

Optimal Control of Switching Linear Systems

Ali Benmerzouga

Department of Mathematics and Statistics, College of Science, Sultan Qaboos University, P. O. Box 36, Al Khod 123, Muscat, Sultanate of Oman, Email: aliben@squ.edu.om

.(1997)

()

.(1997)

ABSTRACT: A solution to the control of switching linear systems with input constraints was given in Benmerzouga (1997) for both the conventional enumeration approach and the new approach. The solution given there turned out to be not unique. The main objective in this work is to determine the optimal control sequences $\{U_i(k), i = 1, \dots, M; k = 0, 1, \dots, N-1\}$ which transfer the system from a given initial state X_0 to a specific target state X_T (or to be as close as possible) by using the same discrete time solution obtained in Benmerzouga (1997) and minimizing a running cost-to-go function. By using the dynamic programming technique, the optimal solution is found for both approaches given in Benmerzouga (1997). The computational complexity of the modified algorithm is also given.

KEYWORDS : Controllability; Bilinear Systems; Dynamic Programming; Switching Linear Systems; Optimization.

1. Introduction

One of the most important objectives of dynamical systems in control theory is to find the optimal control $U(t)$. Most of the work done is primarily in the area of control and analysis of continuous bilinear systems described as

$$\frac{dX(t)}{dt} = \sum_{i=1}^M U_i(t) A_i X(t), \quad (1.1)$$

where $X(t) \in \mathbf{R}^n$, A_i are $n \times n$ matrices, $U_i : [0, \infty] \rightarrow U$ where $U \subseteq \mathbf{R}$.

The work that has been done in bilinear systems considers the control $U(t)$ as a scalar belonging to a compact set U such that $U(t) \leq \delta$. The case where $U(t)$ takes just two values as an ON-OFF system (i.e., $U(t) = 0$ or 1) is given by Benmerzouga (1985, 1997). A discrete solution to the continuous dynamical system is given in Benmerzouga (1997). For both the conventional enumeration approach and the new approach, the solution found was not unique.

Most of the work and emphasis in this case has been on the structural aspects of bilinear systems as given in Aslanis (1983) where $X(t) \in \mathbf{R}^2$ and $U(t) \in \{0, 1\}$. Tarn, *et al.* (1973) tested the controllability of discrete bilinear systems. Goka, *et al.* (1973) gave necessary and sufficient conditions for the controllability of a class of discrete bilinear systems. All the work done in discrete bilinear systems considers the control $U(t)$ as a scalar belonging to a compact set U such that $U(t) \leq \delta$. The case where $U(t)$ takes just two values as an ON-OFF system (i.e., $U(t) = 0$ or 1) as shown by Benmerzouga (1985), has had limited treatment.

The major focus in this work is to determine optimal control sequences $\{U_i(t), i = 1, \dots, M\}$ which transfer the system from a given initial state X_0 to a specific target state X_T (or to be as close to it as possible). In both the conventional enumeration approach and the new approach treated in Benmerzouga (1997), the properties of the discrete solution, which is derived by sampling (or discretizing) the continuous solution of the continuous dynamical system described by equation (1.1), are conserved and used in this paper. The solution found by Benmerzouga (1997) steers the system from an initial state X_0 to a final state X_T through more than one path. That is due, of course, to the performance index used. The only criterion used in Benmerzouga (1997) is the magnitude of the gap between the final state X_N and the target state X_T . There was no restriction whatsoever on how much it would cost to steer the system from the initial state to the final state. Hence, there are no restrictions on the paths which are found to be optimal. A dynamic programming technique is used to find the optimal path through which the control sequence will steer the system. Various studies and algorithms used are described in Section 2. The cost-to-go function and the dynamic programming approach used in the modified algorithm are given in Section 3. Section 4 includes the step-by-step procedure used to solve the problem optimally by using dynamic programming and the computational complexity of the modified algorithm. Finally some concluding remarks are given in Section 5.

