

TABLE I
RESULTS OF INCOMPLETE STATE FEEDBACK CONTROL LAW DESIGN

$\hat{u} = -\hat{K}x$	\hat{J}	IRC (max.)
a) Control law structure \hat{u}_{II} $\hat{K} = \begin{bmatrix} 0 & 0 & 0 \\ 0.765 & 0 & 0 \end{bmatrix}$	25.65452	0.923899
b) Control law structure \hat{u}_{III} $\hat{K} = \begin{bmatrix} 0.42100 & 0 & 0 \\ 0.30419 & 0 & 0 \end{bmatrix}$	12.25000	0.98308
c) Control law structure \hat{u}_{III} $\hat{K} = \begin{bmatrix} 0.44533 & 0.41640 & 0 \\ 0.38959 & 0.63131 & 0 \end{bmatrix}$	10.75364	0.99922

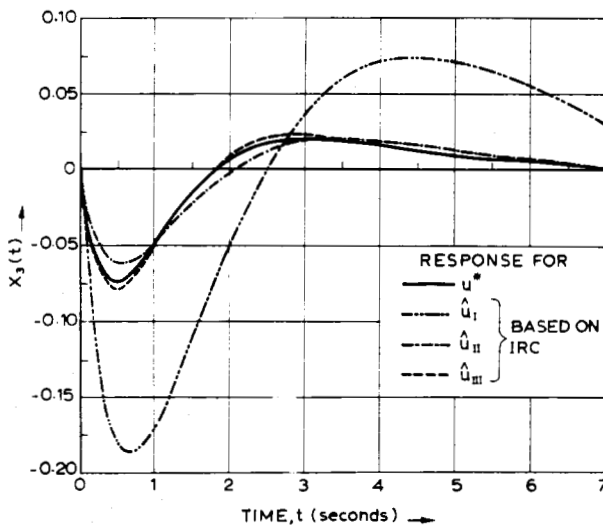


Fig. 1. Transient responses of $x_3^*(t)$ and $\hat{x}_3(t)$.

Example: Consider a two-input third-order system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (10)$$

and

$$J = \int_0^{\infty} (x_1^2 + x_2^2 + x_3^2 + 4u_1^2 + u_2^2) dt. \quad (11)$$

The optimal control law and cost are found to be

$$u^* = - \begin{bmatrix} 0.45682 & 0.39019 & 0.10163 \\ 0.40654 & 0.74090 & 0.38238 \end{bmatrix} x \quad (12)$$

and

$$J^* = 10.64895, \quad \text{for } x_0 = [1 \ 1 \ 1]'. \quad (13)$$

Three different control law structures are designed on the basis of the IRC and the results are given in Table I. The values of \hat{J} given in Table I are computed for $x_0 = [1 \ 1 \ 1]'$.

The transient responses of x_3 with respect to the suboptimal control law structures based on the IRC are shown in Fig. 1 along with the optimal response for $x_0 = [1 \ 0 \ 0]'$. It is noted from Table I that the greater the value of the IRC, the closer \hat{J} is to J^* ; further, Fig. 1 evidences that the IRC also stands as a measure of the closeness of the suboptimal transient response to the optimal transient response. The

suboptimal performance improves from \hat{u}_I to \hat{u}_{III} , as it should. The observations regarding the closeness of \hat{J} to J^* and the closeness of the suboptimal transient response to the optimal transient response are found to be true, for other initial conditions also. Numerical experimentation with other examples has revealed the validity of the proposed approach for the design of an incomplete state feedback control law.

CONCLUSIONS

A time-domain approach has been presented for the design of incomplete state feedback control laws for multi-input linear regulators. An example is given to illustrate the validity of the approach.

ACKNOWLEDGMENT

The authors are grateful to Prof. G. R. Damodaran, Director, PSG Institutions for his encouragement and interest.

REFERENCES

- [1] D. G. Luenberger, "An introduction to observers," *IEEE Trans. Automat. Contr.*, vol. AC-16, pp. 596-602, Dec. 1971.
- [2] H. R. Sirisena and S. S. Choi, "Optimal pole placement in linear multivariable systems using dynamic output feedback," *Int. J. Contr.*, vol. 21, pp. 661-671, Apr. 1975.
- [3] W. S. Levine, T. L. Johnson, and M. Athans, "Optimal limited state variable feedback controllers for linear systems," *IEEE Trans. Automat. Contr.*, vol. AC-16, pp. 785-793, Dec. 1971.
- [4] R. Subbayan and M. C. Vaithilingam, "An approach for incomplete state feedback control systems design," *Int. J. Contr.*, vol. 18, pp. 827-830, Oct. 1973.
- [5] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1971, ch. 3, pp. 35-36.
- [6] L. A. Zadeh and C. A. Desoer, *Linear System Theory: The State Space Approach*. New York: McGraw Hill, 1963, ch. 5, pp. 302-306.
- [7] A. Talbot, "The evaluation of integrals of products of linear system responses," *Quart. J. Mechanics and Applied Mathematics*, vol. 12, pt. 4, pp. 488-494, 1959.
- [8] H. H. Rosenbrock, "An automatic method for finding the greatest or least value of a function," *Computer Journal*, vol. 3, pp. 175-184, Oct. 1960.
- [9] D. L. Kleinman, "An easy way to stabilize a linear constant system," *IEEE Trans. Automat. Contr.*, vol. AC-15, p. 692, Dec. 1970.

