.
- [6] T. Chen and L. Qiu, " \mathcal{H}_∞ design of general multirate sampled-data control systems," *Automatica*, vol. 30, pp. 1139-1152, 1994.
- [7] D.P. Glasson, "Development and applications of multirate digital control," *IEEE Control Systems Magazine*, vol. 3, pp. 2-8, 1983.
- [8] P.P. Khargonekar, K. Poolla, and A. Tannenbaum, "Robust control of linear time-invariant plants using periodic compensation," *IEEE Trans. Automat. Control*, vol. 30, pp. 1088-1096, 1985.
- [9] A. Khayatian and D. G. Taylor, "Multirate modeling and control design for switched-mode power converters," *IEEE Trans. Automat. Control*, vol. 39, pp. 1848-1852, 1994.
- [10] J. Little, "The GUI challenge: CACSD has come full circle and this remains a frontier", *34th IEEE Conf. on Decision and Control*, 1995.
- [11] L. Ljung, *System Identification Toolbox User's Guide*, The MathWorks, Inc., 1995.
- [12] S. Monaco and D. Normand-Cyrot, "An introduction to motion planning under multirate digital control," *Proc. 31st IEEE Conf. on Decision and Control*, pp. 1780-1785, 1992.
- [13] L. Qiu and T. Chen, " \mathcal{H}_2 -optimal design of multirate sampled-data systems," *IEEE Trans. Automat. Control*, vol. 39, pp. 2506-2511, 1994.
- [14] L. Qiu and T. Chen, "Multirate sampled-data systems: all \mathcal{H}_∞ suboptimal controllers and the minimum entropy controller," *Proc. 33rd IEEE Conf. on Decision and Control*, pp. 3707-3712, 1994.
- [15] T. Semba, "Model-following digital servo using multirate sampling for an optical disk drive," *Japanese Journal of Applied Physics, Part 1*, vol. 32, pp. 5385-5391, 1993.
- [16] P.G. Voulgaris and B. Bamieh, "Optimal \mathcal{H}_∞ and \mathcal{H}_2 control of hybrid multirate systems," *Systems & Control Letters*, vol. 20, pp. 249-261, 1993.
- [17] K.Y. Zhu and K.V. Ling, "Multirate control for a gasification process", *Proc. American Control Conference*, pp. 3268-3272, 1994.