2. Dynamics of Switching Control Systems

The dynamics of switching control systems can be studied for both continuous and discrete time systems as mentioned in Benmerzouga (1985) and (1997). The emphasis, in this paper, is mainly about discrete time systems obtained by sampling (or discretizing) a continuous system given by equation (1.1). This choice is made because of the following two main reasons that were mentioned in Benmerzouga (1997): (i) most control algorithms are implemented on a digital computer, and (ii) when discretizing a continuous system the discrete system obtained will have a non-singular characteristic matrix. For more

details see Wismer and Chattergy (1978). Such ground will make the computations of the switching controls much easier, as will be shown in the subsequent sections.

2.1 Solution of the Continuous Switching System

As stated earlier, the dynamics of the continuous switching system is given by equation (1.1) in addition to the constraints,

$$U_i(t) \cdot U_j(t) = 0, \quad i \neq j, \quad \text{and} \quad \sum_{i=1}^M U_i(t) = 1, \quad (2.1)$$

where M is the number of systems that can be used when steering the system from the initial state X_0 to the target state X_T .

The solution of such a dynamical system, with only one active system, in the interval $[0, t]$ is

$$X(t) = e^{Bt} \cdot X(0), \quad (X(0) \text{ is given}). \quad (2.2)$$

where the $n \times n$ matrix B is equal to any of the given state matrices A_i 's, depending on which control $U_i(t)$ is equal to 1, *i.e.*, which system is actually ON or active. (For more details see Brockett (1970), Kailath (1980) and Sandell and Athans (1974)).

2.2 The Discrete Solution for the Switching System

The discrete solution of the switching system is obtained by sampling (or discretizing) the continuous solution given by Equation (2.2). Since the initial state $X(0)$ is given, then the discrete solution of the switching system at each stage k is given by the following difference equation:

$$X(k+1) = e^{B\Delta T} \cdot X(k), \quad k = 0, 1, \dots, N-1. \quad (2.3)$$

Since there are M systems in the above dynamics, and since the initial state X_0 is given, and in view of the details given by Benmerzouga (1997), the discrete time solution of the switching system with a single state matrix is now completely defined by the difference equation

$$X(k+1) = \sum_{i=1}^M U_i(k) \cdot P_i \cdot X(k), \quad (2.4)$$

where P_i are $n \times n$ matrices and have the same definitions and properties given in Benmerzouga (1997).

Having the discrete solution of the switching system given by (2.4), the main purpose of this work is to find a computationally efficient approach that will optimally steer the system from an initial state X_0 to a target state X_T (or to be as close as possible). In addition to that, we should find the optimal control sequence $\{U_i(k), i = 1, 2, \dots, M, k = 0, 1, \dots, N-1\}$ that will achieve such goal in a given number of steps. Therefore, the first objective is to minimize the total running gap (or cost) accumulated when steering the system from an initial state X_0 to the target state X_T in a given number of steps and the second objective is to find the optimal control sequence $\{U_i(k), i = 1, 2, \dots, M, k = 0, 1, \dots, N-1\}$.

The accumulated gap (or costs) can be computed using different measures depending on the system we are actually using. This measure could be a distance, a quadratic form, or something different that will quantify and measure the cost. The form of the gap (or cost) used in this work and how to minimize it are described in detail in Section 4.

3. Previous Methods

Since Equation (2.4) gives the discrete solution of the switching system, the main purpose of the previous methods is only to minimize the gap (or cost) between the final state of the system $X(N)$ and the target state X_T , when steering the system from an initial state X_0 to the target state X_T in a given number of steps. In addition to that, the control sequence $\{U_i(k)\}_{k=0}^{N-1}$ that will achieve this goal in a given number of steps has to be found. The gap (or cost) used as a criterion for optimality is given by

$$g[X(N)] = \|X_T - X(N)\|^2. \tag{3.1}$$

3.1 Method 1: The Conventional Method

By using the conventional enumeration approach, all the states at all stages are computed and stored in order to reach the last stage as shown on Figure 3.1.