Non-Real-Time Realization of Any Finite-State Sequential Machine Using a Single Shift-Register

PRADIP K. SRIMANI, BHABANI P. SINHA,
AND ARUN K. CHOUDHURY

Abstract—A model for simulating any finite-state sequential machine is presented as an alternative to the single shift-register model in [1]. The proposed model not only uses a single shift-register but also has the advantage of allowing a machine to be synthesized whose speed is improved by a factor of 2 and whose cost is reduced by nearly a factor of 2 over machines produced by the synthesis procedure associated with the former model.

I. INTRODUCTION

It is well known that every finite-state sequential machine is not single shift-register realizable. The problem to determine whether a given finite-state machine is shift-register realizable and if so, to find out the desired assignment, has already been extensively investigated [2], [3]. It has also been shown in [3], [4] that any arbitrary machine is shift-register realizable if several shift-registers are allowed, and the minimal solution of the assignment problem in this case is what gives the minimum number of shift-registers.

Besides this type of real-time realization of finite-state machines by single or multiple shift-registers, there is another type of non-real-time

realization of sequential machines by shift-registers, called simulation in [1]. A synchronous binary feedback shift-register simulation machine is structurally identical to the conventional shift-register realization machines, but is behaviorally distinguishable in that in conventional realization machines, all the possible shift-register codings are interpreted as stable states whereas in simulation machines some of the possible codes are interpreted as transitional states, i.e., states associated with prohibited outputs (i.e., with no outputs). In a conventional realization machine, the specified transition between two stable states as given in the flow table of the machine to be realized, takes place by a single shift pulse whereas in a simulation machine this is done by a multiple of shift pulses through a sequence of transitional states between the two stable states. It has been shown in [1] that all machines which are not realizable by a single shift-register, can be simulated on a single shift-register machine when transitional states are allowed.

The purpose of the present letter is to introduce a model, different from that described in [1], for non-real-time shift-register realization of any arbitrary sequential machine. The model yields a realization containing a single shift-register among other few memory elements. It will be shown that a simulation machine developed through this scheme operates more economically with respect to cost of construction as well as time of operation as compared to a simulation machine developed through the canonical synthesis procedure in [1].

II. DESIGN OF THE SHIFT-REGISTER SIMULATION MACHINE

Shift-registers form a special class of finite-state machines and are characterized by many interesting properties [2]-[4]. Using these properties, we will develop certain obvious results about shift-registers upon which our model and its related synthesis procedure will be based.

An n -length shift-register is a strongly-connected machine of 2^n states, with each state mapping to and being mapped from two distinct states. Each state S_i of a shift-register is reachable from any other state S_j by a path length of exactly n . Every finite-state machine, which is not single shift-register realizable, can be made strongly-connected and single shift-register realizable by judicious addition of transitional states [1].

Now consider any finite-state machine of k states. Let us assign a binary coding to each of the states with $\lceil \log_2 k \rceil = n$ state variables. In our model which uses an n -length shift-register, if we assume that a valid state transition occurs after n shift pulses and consider the intermediate state sequence to be transitional states, then by Theorem 1 we can reach any arbitrary state S_i starting from any state S_j , required by the given flow table, provided we can generate the required feedback sequence in a deterministic way by a combinational logic network. Of course, we assume, as in [1], that the external inputs required for the valid state transition are kept constant during the n shift pulses. Since the design procedure is to encompass all possible finite-state machines, our model must include the specification of the other memory elements to be able to determine the next state of a simulated machine. There are some cases where additional memory elements are needed. Consider two valid state transitions from states S_i to S_p and S_j to S_q , under the same external input, as can be required from the given flow table. It is possible that we require an intermediate transition from S_k to S_n in the first case and another from S_k to S_m in the second case, where S_k, S_m, S_n are all transitional states for the concerned valid state transitions. Obviously, information of S_k and external inputs alone are not sufficient to generate the correct feedback sequence in both the cases, and information of the starting valid state is required. Hence, if the starting state can be preserved in a separate register at the beginning of the n shift pulses, and the combinational feedback network uses this register as input, the aforementioned ambiguity in generating the correct feedback sequence is resolved.

