

# $H^\infty$ -Optimal Control for Robust Financial Asset and Input Purchasing Decisions

David Hudgins, Joon Na

Department of Economics, 323 Cate Center 1, University of Oklahoma, Norman, USA  
Email: [hudgins@ou.edu](mailto:hudgins@ou.edu)

Received April 22, 2013; revised June 19, 2013; accepted June 30, 2013

Copyright © 2013 David Hudgins, Joon Na. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## ABSTRACT

This analysis formulates an approach for converting minimax LQ (linear-quadratic) tracking problems into LQ regulator designs, and develops a Matlab application program to calculate an  $H^\infty$  robust control for discrete-time systems with perfect state measurements. It uses simulations to explore examples in financial asset decisions and utility input purchasing, in order to demonstrate the method. The user is allowed to choose the parameters, and the program computes the generalized Riccati Equation conditions for the existence of a saddle-point solution. Given that it exists, the program computes a minimax solution to the linear quadratic (LQ) soft-constrained game with constant coefficients for a general scalar model, and also to a class of matrix systems. The user can set the bound to achieve disturbance attenuation.

**Keywords:**  $H^\infty$ -Optimal Control; Minimax Design; Linear-Quadratic (LQ) Methods

## 1. Introduction

When addressing any economic, finance, or engineering problem where there is uncertainty, there must be some approaches for modeling the disturbances. The use of  $H^\infty$ -optimal control modeling was the primary subject of control systems research in the field of engineering in the 1980s and 1990s. Many of the results of the research for continuous and discrete time systems with different information structures are presented in [1].  $H^\infty$ -control, which is a minimax approach, seeks to achieve a robust design by minimizing a performance index under the worst possible disturbances, where the disturbances maximize that same performance index. In this case, robust means guaranteed performance for any disturbance sequence that satisfies the  $H^\infty$ -norm bound.

This is an alternative to the approaches that were primarily advanced in the 1970s. Most of that research involved the linear-quadratic-Gaussian (LQG) approach where the disturbances were treated as random and modeled with a Gaussian distribution [2,3], or with adaptive control, where the unknown disturbances are learned through feedback and feed forward loops [4]. The problem with these probabilistic approaches, such as the LQG, is that minimizing the expected value of a performance index leads to maximum system performance in the ab-

sence of misspecification, but it leads to poor performance and instability under small or large misspecifications [5]. Hence, there is a need for robust design.

In contrast to its widespread use in the engineering area,  $H^\infty$ -control has not been widely applied in economics and finance. One of the problems is that the system must be formulated properly with an appropriately accurate model of the system; otherwise, optimizing the wrong controller makes performance worse instead of better. Another problem with applying the  $H^\infty$ -approach to macroeconomic problems is that many analyses model the interdependent prices and agent's decisions separately, so that the existing  $H^\infty$ -methods cannot be applied [5]. Another problem has been the modeling and computational difficulty.

There have been some economic applications, such as [6-8]. The robust method in [9] has been widely used in the area of macroeconomics. However, that method concentrates on applying entropy as a distance measure to bound uncertainty. Its examination of  $H^\infty$ -optimal control is limited to the frequency domain where it can be compared to the entropy approach to robust control.

The model in [5] applies the  $H^\infty$ -control approach to a model where agents seek robust consumption and portfolio strategies to deal with misspecifications, rather than misperceptions. He finds that  $H^\infty$ -forecasts are more sen-

sitive to news than rational expectation forecasts, since robust agents behave as if they misperceive shocks to have more persistence than they actually do. The model in [5] also provides an explanation for some well-known asset pricing anomalies, including the predictability of excess returns, excess volatility, and the equity-premium puzzle. The analysis in [5] performs simulations that demonstrate that when robustness is high, prices are more volatile than dividends.