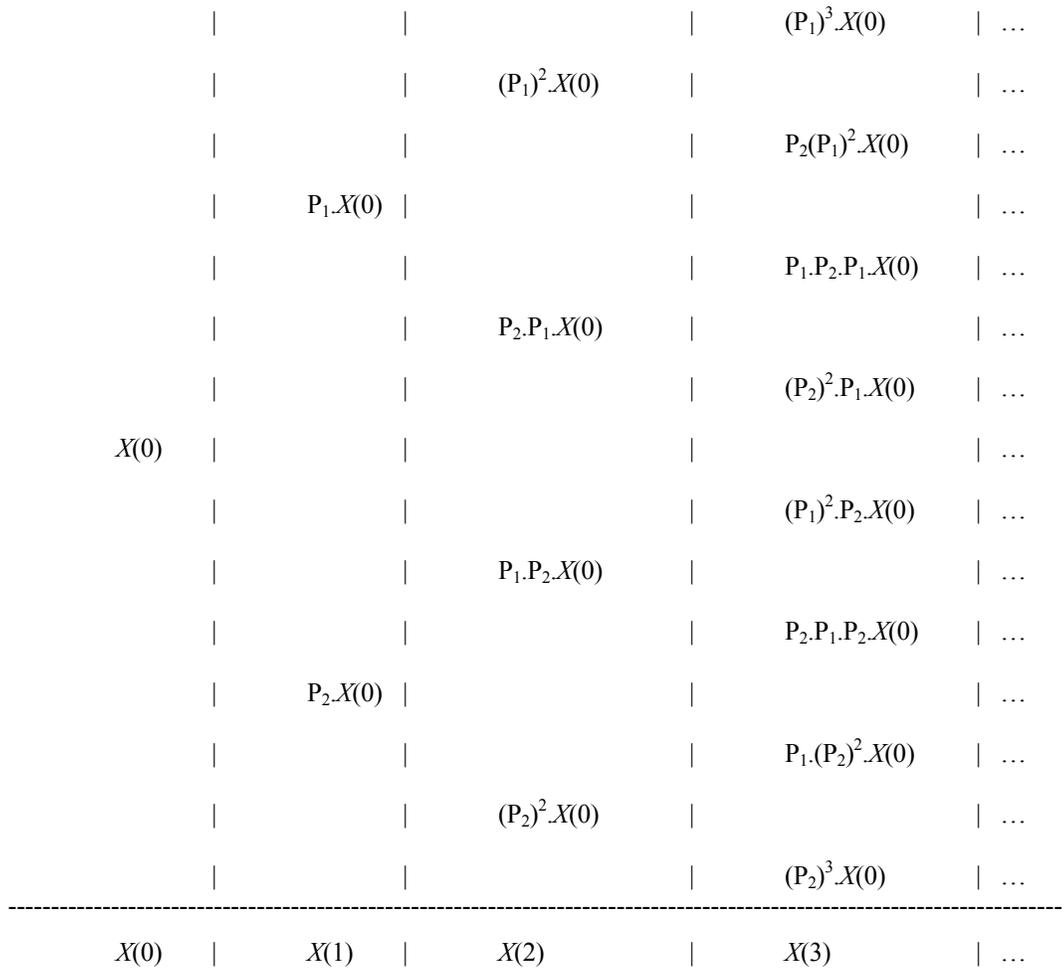


Figure 1. All Possible Stages and States for $k=3$ and $N=2$.

When solving Example 5.1 in Benmerzouga (1997), the final cost was obtained by two different control sequences, namely, $(U(0), U(1), U(2)) = (0, 1, 1)$ and $(1, 1, 0)$, respectively. In addition to that, the solution to the conventional problem was found to be not unique. The value of the terminal cost, namely, $g[X(N)]$, is found to be equal to zero through two different trajectories. All the details of the computations are fully executed and presented in Benmerzouga (1997).

