Engineering Journal of the University of Qatar, Vol. 15, 2002, pp. 111-133

LQR REAL-TIME CONTROL FOR PLANTS WITH VARIABLE PARAMETERS

H.M.Al-Hamadi and **H.K.Hussien**
University of Qatar
College of Engineering
BOX 2713
Doha Qatar
Emails: helal@qu.edu.qa, hkasem@qu.edu.qa

ABSTRACT

This paper deals with real-time optimal LQR control of plants that experience sudden changes in their parameters and states. The technique is based on real-time LQR control problem. Riccati equation is solved recursively on line forward in time. The controller's gains are adjusted in real-time according to changes in system parameters and states. The technique is useful for controlling plants that undergo real-time sudden changes in their parameters and states. The approach is tested on several examples.

1. INTRODUCTION

Linear quadratic regulator (LQR) problem is a classical multivariable optimal control problem based on state space approach. It minimizes a quadratic state and control cost function. The solution of the LQR control problem involves the computation of Riccati equation [1-5]. For a system of order n, Riccati equation is a matrix equation that requires the solution of n^2 simultaneous nonlinear time-varying scalar differential equations in n^2 variables. However, utilizing symmetry, only n(n-1) equations must be solved for the same number of unknowns. For the discrete case the differential equations are difference equations for which boundary conditions at time k=0 and an arbitrary time k=N are known.

In general, the solution of Riccati equation is a difficult task due to its high order and complexity. However, relaxing the problem to time invariant controllable systems the solution tends to be constant values. Because of the importance of Riccati equation in applied mathematics, science and engineering, a number of algorithms have been proposed over the past three decades for solving it numerically. The solutions are based on solving two-point boundary value problem. These methods include conventional numerical methods such as Runge-Kutta and Linear Multi-Step methods [7,8], and unconventional numerical methods arising from optimal control theory [9-11]. Conventionally, Riccati equation is solved backward in time. Starting with boundary condition at a final time N and solving the Riccati equation backward in time eventually results in a constant solution matrix for k<<N.

Solving Riccati equation backward in time prevents the Optimal LQR gains to be evaluated iteratively in real-time. In other words, the Riccati equation is solved off line backward in time; the optimal LQR gains are evaluated off line then utilized in the feedback control loop. In this correspondence, we propose a forward in time algorithm to compute the optimal LQR gains of the control problem recursively online in real-time.

Most of the studies of optimal LQ control are based on solving Riccati equation backward in time. Fewer studies are devoted to solve Riccati equation forward in time. A forward in time solution that is based on implicit numerical scheme is presented in [12], and another based on reflexive numerical methods in [13]. An inversion state transformation is used to solve Riccati equation forward in time [6]. The primary contribution of this paper is to compute the optimal LQR gains recursively on line in real-time. This is done by developing an algorithm that produces an on line optimal control low by solving Riccati equation recursively at each loop iteration forward in time. Therefore, unlike the conventional technique that starts at boundary condition at time k=N, the algorithm starts with a boundary condition at an initial time say t=0 then iterating forward in time. Thereafter, at each loop iteration the algorithm updates optimally the gains obtained from the previous iteration until steady state values are obtained.

The results obtained in this paper present a powerful tool for optimal LQR control of systems that may undergo real-time sudden and severe changes in system states and parameters. In such case the control loop must responds promptly to the changes in real-time. The optimal gains must be computed and adjusted automatically on line. Without such on line adjustment, the system would use the old initial computed gains, which are sub-optimal and probably produce unacceptable response as demonstrated by the numerical examples in Section 4.

The paper is organized as follows. Section 2 presents LQR mathematical model and the algorithm for solving Riccati equation forward in time. Section 3 presents the real-time on line control utilizing iterative algorithm to solve Riccati equation forward in time. In Section 4, examples and simulation results are presented. The conclusion is provided in Section 5.

2. OPTIMAL LQR MATHEMATICAL MODEL

Consider the time-invariant discrete time system represented by:

$$x(k+1) = A\,x(k) + B\,u(k)$$
$$y(k) = C\,x(k) \quad (1)$$

With arbitrary given initial condition $x(0) = x_0$, and the performance index is given by:

$$J= \frac{1}{2}\sum_{k=0}^{N-1}[x^T(k)Qx(k)+u^T(k)Ru(k)]+\frac{1}{2}x^T(N)Hx(N) \qquad (2)$$

where A, B and C are n x n, n x m and m x n matrices, receptively, representing the plant to be controlled. The performance index matrices, R and Q are n x n symmetric positive definite and m x m symmetric positive semi-definite matrices, respectively.

The LQR problem is to find, for any given initial state x(0), the optimal control sequence u*(k), $0 \le k \le N$, which minimizes the quadratic performance index J and transfer the initial state x(0) to the desired equilibrium state, the origin, assuming that all external system inputs are zeros.

The optimal control u*(k) that minimizing J is [14], assuming the states are measurable:

$$u^*(k)=K(k)x(k) \qquad (3)$$

where K(k) is the optimal feedback gain defined as :

$$K(k) = -[R+B^TP(k-1)B]^{-1}\,B^TP(k-1)A \qquad (4)$$

And P(k) satisfied algebraic Riccati equation:

$$P(k)= Q+A^T P(k-1)A+A^TP(k-1)BR^{-1}B^TP(k-1)A \qquad k=1,2,\dots\dots\dots \qquad (5)$$
$$P(0)= H$$

The algorithm for developing the optimal LQR control u*(k) in forward time is presented in the following steps:

Step 1: Set

$x(0) = x_0$ an arbitrary initial state.

$P(0) = H$ initial value of Riccati gain matrix.

$u^*(0) = P(0)\,x(0)$ initial value of optimal control.

Step 2: Set $k = k + 1$ next iteration

Step 3: Evaluate
$K(k) = -[R+B^TP(k-1)B]^{-1} B^TP(k-1)A$ optimal feedback gain (Kalman Gain)

Step 4: Set : $V(k) = A + B K(k)$ intermediate step for computer programming convenience.

Step 5: *Evaluate* Riccati gain matrix
$P(k) = V^T(k)P(k-1)V(k)+K^T(k)RK(k)+Q$

Step 6: *Evaluate* $x(k) = Ax(k-1) + B u^*(k-1)$ system states

Step 7: *Evaluate* $u^*(k) = K(k) x(k)$ optimal control at stage k

Step 8: *Evaluate* $J = 0.5 x^T(k)P(k)x(k)$ cost up to stage k
GO TO **Step 2** go to next iteration.

3. REAL-TIME ONLINE CONTROL

When evaluating Riccati equation forward in time, the optimal control law is evaluated based on past and present information. Thus the requirement of predicting future states in the conventional method is avoided. The algorithm in Section 2 is employed to adaptively calculate the optimal control gains for systems that undergo real-time severe changes in the system states and parameters. In this section, we demonstrate the effect of the state and system parameters changes on the optimal control law.

Consider regulation control using the conventional LQR. The optimal Kalman gains are computed offline by solving Riccati equation backward in time. The optimal gains are then employed in the feedback loop to drive the state to equilibrium. Now consider sudden changes occur in the system states and parameters. Since the LQR gains are constant that have been calculated offline, they will not be adaptively changed in response to the new changes. Therefore, the resulting control is not necessarily optimal or not necessarily stable as illustrated by the examples. In the proposed real-time LQR algorithm the optimal LQR gains are calculated online in real-time at each time sample by solving Riccati equation forward in time. Therefore, for sudden changes in the system states and parameters the gains are calculated optimally and the system is driven to equilibrium. The price paid is in the time required for computing a single iteration of Riccati equation solution in every time sample. The cost is justified with the high-speed computing devices. The time required to calculate the optimal gains online in a single time step is approximately 0.1 m second for a system of order 3. Table 1 shows the pros and cons of the two approaches together with computation time.

Table 1. Comparison between conventional LQR and real-time LQR

	Conventional LQR	LQR Real-time
Riccati equation	Backward in time	Forward in time
Adaptive response to sudden change	No	Yes
Gains setup time (off line computation)	Yes (5 m seconds for N=50 iterations, system of order 3)*	No (Gains are calculated online)
Gains on line Competition	No (Gains are calculate offline)	Yes (0.1 m seconds for each iteration, system of order 3)*

* Computations are made on Pentium III PC 450 MHz using Turbo C++.

4. SIMULATION RESULTS

The above algorithm has been run on several examples. The examples ranged from single-input single-output systems to multiple-input multiple-output systems. The results are compared with the conventional LQR (Reccati equation solved backward in time). Figures 1, 2 and 3 show three examples. In each example the original plant is given followed by two columns to compare the conventional LQR method with the real-time LQR one. Next a sudden plant parameter change is simulated followed by two columns to compare the two methods. The numerical examples show that the real-time LQR is stable and drives the system states to equilibrium, while in the conventional LQR case the states diverge resulting in an unstable system after the change.