After introduction of the initial state information in the feedback network, still another ambiguity can occur. Consider a valid transition from S_i to S_j . If there exists a path from S_i to S_j of length less than n , then in the n -length path there will exist a transitional state S_k which will occur more than once with different next states. To generate the correct feedback sequence in such cases a counter is used to count the n shift pulses to indicate the end of a valid transition. We now state the following obvious theorem.

Theorem 1: Any arbitrary k -state machine can be simulated on the model using an n -length shift-register, $n = \lceil \log_2 k \rceil$, with an n -length auxiliary register and a $\lceil \log_2 n \rceil$ -length counter.

It is to be noted that for a particular external input combination, a particular state can't be required to go to more than one next state since we are provided with deterministic flow tables. At this stage, the procedure to synthesize the simulation machine can be described as follows.

Step 1: Given a k -state machine, assign any arbitrary binary coding to each of the states with $n = \lceil \log_2 k \rceil$ state variables.

		0	1		0	1	$s_1 s_2$	$r_1 r_2$	c_1
s_1	s_1	s_2	00	s_1	\triangle	\triangle	00	00	0
s_2	s_1	s_3	01	\triangle	s_1	-	00	00	1
s_3	s_2	s_3	10	\triangle	-	s_2	10	00	1
				s_2	\triangle	\triangle	01	01	0
				\triangle	s_1	s_3	00	01	1
				s_3	\triangle	\triangle	10	10	0
				\triangle	s_2	-	11	10	1
				\triangle	-	s_3	01	10	1

Fig. 1. Flow table of the example machine.

Step 2: For each state transition in the given flow table determine an n -length state sequence by the end of which required final state is obtained and also note the required feedback sequence for each such state transition.

Step 3: Construct the truth table of the feedback logic network to generate the feedback sequences of Step 2 from the shift-register state variables, initial state register variables, counter variables and external input variables. Minimize this truth table by any standard technique to complete the synthesis of the simulation machine.

Thus the present simulation model is distinguished from the Freeman and Levy model [1] in that it uses a single-shift register, a regular register, a counter and a feedback logic where the model of [1] uses a single shift-register and feedback logic. In the simulation machine developed through our scheme, each valid transition takes a time of n shift pulses, the feedback logic is a function of $(2n + \lceil \log_2 n \rceil + x)$ variables, x denoting the number of external input variables, and the machine uses $2n$ memory elements, whereas in the Freeman and Levy model, each valid transition takes a time of $2n$ shift pulses, the feedback logic is a function of $(4n + x)$ variables and the resulting machine uses $4n$ memory elements. Evidently the present procedure gives economy in time of computation by a factor of 2 and also minimizes the cost of implementation (which is usually measured in terms of number of memory elements and number of input variables in the feedback network) by a factor of nearly 2. It should also be noted that the present realization algorithm, like that in [1], in no way claims to give an optimal assignment in any sense, but it shows that the proposed model can realize in a non-real-time fashion any finite-state machine with less cost and less time of operation than the synthesis procedure for the Freeman and Levy model.

III. AN ILLUSTRATING EXAMPLE

Consider the following 3-state machine which is not single shift-register realizable. To construct the single shift-register simulation machine, we require a 2-length ($\log_2 3$) shift-register, a 2-length initial state register and a scale of 2 counter. Let the shift-register state variables be (s_1, s_2) , the initial state register variables be (r_1, r_2) and the counter variable be c_1 . The 3 states in the flow table can be arbitrarily coded as $h(S_1) = 00, h(S_2) = 01$, and $h(S_3) = 10$ (where h is a state assignment function) (see Fig. 1). The given flow table is augmented by transitional states $\triangle a, \triangle b, \triangle c, \triangle d$, and $\triangle e$ in order to produce a 2-length transitional state sequence between state transitions in the original flow table. The combinational logic function $F(s_1, s_2, r_1, r_2, c_1)$ can be constructed to generate these transitional state sequences. This combinational logic can be minimized as usual. It is to be noted that this minimization is usually simple since there will ordinarily exist a large number of DON'T CARE next states.

REFERENCES

- [1] L. S. Levy and M. Freeman, "Every finite-state machine can be simulated (realized) by a synchronous (asynchronous) binary feedback shift-register machine," *IEEE Trans. Comput.*, vol. C-23, pp. 174-178, Feb. 1974.
- [2] R. L. Martin, *Studies in Feedback Shift Register Synthesis of Sequential Machines*. Cambridge, MA: M.I.T. Press, 1969.
- [3] C. C. Su and S. S. Yau, "Unitary shift-register realizations of sequential machines," *IEEE Trans. Comput.*, vol. C-17, pp. 312-324, Apr. 1968.
- [4] A. J. Nichols, "Minimal shift-register realizations of sequential machines," *IEEE Trans. Electron. Comput.*, vol. EC-14, pp. 688-700, Oct. 1965.