3.2 Method 2: The New Approach

The new approach description, mathematical development and solution are completely given in detail in Benmerzouga (1997). When solving the same example, the minimum gap (or cost) was found to be equal to 1 and is given by two different paths through the two different control sequences $(0, 1, 1)$ and $(1, 1, 0)$. This is exactly what we obtained in subsection 3.1. It was shown in Benmerzouga (1997) that the new approach is much better than the enumeration method from the storage and computations point of view and from the very simple way used to get the control sequence corresponding to the minimum final cost. As we can see in the previous two methods we are getting more than one control sequence as a solution. In other words, there is more than one trajectory to steer the system from an initial state X_0 to a final state X_N . The question that remains open is which of the above paths is really the best to go through from a cost point of view. This will be fully explained and illustrated in the next section.

4. The Cost-to-Go Function

The performance index used in Benmerzouga (1997) was basically focusing on the smallest terminal or final cost $g[X(N)]$ and not on the smallest overall running cost. The procedure used in Benmerzouga (1997) was kind of memoryless of what is happening between steps and through those trajectories. The only objective in Benmerzouga (1997) was to pick the path with minimum terminal cost and not the overall running cost. This was the main reason for having more than one path as solutions to the problem.

The performance index used in this analysis is the one that minimizes the total running cost incurred while steering the system from the initial state X_0 to the target state X_T in a given number of steps, N , or the one that will give the minimum cost-to-go when being as close as possible to the target state X_T . The cost-to-go function is defined by the recurrent formula given by

$$V = \sum_{k=0}^{N-1} F[U(k), X(k)] + g[X(N)], \quad (4.1)$$

where V is the final running cost, $U(k)$ is the control sequence which takes the values 0 or 1 and $F[U(k), X(k)]$, and $g[X(N)]$ are given by

$$F[U(k), X(k)] = \|X_T - X(k)\|^2 = [X_T - X(k)]^T [X_T - X(k)], \quad (4.2)$$

$$g[X(N)] = \|X_T - X(N)\|^2 = [X_T - X(N)]^T [X_T - X(N)]. \quad (4.3)$$

Equation (4.1) has a structure that requires the use of a dynamic programming approach for its minimization. More details about dynamic programming are given in Dreyfus and Law (1977). All the criteria, structures and restrictions, mentioned in Benmerzouga (1997), are conserved and used in this paper.

4.1 The Modified Algorithm

This modified algorithm is built on the basis of the details of the algorithm provided in Benmerzouga (1997). As mentioned earlier, one of the objectives in this work is to steer the system from an initial state X_0 to a target state X_T (or as close as possible) in a given number of stages, N , with minimum running cost (or distance) and not a minimum terminal cost. Therefore the required steps are:

- Step 1:** Given the number of stages N and an initial state X_0 , a matrix $S(I, J)$ is generated which will contain all different values of $Z(N)$, where I is the space dimension and J the number of states at stage N . This step is the same as Step 1 in the old algorithm given by Benmerzouga (1997).
- Step 2:** When searching for the minimum running distance it is necessary to go back from Z space to X space using the relationship $X(k) = B^k \cdot Z(k)$. Hence it is necessary to compute the matrix B^k .
- Step 3:** Compute $X(k) = B^k \cdot Z(k)$, namely, the transformation from Z -space to X -space for each k .
- Step 4:** Compute all possible costs (or performance indices) at each stage k .
- Step 5:** Store all the computed costs in the matrix $CM(I, J)$ as shown in Table 4.
- Step 6:** Find the final running cost on each path by adding up the costs of all the previous stages.
- Step 7:** Given the rank, (J) , of the minimum running cost, a binary representation is generated to get the optimal control sequence $\{U_i(k), k = 0, 1, \dots, N-1 \text{ and } i = 1, \dots, M\}$.

The above algorithm not only makes the computations tractable, but it also facilitates the way to get the optimal control sequence to reach the final target for the problem. Therefore, given an initial state of the system X_0 , a target state X_T , and a number of steps N , the algorithm will give us the optimal state and its corresponding sequence $\{U_i(k), k = 0, 1, \dots, N-1 \text{ and } i = 1, \dots, M\}$.