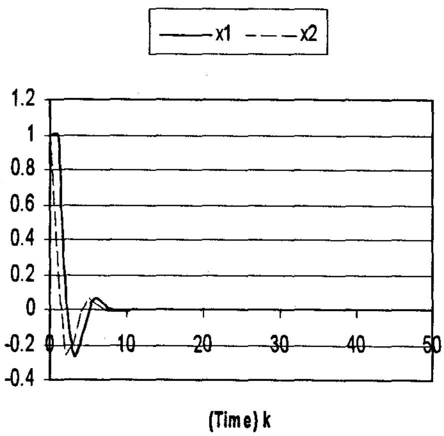
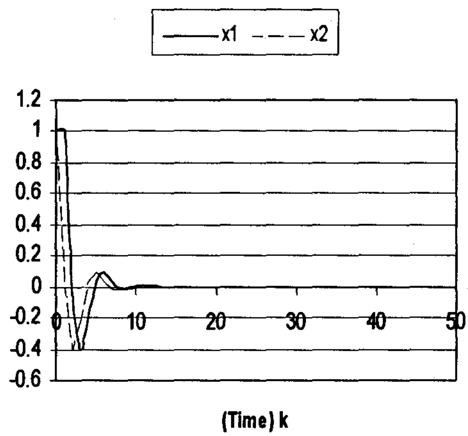
The results show the effect on the states and controls when the plant parameters change suddenly. In the conventional method the control gain is calculated off line then they are used to run the controller. They will remain the same after the system parameters and states sudden change. While in the real-time LQR method the gains are calculated optimally on line before and after the change. However, in the conventional case, the results show that the gains and the control are optimal before the change and sub-optimal after the change.

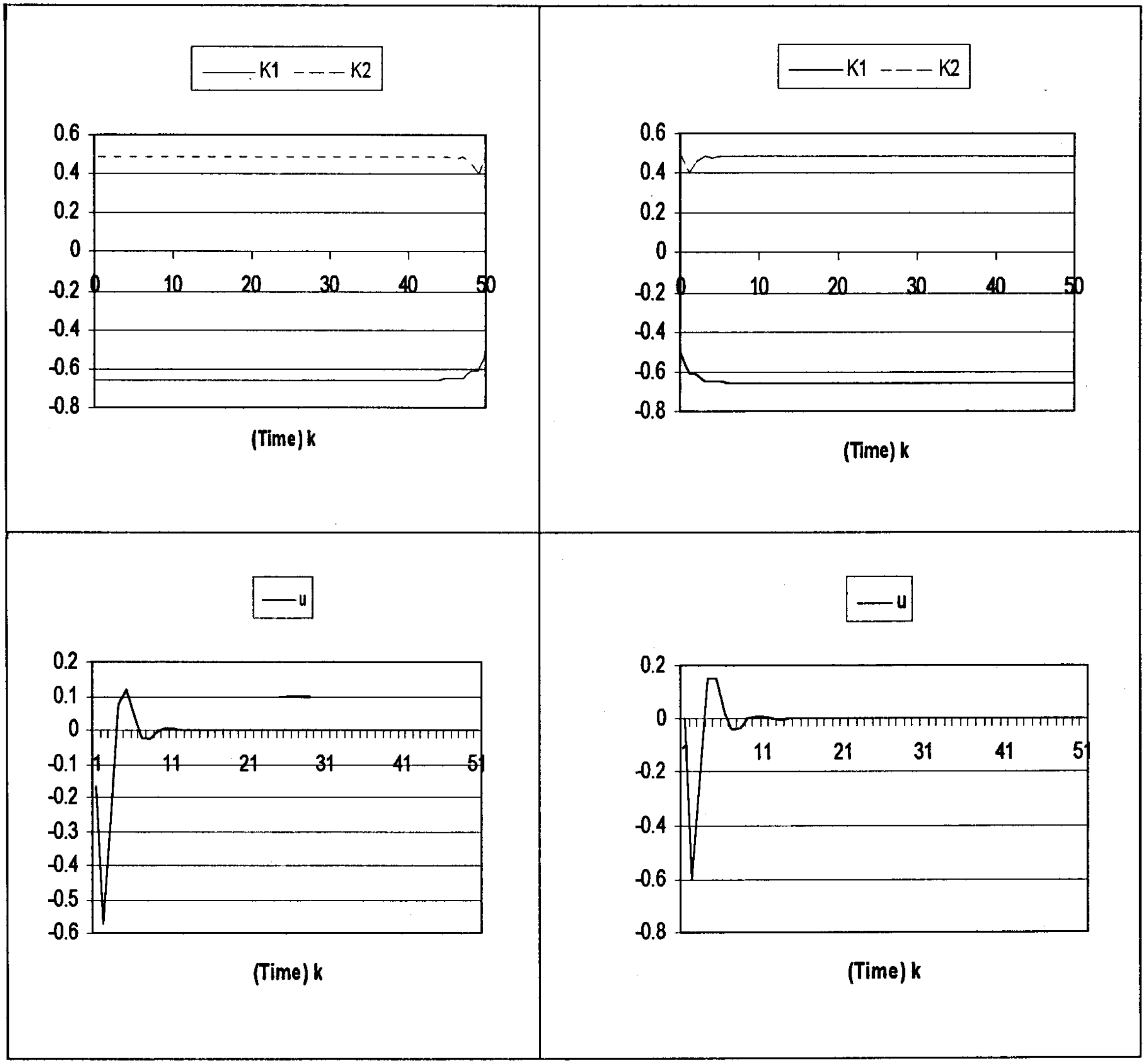
A comparison of the performance indices of the three examples is also shown. The results clearly demonstrate the superiority of the real-time LQR over the conventional

LQR in reducing the cost to minimum when system parameters vary. To compare the results with another forward in time LQR method a time varying example from [6] is simulated. The method used in [6] is based on an inversion state translation. Example 4 presents a third order two inputs time invariant system from [6]; it has been simulated using the new real-time LQR method. The simulation results shows faster convergence of our method over the method in [6]on the expense of higher control gains.

Example 1:

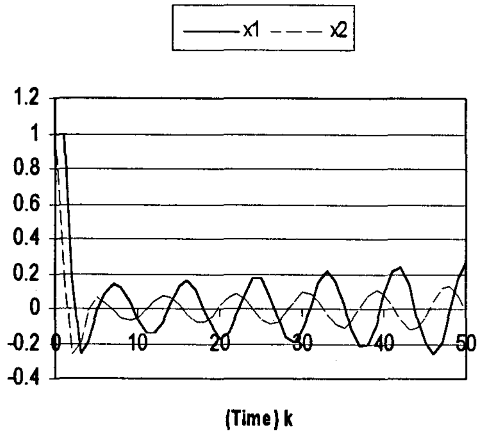
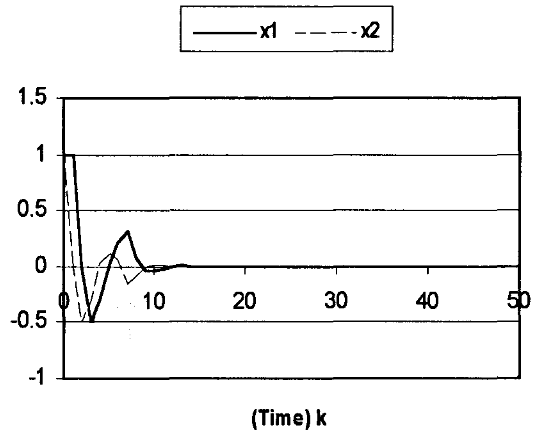
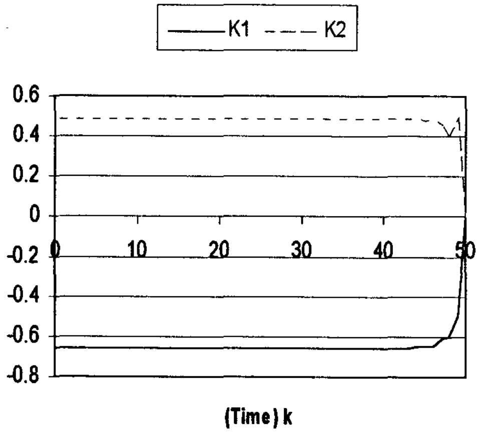
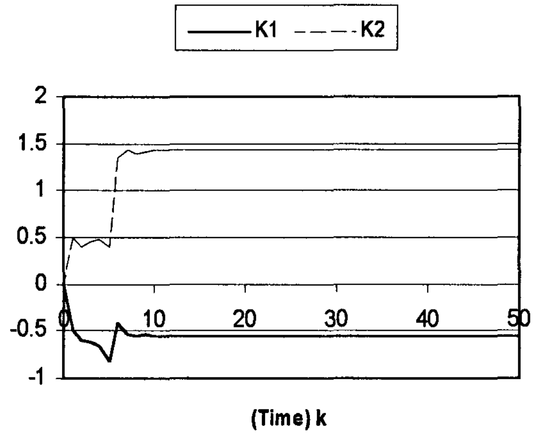
$$A=\begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix},\ B=\begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \text{Initial Condition } x(0)=\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