Without loss of generality, the dynamic programming approach and the overall running costs criteria are used for the two methods given in the previous section. The computations are carried out for the following bilinear example:

Example 4.1

The same example used in Benmerzouga (1997) is to be used here. Let P_1 and P_2 be 2×2 matrices, X_0 and X_T be 2×1 vectors, and the number of stages, N , equals to 3.

$$P_1 = \begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}, P_2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, X_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, X_T = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

4.2 Method 3: The Dynamic Programming Technique Applied to the Conventional Problem

Now we need to use the dynamic programming approach to the conventional problem by minimizing the cost-to-go function given in (4.1) and not only the terminal cost. In order to compute the intermediate

SWITCHING LINEAR SYSTEMS

running cost, a complete enumeration and computation of all costs for all the possible states at all stages has to be performed as it is shown in Table 1.

By using the recurrent formula given by Equation (4.1), the costs-to-go from one stage to the next one, namely, $F[U(2), X(2)]$, $F[U(1), X(1)]$ and $F[U(0), X(0)]$, will lead to the optimal solution for this case and its optimal control sequence. The results of the different computations are summarized in Table 2 (a)-(c). The sequence that generates the optimal path when using dynamic programming is unique and it is equal to $(U_1(0), U_1(1), U_1(2)) = (0, 1, 1)$. Since we are using a bilinear system the other control sequence is just the complement to this one, i.e., $(U_2(0), U_2(1), U_2(2)) = (1, 0, 0)$. As we can see from Table 2 (c), we have just one single optimal path with a minimum running cost of 5.

4.3 Method 4: The Dynamic Programming Technique Applied to the New Approach

Given the data obtained using subsection 4.2 and by using the dynamic programming technique, the minimum distance is found to be equal to 1 through a unique control sequence $(0, 1, 1)$ with a minimal overall running cost of 5. Knowing the structure of the new approach and how the data and the computations are to be handled and stored, a useful new data is obtained and stored. This data is used to find the corresponding value of $Z(k)$ in our original X -space as shown in Table 3. The latter one is going to be used to compute the running costs. This is done by using just one extra matrix, namely, B , until an initial state is obtained. This can be done without any extra storage, except for a matrix $CM(I, J)$ that is used to store different distances at each stage, where I is the number of paths and J is the number of stages. The matrix $CM(I, J)$ is given in Table 4.

Table 1. The Complete Computational and Sequence Picture, when $N = 3$.

$X_r - X(0)$	$f[U(0), X(0)]$	$X_r - X(1)$	$f[U(1), X(1)]$	$X_r - X(2)$	$f[U(2), X(2)]$	$X_r - X(3)$	$g[X(3)]$
						-2	1
						3	13
						0	0
		-2	13	-2	4	-3	25
-1	4			0	1	0	1
1		0		-3		-1	
		0	0	0	9	0	0
				0	4	-1	5
				2		2	
						0	1
						-1	
						-2	8
						2	

By localizing the rank of the minimum final running cost in the matrix $CM(I, J)$, a binary representation is given to that rank in order to find the optimal control sequence $U_i(k)$. As we can see

from Table 4 the final cost is equal to 1 and its rank in the $CM(I, J)$ matrix is 3. The only thing left is to get the binary representation of rank $J = 3$ to get the optimal control sequence $U_i(k)$, which is equal to (0,1,1) and of course it is unique.

Table 2 (a). The Different Paths, Gaps, Combinations of $U_i(k)$, and Minimum Gaps when $N = 3$.

$X(2)$	$f[U(2), X(2)]$	V_2	$U(2)$
1 1	4	4	0
2 1	9	10	0
-1 0	1	1	1
-1 -1	4	5	1

Table 2 (b). The Different Paths, Gaps, Combinations of $U_i(k)$, and Minimum Gaps when $N = 3$.

$X(1)$	$f[U(1), X(1)]$	V_1	$U(1)$
1 -2	13	17	1
-1 1	0	1	1

Table 2 (c). The Minimum Running Cost.

$X(0)$	$g[U(0), X(0)]$	V	$U(0)$
1 1	4	5	0

By using the dynamic programming approach, the following is observed: (i) in methods 3 and 4 we ended up with the same final results, namely, the minimum distance is unique, equal to 1 and obtained through the unique control sequence (0, 1, 1), (ii) when applying the dynamic programming approach directly to the original problem, all the states have to be computed and stored and (iii) when applying the dynamic programming to the new approach that is devised in Benmerzouga (1997), just half of the states have to be computed and stored and the optimal control sequence is obtained by using a very simple combinatorial approach.

4.4 The Complexity of the Modified Algorithm

The complexity of the modified algorithm can be traced back to the continuous system given by Equation (1.1). The parameter n , which is the state space of the system and k , which is the number of stages desired, and all types of computations and comparisons done at each stage must be fully taken into account and used to find the complexity of the whole procedure.

SWITCHING LINEAR SYSTEMS

Table 3. The Different Values of $Z(N)$ and $X(N)$, their Corresponding Ranks and the Minimum Gaps When $N = 3$ and $M = 2$.

Z(3)	X(3)	$g[X(3)]$	Rank
1	1		
1	-1	8	0
-2	-1		
-1	2	1	1
1	0		
0	-1	5	2
-1	-1		
-1	1	0^*	3
-2	-1		
-1	2	1	4
3	2		
2	-3	25	5
-1	-1		
-1	1	0^*	6
2	1		
1	-2	13	7

Proposition 4.1:

The complexity of the above procedure is of order $2^k n^3$, where n is the state space of the system and k is the number of stages involved.

Proof:

First of all, the transition matrix given in Equation (1.1) is found by using the eigenvalue and eigenvector methods for both state matrices $(A_i, i = 1, 2)$. This is a very classical method used in diagonalizing matrices. For more details see Gerald and Wheatley (1994). This step requires $(10n^3 + 2n^2)$ number of computations. The next step involves propositions 4.2 and 4.3 given in Benmerzouga (1997). Secondly, the computations of different states for every stage requires $(2kn^3 + n)$ number of computations for computing the matrices $Q(k)$ for all stages and $(2^k - 1)(n^3 + n)$ number of computations to find all states at all stages. For more details about the complexity of this step see Gerald and Wheatley (1994). Thirdly, we go from the new Z -space to the old X -space $[(k - 1)n^3 + 2^k n^2]$ computations are required. Fourthly, the number of computations needed to get the performance index $g[X(N)]$ is $2^k(3n - 1)$ and finally the number of comparisons involved to find the smallest distance is 2^k .

In summary, to obtain the complete complexity figure, all of the above has to be taken into account. Hence the complexity of the above procedure is equal to

$$(2^k + 3k + 8)n^3 + (2^k + 2)n^2 + 2^{k+1}n.$$

Therefore the complexity of our optimal control problem is of order $(2^k n^3)$. This number is quite large when both k and n are large.

Table 4. $CM(I, J)$ Matrix : All the Different Distances Using the Running Cost

$\ X_T - X(1)\ ^2$	$\ X_T - X(2)\ ^2$	$\ X_T - X(3)\ ^2$	Rank J	Final Cost
0	4	8	0	12
0	4	1	1	5
0	1	5	2	6
0	1	0	3	1**
13	9	1	4	23
13	9	25	5	47
13	4	0	6	17
13	4	13	7	30

5. Concluding Remarks

This work presents two new ways of solving a switching control problem for discrete time systems, namely, a dynamic programming technique applied to the original problem and a dynamic programming technique applied to the new approach. The discrete time system is obtained by sampling (or discretizing) the continuous time system.