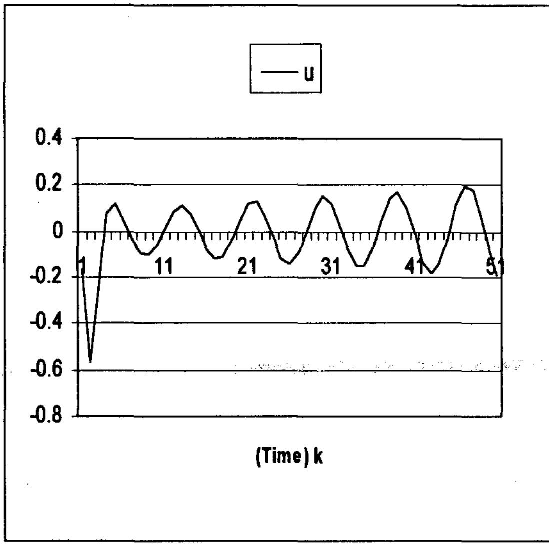
Conventional LQR	LQR Real-time
	

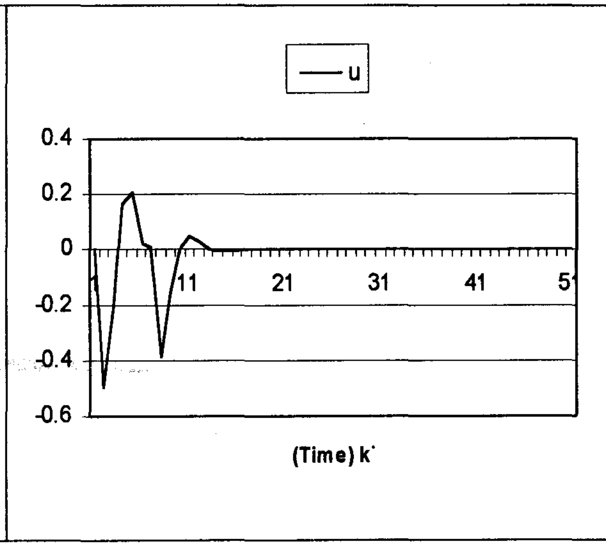

K1
K2
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
10
20
30
40
50
(Time) k
K1
K2
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
10
20
30
40
50
(Time) k
u
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
1
11
21
31
41
51
(Time) k
u
0.2
0
-0.2
-0.4
-0.6
-0.8
1
11
21
31
41
51
(Time) k

Example 1: (Change System Parameters at k=5 sec)

A is set at k= 5 sec to be $A=\begin{bmatrix}1 & 1.5\\-1 & 1\end{bmatrix}$, Initial Condition x(0)=$\begin{bmatrix}1\\1\end{bmatrix}$

Conventional LQR	LQR Real-time
	
	

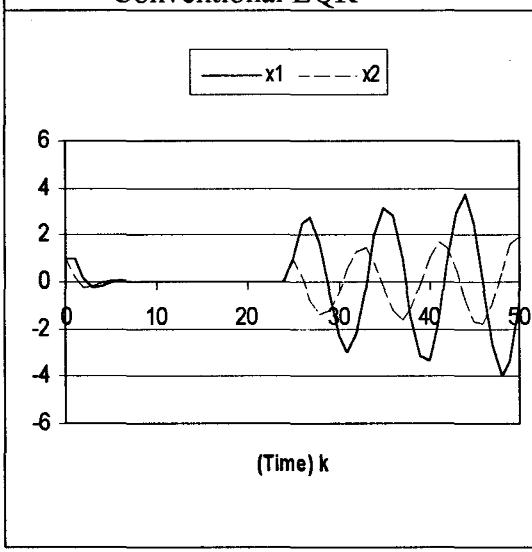


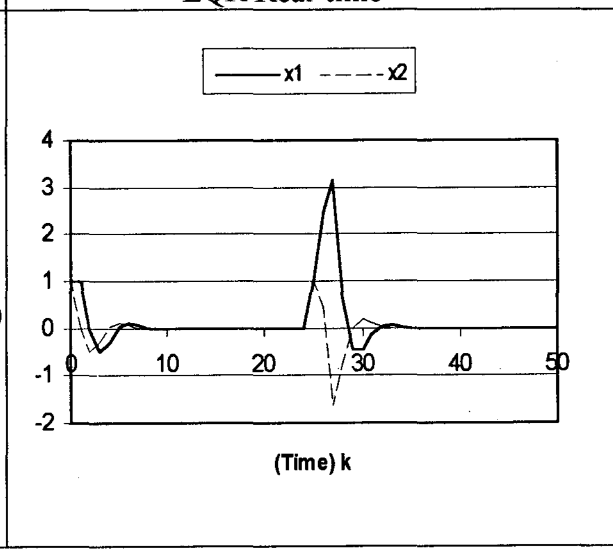


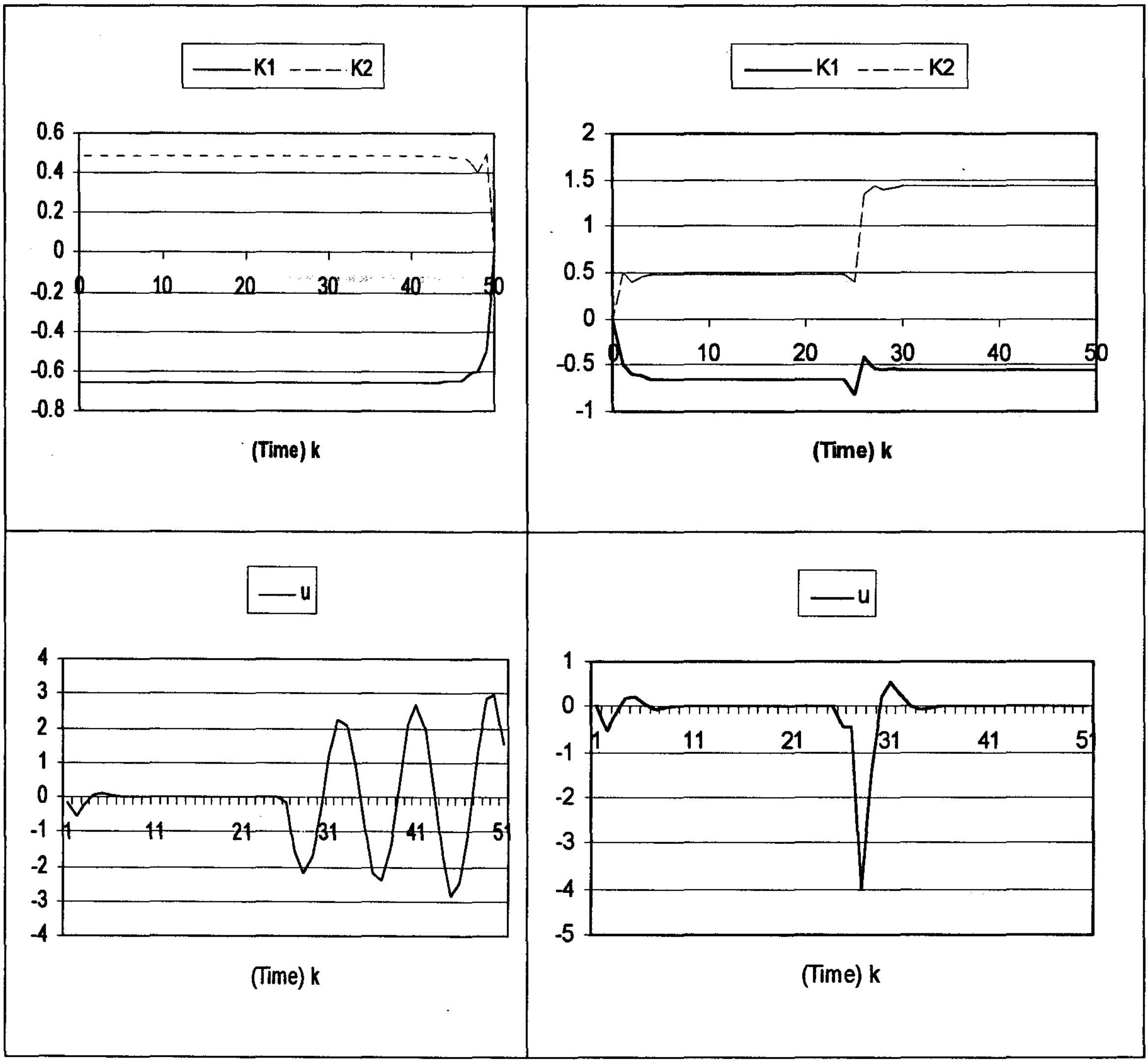
Example 1: (Change State and System Parameters at k=25 sec)

A is set at k= 25 sec to: $A = \begin{bmatrix} 1 & 1.5 \\ -1 & 1 \end{bmatrix}$, state $x(25) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Conventional LQR	LQR Real-time

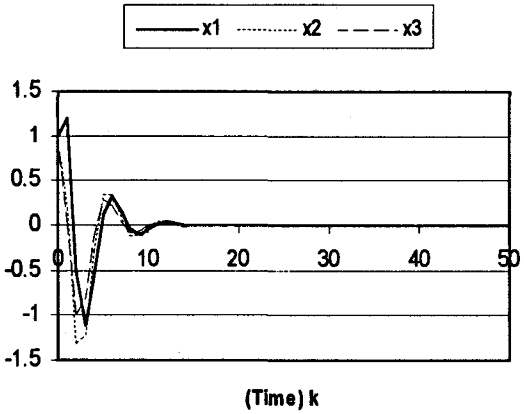
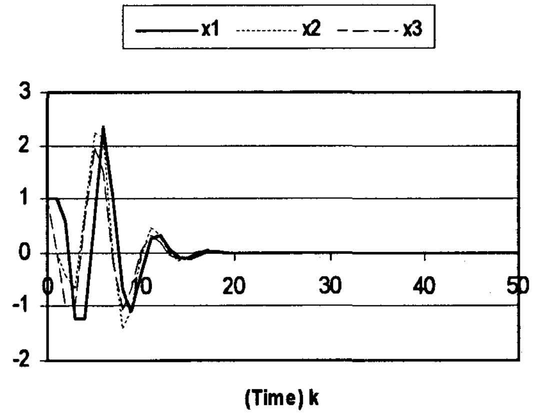
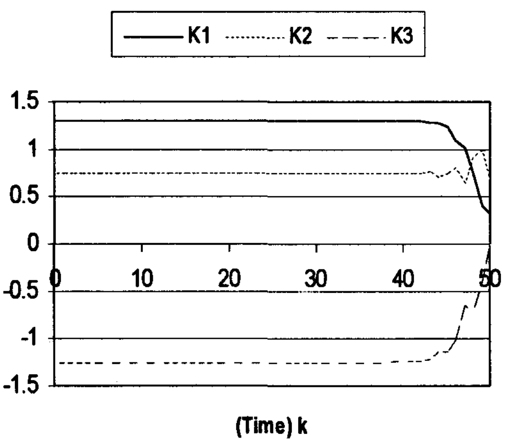
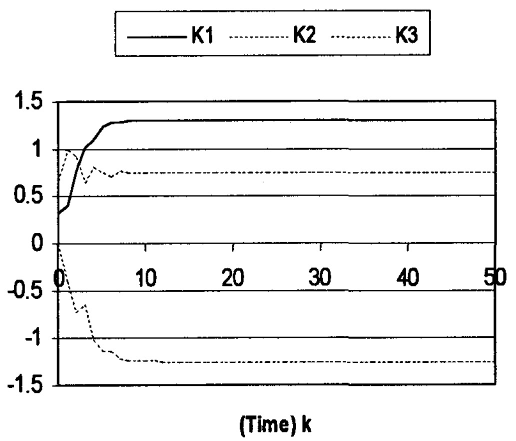


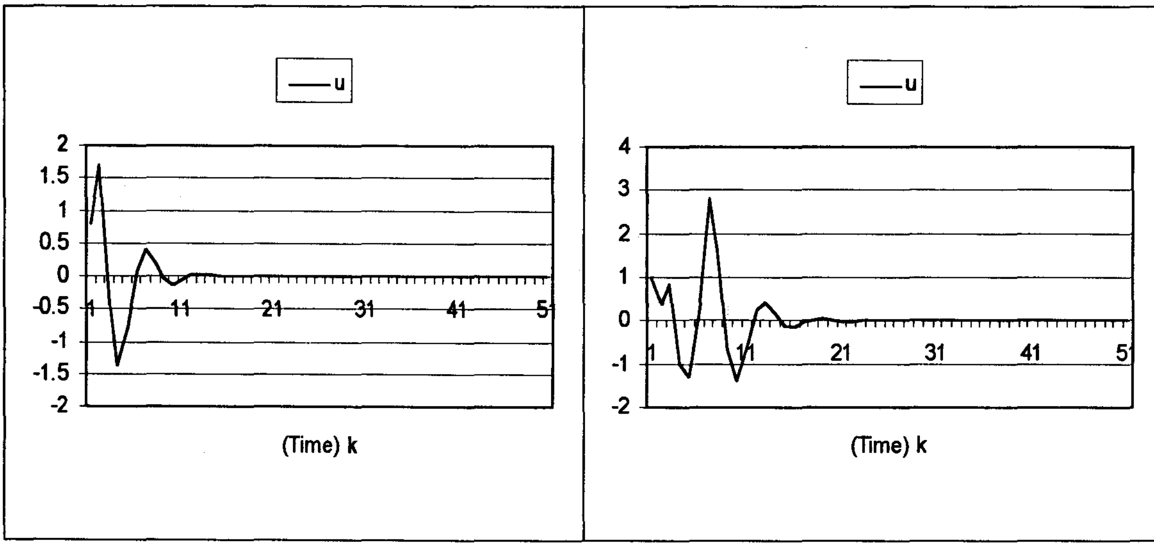



K1
K2
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
10
20
30
40
50
(Time) k
2
1.5
1
0.5
0
-0.5
-1
(Time) k
u
4
3
2
1
-1
-2
-3
-4
1
11
21
31
41
51
(Time) k
-5
(Time) k

Example 2

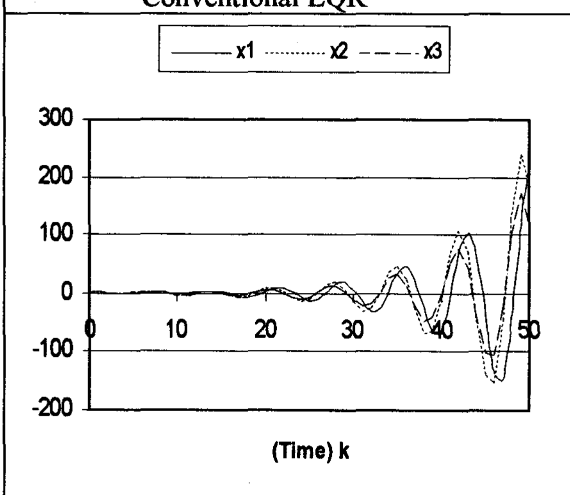
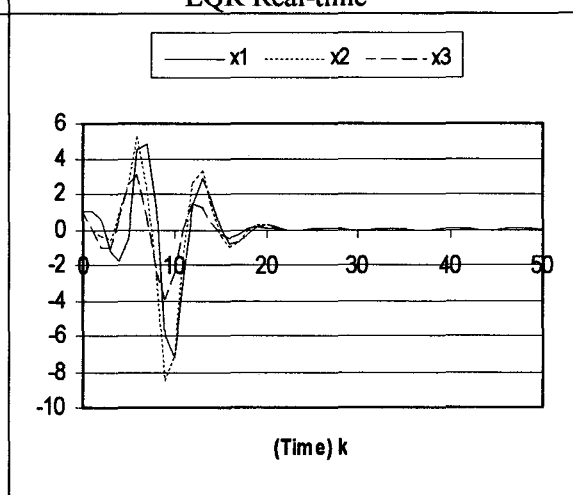
$$A=\begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & -1 \\ -1 & 1 & 0 \end{bmatrix},\ B=\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix},\ \text{Initial Condition } x(0)=\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