The same restrictions are maintained as in Benmerzouga (1997). The same results were obtained when applying the dynamic programming technique to both the conventional enumeration and the new approach when solving Example 4.1. But when using the conventional enumeration approach and the dynamic programming approach applied to the conventional enumeration approach, all the states at each stage have to be computed and stored. On the other hand just half of the states have to be computed and stored with the new approach and the dynamic programming applied to the new approach.

The nice thing about the dynamic programming technique when applied to both the original problem and the new approach, it manages to optimally steer the system from the initial state X_0 to the target state X_T and gives us the cheapest path possible. On the other hand, when solving the original problem directly or by using the new approach, there was no concern on how much it would cost us in order to reach a pre-specified target X_T or to be as close as possible.

Also, all the good properties about the new approach mentioned in Benmerzouga (1997) are kept untouched and are added to all the nice and powerful characteristics known about the dynamic programming approach. Economically speaking, the savings are huge when adopting this modified algorithm. The savings are made in the number of computations, the storage required and in the overall cost to go from the initial state X_0 to the target state X_T . All of these put together, have made this work much more efficient than the work given in Benmerzouga (1997). When the number of steps is small the procedure is not computationally difficult.

Finally, when applying the dynamic programming technique to the original problem and to the new approach the performance is basically the same for a small number of steps. But when the number of steps increases applying dynamic programming to the new approach will take over all the way. In other words, the computations and storage involved in the procedure will be cut down by at least 50 % as shown in

Benmerzouga (1997). In addition to the above, another important ingredient that improves the efficiency of the algorithm when applied using the dynamic programming approach to the new approach is the hassle-free way to determine the control sequence $\{U(k), k = 0, \dots, N-1\}$, which is complicated and not easy when finding it in the conventional enumeration approach.

A modified algorithm, corresponding to the dynamic programming approach when applied to the new approach that is given in Benmerzouga (1997) was developed. The computer simulations showed that the obtained modified algorithm performed successfully in computing the optimal control sequences $\{U(k), k = 0, \dots, N-1\}$. The modified algorithm was shown to perform adequately when the number of steps increases.

6. Acknowledgement

Thanks are due to anonymous referees for their many helpful suggestions.

7. References

- ASLANIS, J.T. 1983. Analysis of Switched Linear Systems in the Plane, *M.Sc. Thesis*, Department of Systems Engineering, Case Western Reserve University, Cleveland, USA.
- BENMERZOUGA, A. 1985. The Control of Switched Linear Systems, *M.Sc. Thesis*, Department of Systems Engineering, Case Western Reserve University, Cleveland, USA.
- BENMERZOUGA, A. 1997. Using Switching Linear Systems to Reach a Pre-specified Target. *Sultan Qaboos University Journal for Scientific Research -Science and Technology*, **2**: 69-76.
- BROCKETT, R.W. 1970. *Finite-Dimensional Linear Systems*, Wiley, New York.
- DREYFUS, S.E. and LAW, A.M. 1977. *The Art and Theory of Dynamic Programming*, Academic Press, New York.
- GERALA, C.F. and WHEATLEY, P.O., 1994. *Applied Numerical Analysis*. Addison-Wesley, New York.
- GOKA, T., TARN, T.J. and ZABORSZKY, J. 1973. On the Controllability of a Class of Discrete Bilinear Systems, *Automatica*, Vol. **9**, pp. 615-622. Pergamon Press.
- KAILATH, T. 1980. *Linear Systems*, Prentice Hall, New Jersey.
- SANDELL, N.R. and ATHANS, M. 1974. *Modern Control Theory*, Massachusetts Institute of Technology.
- TARN, T.J., ELLIOTT, D.L. and GOKA, T. 1973. Controllability of Discrete Bilinear Systems with Bounded Control, *IEEE Transaction on Automatic Control*, **AC-18**: 298-301.
- WISMER, D.A. and CHATTERGY, R. 1978. *Introduction to Nonlinear Optimization*, Elsevier Science Publishing Co. Inc.

Received 19 February 2002

Accepted 31 March 2004