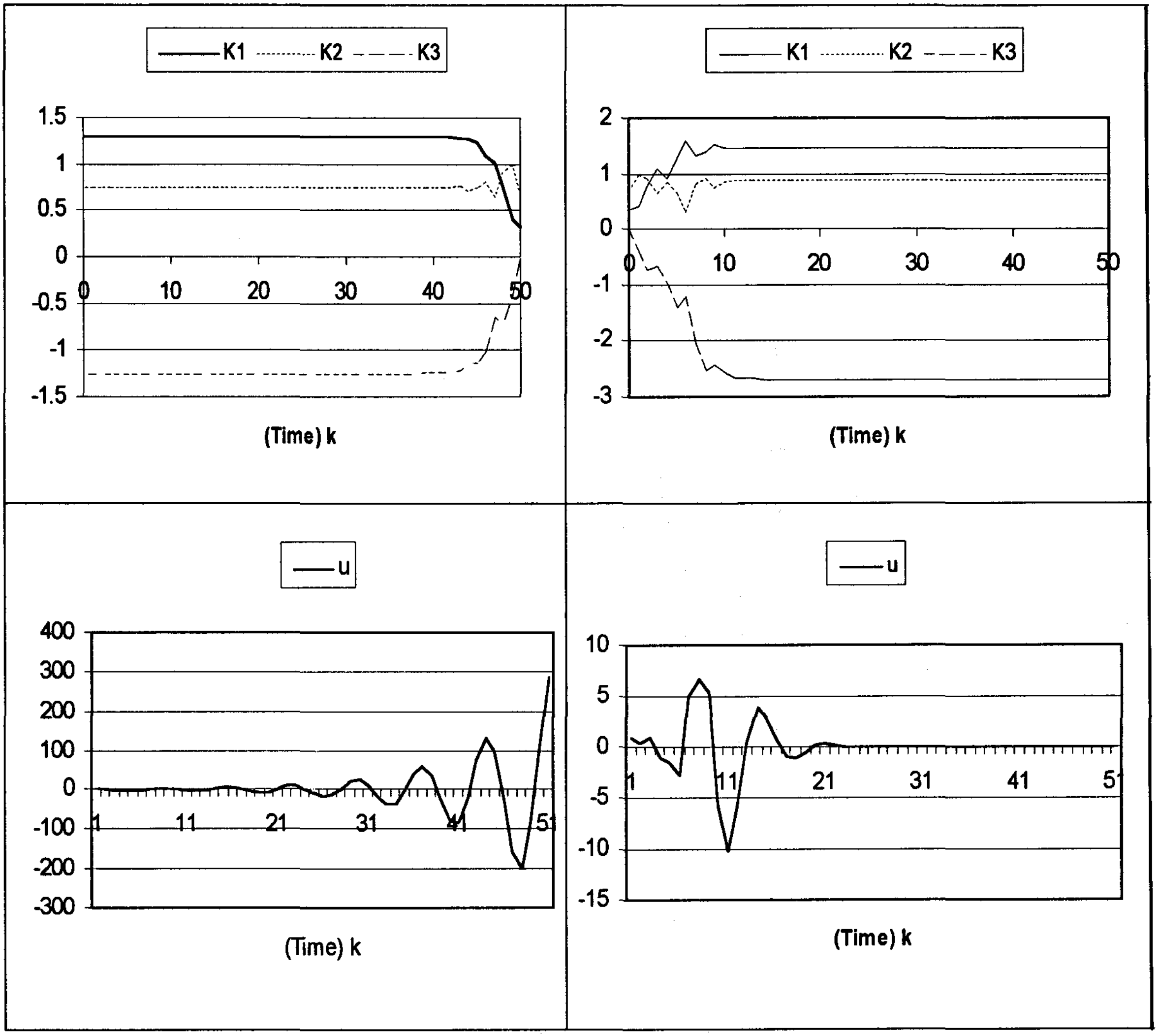
Conventional LQR	LQR Real-time
	
	



Example 2: (Change system parameters at k=3 sec)

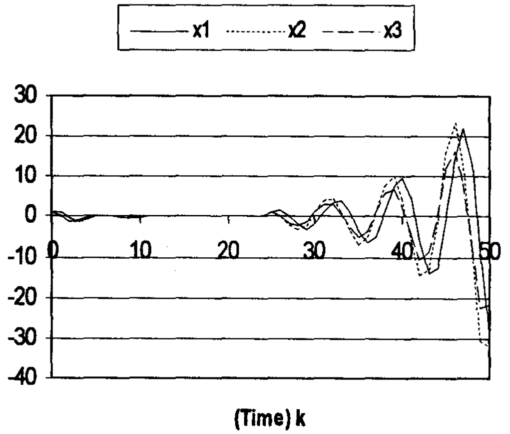
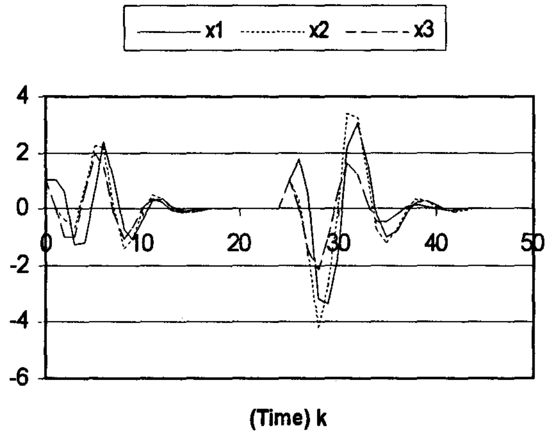
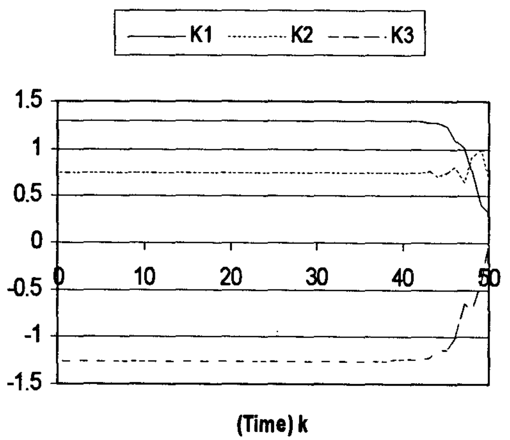
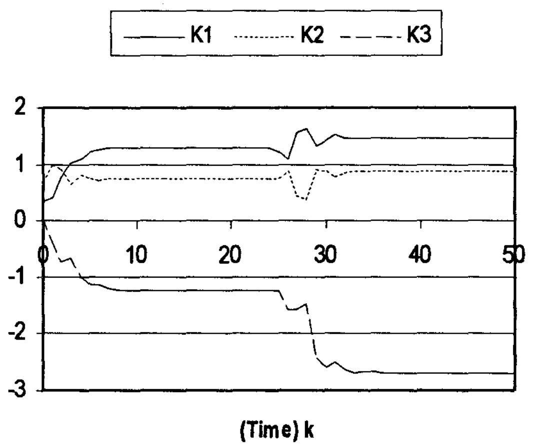
A is set at k= 3 sec to: $A = \begin{bmatrix} 1.5 & 0 & 1 \\ 0 & 2 & -1 \\ -1 & 1 & 0 \end{bmatrix}$

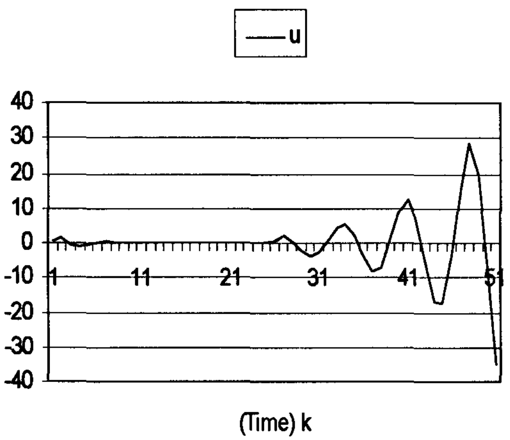
Conventional LQR	LQR Real-time
	

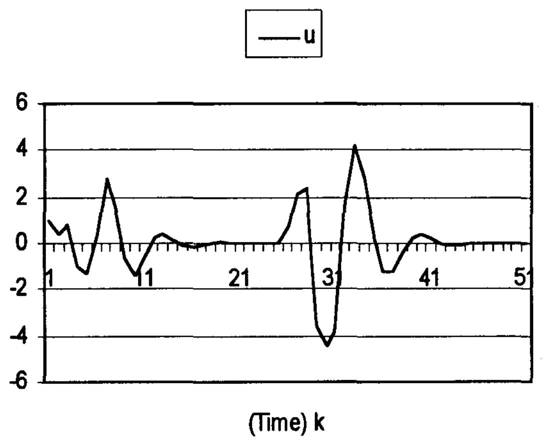

K1 K2 K3
1.5
1
0.5
0
-0.5
-1
-1.5
0 10 20 30 40 50
(Time) k
K1 K2 K3
2
1
0
-1
-2
-3
0 10 20 30 40 50
(Time) k
u
400
300
200
100
0
-100
-200
-300
1 11 21 31 41 51
(Time) k
u
10
5
0
-5
-10
-15
1 11 21 31 41 51
(Time) k

<u>Example 2:</u>(Change State and System Parameters at k=25 sec)

A is set at k= 25 sec to: $A = \begin{bmatrix} 1.5 & 0 & 1 \\ 0 & 2 & -1 \\ -1 & 1 & 0 \end{bmatrix}$, state $x(25) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

Conventional LQR	LQR Real-time
	
	

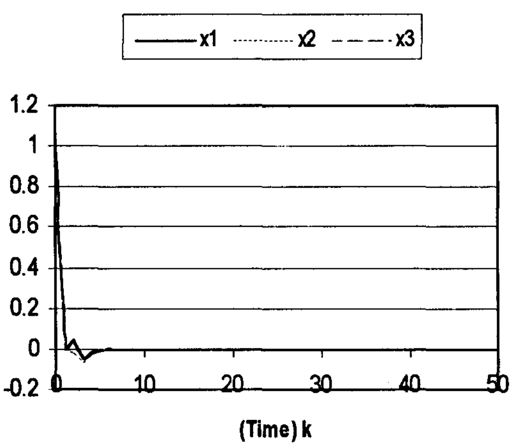




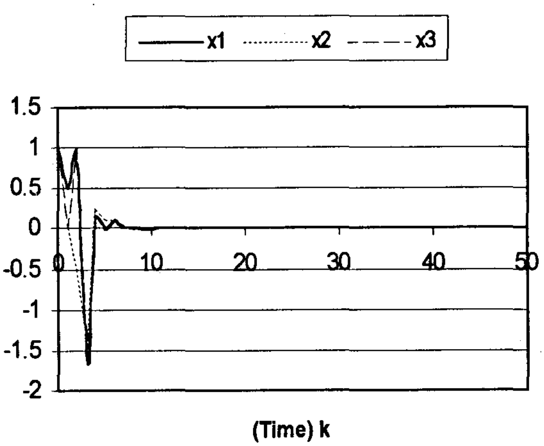
Example 3:

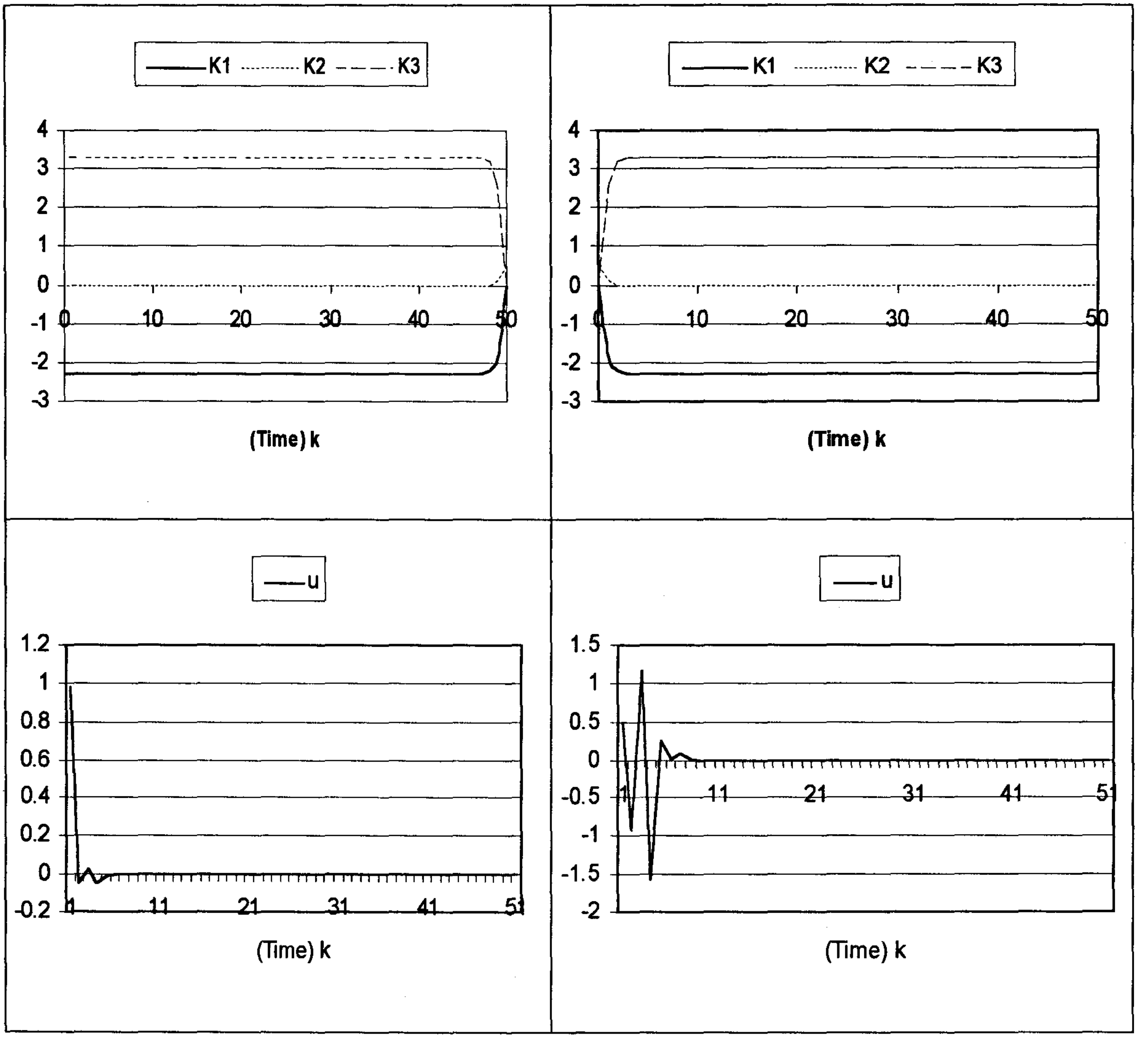
$$A=\begin{bmatrix}0 & 1 & 0\\ -1 & 1 & 0\\ 2 & 1 & -3\end{bmatrix},\ B=\begin{bmatrix}1\\0\\0\end{bmatrix},\ \text{Initial Condition } x(0)=\begin{bmatrix}1\\1\\1\end{bmatrix}$$

Conventional LQR



LQR Real-time





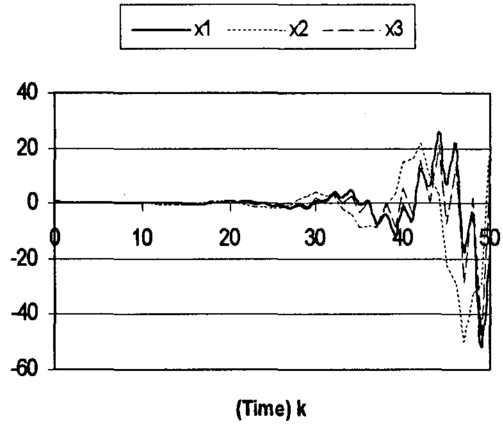
Example 3: (Change System Parameters at k=3 sec)

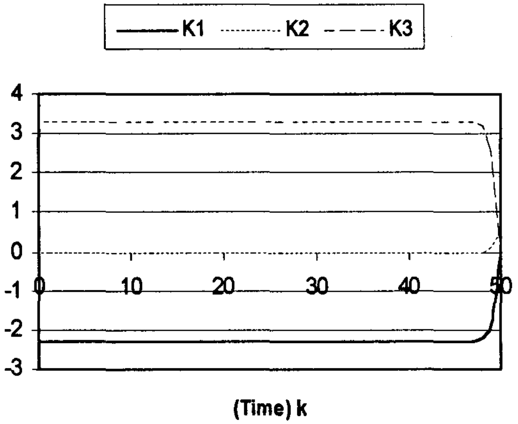
A is set at k= 3 sec to: $A = \begin{bmatrix} 5 & 1 & 0 \\ -1 & 1 & 0 \\ 2 & 1 & -3 \end{bmatrix}$

LQR Real-Time Control for Plants With Variable Parameters

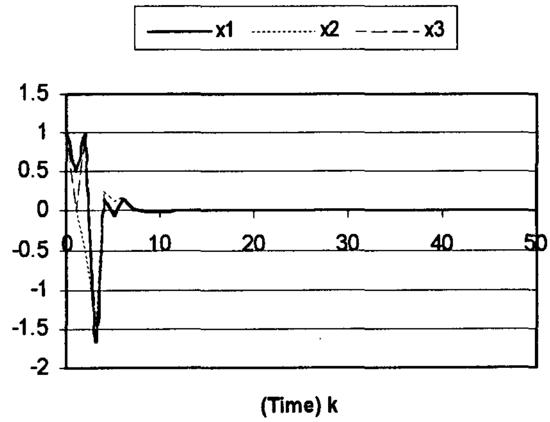
Conventional LQR	LQR Real-time

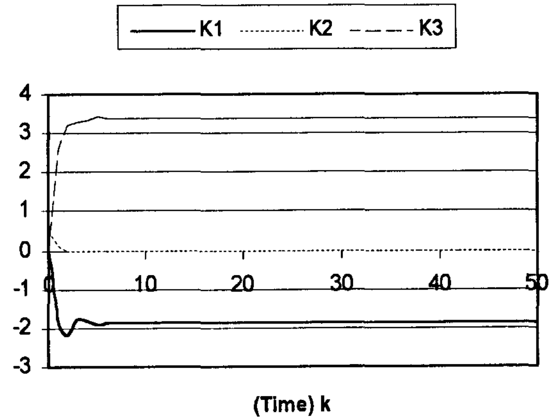
Conventional LQR

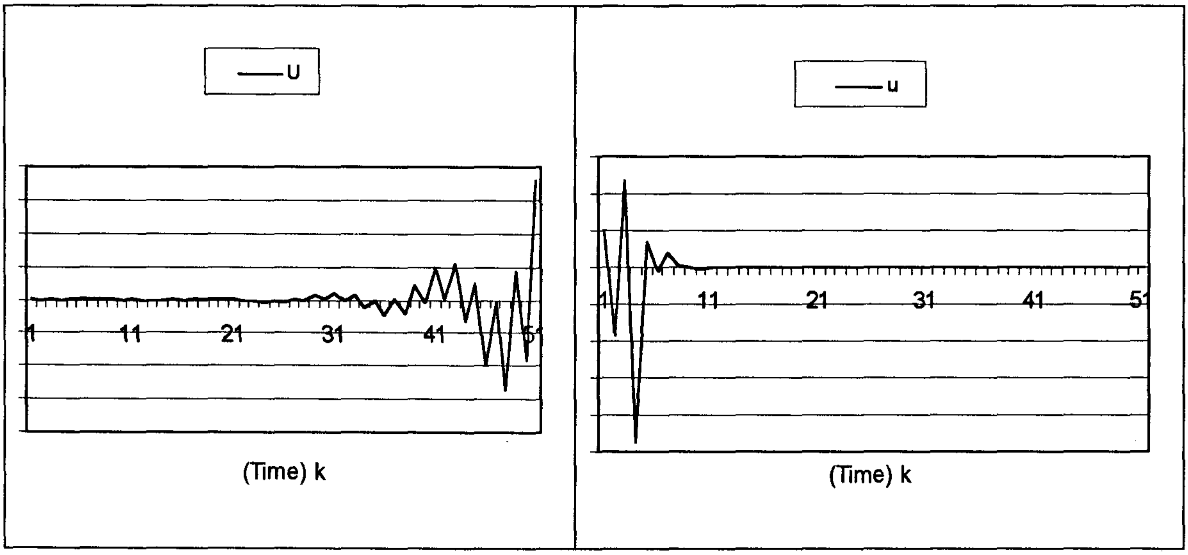




LQR Real-time



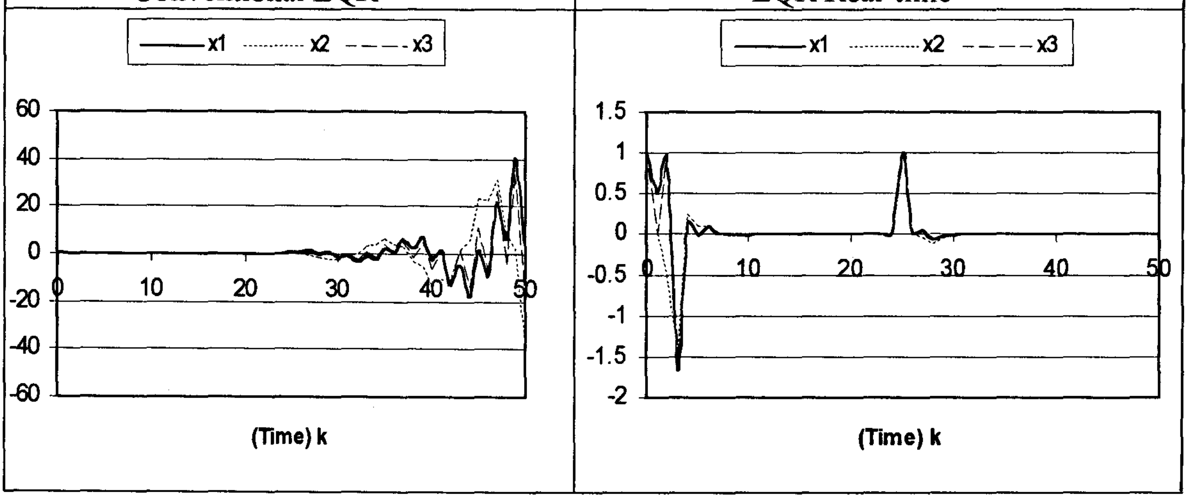


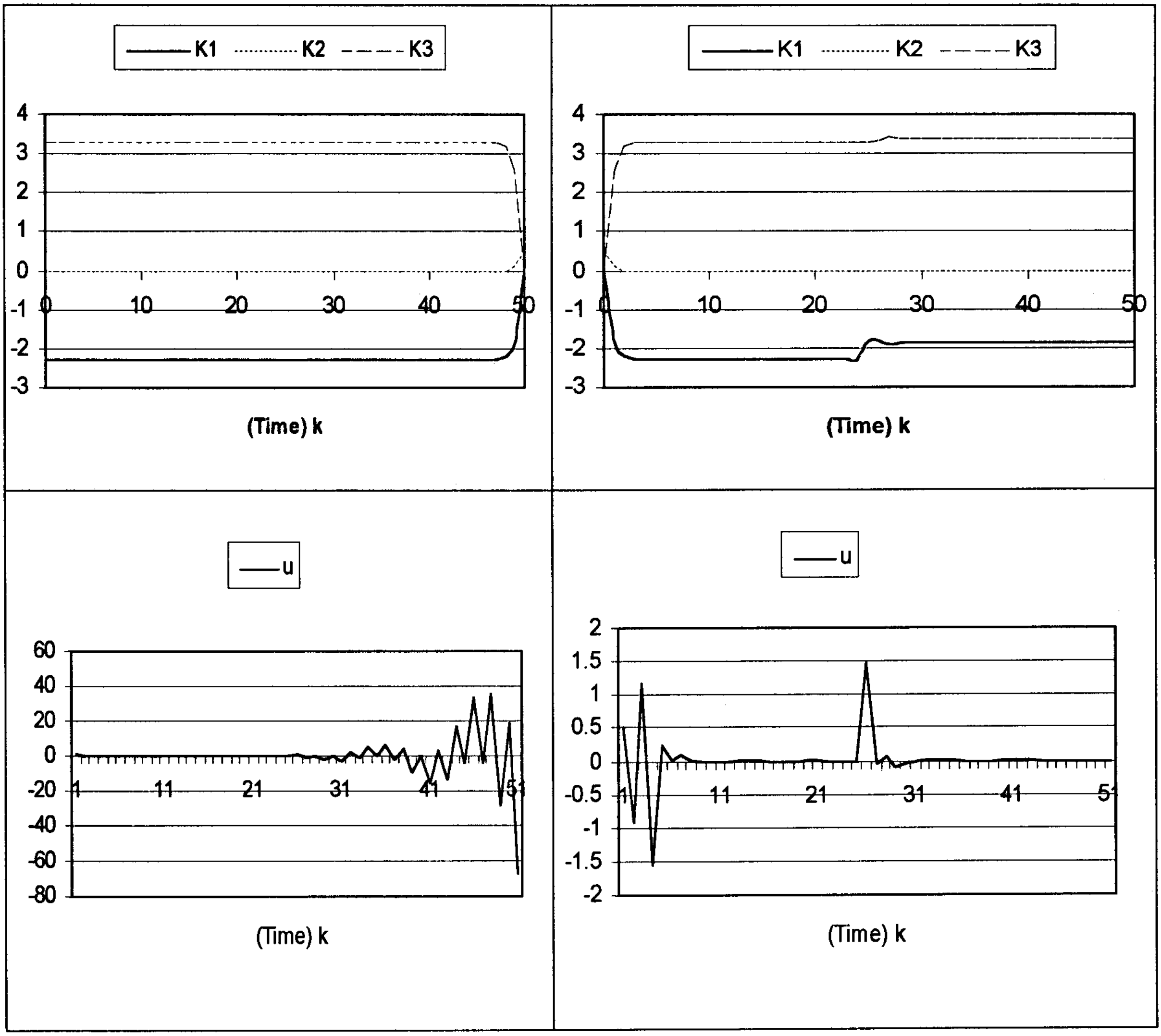


Example 3: (Change A and IC at k=25 sec)

A is set at k= 25 sec to: $A = \begin{bmatrix} 5 & 1 & 0 \\ -1 & 1 & 0 \\ 2 & 1 & -3 \end{bmatrix}$, States $x(25) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

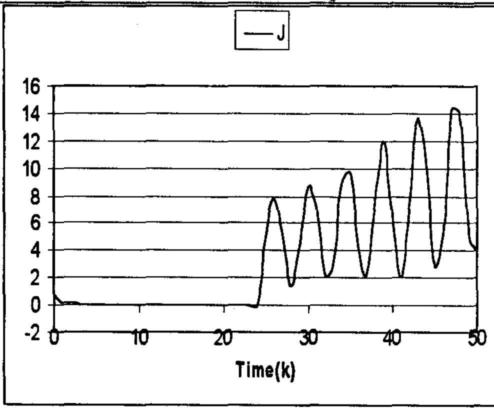
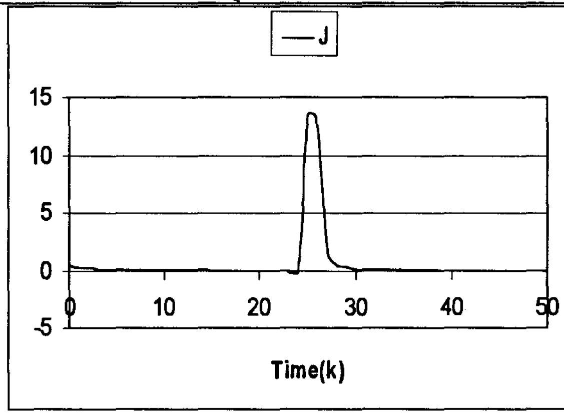
Conventional LQR	LQR Real-time





<u>Example 1 Continued:</u> Comparing Cost function. (Change State and System Parameters at k=25 sec)

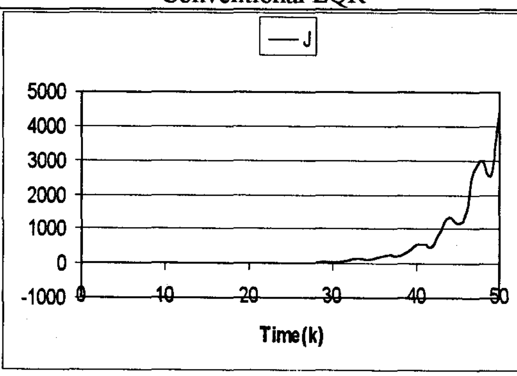
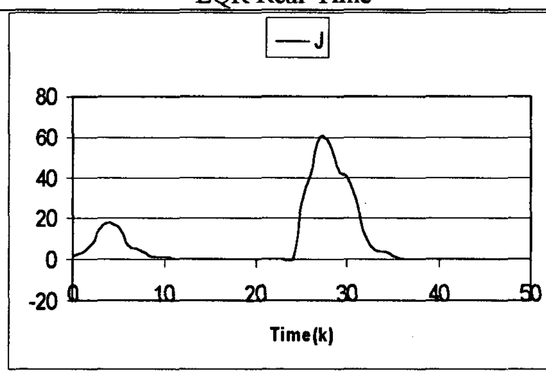
A is set at k= 25 sec to: $A=\begin{bmatrix}1 & 1.5\\ -1 & 1\end{bmatrix}$, state $x(25)=\begin{bmatrix}1\\ 1\end{bmatrix}$, Q= $\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, R= 1

Conventional LQR	LQR Real-Time
	

<u>Example 2 Continued:</u> Comparing Cost function. (Change State and System Parameters at k=25 sec)

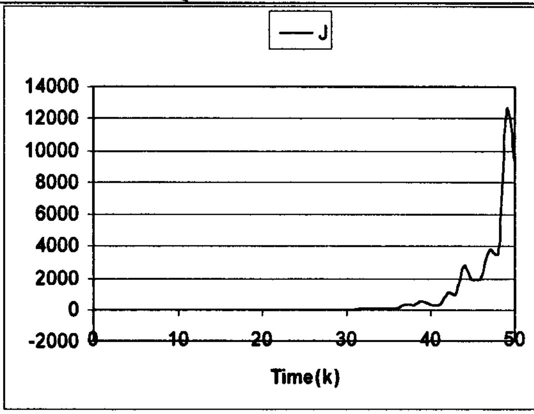
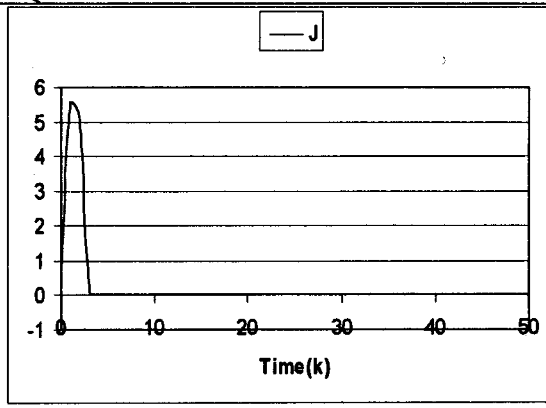
A is set at k= 25 sec to: $A=\begin{bmatrix}1.5 & 0 & 1\\ 0 & 2 & -1\\ -1 & 1 & 0\end{bmatrix}$, state $x(25)=\begin{bmatrix}1\\1\\1\end{bmatrix}$, Q=$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$,

R= 1

Conventional LQR	LQR Real-Time
	

<u>Example 3 Continued:</u> Comparing Cost function: (Change A and IC at k=25 sec)

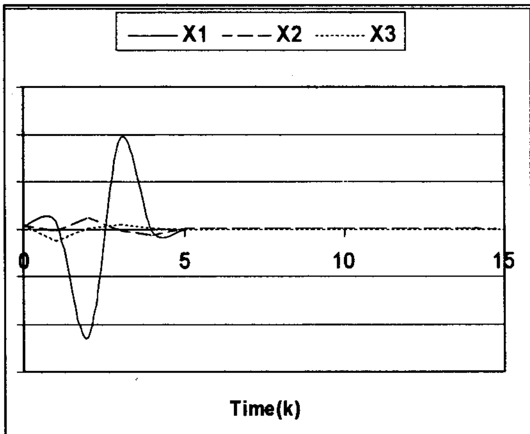
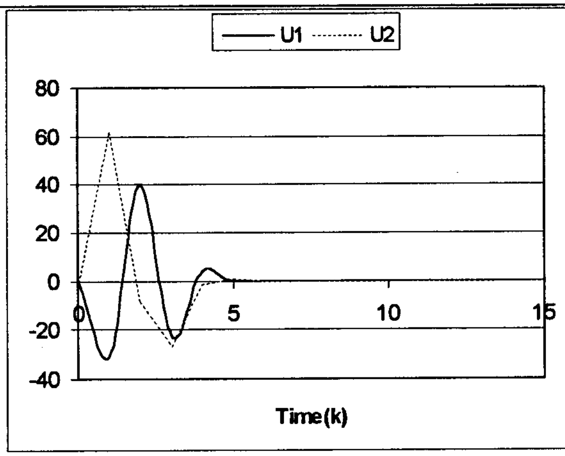
A is set at k= 25 sec to: $A=\begin{bmatrix}0.5 & 1 & 0\\ -1 & 1 & 0\\ 2 & 1 & -3\end{bmatrix}$, States $x(25)=\begin{bmatrix}1\\1\\1\end{bmatrix}$, Q=$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$, R= 1

Conventional LQR	LQR Real-Time
	

<u>Example 4.</u> Third order, two inputs, time varying system.

$$A(t)=\begin{bmatrix} -1+1.5\cos^2 t & 1-1.5\cos t\sin t & 1 \\ -1-1.5\sin t\cos t & -1+1.5\sin^2 t & 1 \\ 0 & -\sin t & -5+\sin t \end{bmatrix}, B(t)=\begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 0 & 1 \end{bmatrix}, \text{ I.C. } x^T(0)=[3,3,3]$$

$P(0) = Q(0)$, $Q(t) = I_{3x3}$ and $R(t) = I_{2x2}$

5. CONCLUSION

Real-time LQR control problem is presented with an algorithm for solving Riccati equation forward in time. Kalman gains are evaluated on line in real-time. The results demonstrate the effectiveness of adaptively controlling the plant by evaluating the gains on-line. This is useful for applications in which system states and parameters undergo repeatedly on-line sudden changes. This technique is superior to the conventional LQR in the sense that it evaluates the optimal gains optimally in real-time. In the conventional case the gains are evaluated optimally off line for the original plant, they will not be optimal after on-line sudden changes in system parameters and states.

REFERENCES

1. **R. E. Kalman, 1960.** Contribution to the theory of optimal control. Bol. Soc. Matem. Mex, pp 102-119.

2. **R. E. Kalman** and **R. S. Bucy, 1961.** New results in linear filtering and prediction theory. ASME J. Basic Engnrg. Ser. D, 83 pp 95-108.

3. **R. E. Kalman, 1960.** A new approach to linear filtering and prediction problems. ASME J. Basic Engnrg. Ser. D, 82 pp35-45.

4. **Kalman, R. E. 1964.** When is a linear control system optimal? ASME Journal of basic engineering, pp. 51-60.

5. **Kwakernaak H.** and **Sivan, R., 1972.** Linear optimal control systems, Wiley, New York.

6. **Min-shin Chan** and **Chung-Yao Kao, 1996.** Control of linear time-varying system using forward Riccati equation. Transaction of the ASME, Journal of Dynamics Systems, Measurement and control Vol. 119, no.3 pp 536-40.

7. **C. H. Chot** and **A. J. Laur, 1990.** Efficient matrix-valued algorithm for solving stiff Riccati differential equations, IEEE Transactions on Automatic Control, 35, pp. 770-776.

8. **L.Dieci, 1992.** Numerical integration of the differential Riccati equation and some related issues, SIAM Journal on Numerical Analysis, 29, pp. 781-815.

9. **C. S. Kenney** and **R .B. Leopink, 1985.** Numerical integration of the differential matrix Riccati equation, IEEE Transitions on Automatic Control, AC-30, pp. 962-970.

10. **D. G. Lainiotis, 1976.** Partitioned Riccati solutions and integration-free doodling algorithm, IEEE Transitions on Automatic Control, AC-21, pp. 728-732.

11. **I. Rusnak, 1988.** Almost analytic representation for the solution of the differential matrix Riccati equation, IEE Transitions Automatic Control, 33, pp. 191-193.

12. **Francois Dudois** and **Abdelkader Saidi 2000**, Unconditionally stable scheme for Riccati equation, ESAIM Proceedings. Vol. 8.2000, pp. 39-52.

13. **Ren-Cang Li, 2000**. Unconditionally Reflexive Numerical Methods for Matrix Differential Riccati equation, Lexington , Technical Report 2000-36 .

14. **Kailath,T. 1980,** Linear systems, Prentice Hall, Englewood Cliffs, New Jersey.