These findings add further impetus for the use of a robust approach in many areas of finance and economics. For example, it would be useful in asset and options pricing in financial markets, where common approaches, such as the Black-Scholes methodology, are sensitive to assumptions regarding the statistical distribution of the disturbances and to extreme price changes. However, [10] cautions that robust control does not imply more cautious responses than those obtained under an uncertain model. They can be more aggressive, as in the optimal monetary policy approaches of [11], or they can respond less aggressively to incoming news, as in [12].

The purpose of this paper is to model economic and financial applications using a discrete-time  $H^\infty$ -approach, and to develop Matlab software programs that allow users to simulate optimal solutions under a flexible choice of system parameters. The paper uses the minimax optimization approach to some fundamental economic problems where it has not previously been considered, and then compares the simulated solutions to those solutions that would have been obtained under deterministic optimal control strategies with varying parameters and terminal conditions. We develop a matrix application and a scalar application program. These programs allow the analyst to make better decisions by exploring the worst-case disturbance strategy in conjunction with other approaches to dynamic systems that contain disturbances and uncertainty.

## 2. Model Derivation

This analysis considers the discrete-time minimax controller design problem with perfect state measurements. The problem is formulated in expression (1) as a soft-constrained linear-quadratic (LQ) game, which is a generalized version of that in [1]. The controller  $u$  is the minimizing player and the disturbance term  $w$  is the maximizing player.

$$\min_u \max_w J_\gamma(u, w) = x_{K+1}^T Q_f x_{K+1} + \sum_{k=1}^K [x_k^T Q_k x_k + u_k^T R_k u_k - r w_k^T w_k] \tag{1}$$

subject to

$$x_{k+1} = A_k x_k + B_k u_k + D_k w_k; \quad x(1) = x_1 \tag{2}$$

where the initial value of the state vector at the initial time  $k=1$  is fixed at  $x(1) = x_1$ , and  $r = \gamma^2 > 0$ ;  $R_k > 0$ ;  $Q_f, Q_k \geq 0$ ;  $k = 1, 2, \dots, K$ .

The sizes of the matrices and vectors are as follows:

$Q_f, Q_k, A_k$  are  $n \times n$ ;  $R_k$  is  $m \times m$ ;  $B_k$  is  $n \times m$ ;  $D_k$  is  $n \times p$ ;  $x_k$  is  $n \times 1$ ;  $u_k$  is  $m \times 1$ ;  $w_k$  is  $p \times 1$

First, consider an open-loop information structure. As shown in [13], and later applied in [1], the game admits a unique saddle-point solution trajectory  $(x_k^*, u_k^*, w_k^*)$  if and only if

$$rI - D_k^T S_{k+1} D_k > 0, \quad k = 1, 2, \dots, K; \tag{3}$$

where  $S_k, k \in [1, K]$ , is computed by the Riccati Equation expressed in Equation (4).

$$S_k = Q_k + A_k^T S_{k+1} A_k + A_k^T S_{k+1} D_k [rI - D_k^T S_{k+1} D_k]^{-1} D_k^T S_{k+1} A_k; \tag{4}$$

$$S_{K+1} = Q_f;$$

Assuming that the condition given in (3) has been met, the solution trajectory is found as follows. Define the recursive sequence of matrices  $M_k$  and  $\Lambda_k, k \in [1, K]$ , where  $\Lambda_k$  is invertible, as

$$M_k = Q_k + A_k^T M_{k+1} \Lambda_k^{-1} A_k; \quad M_{K+1} = Q_f; \tag{5}$$

and

$$\Lambda_k = I + (B_k B_k^T - r^{-1} D_k D_k^T) M_{k+1} \tag{6}$$

Equations (5) and (6) can also be combined so that they can equivalently be expressed by Equation (7) as

$$M_k = Q_k + A_k^T (M_{k+1}^{-1} + B_k B_k^T - r^{-1} D_k D_k^T)^{-1} A_k; \tag{7}$$

$$M_{K+1} = Q_f$$

The unique saddle-point optimal control policy is given by

$$u_k^* = -B_k^T M_{k+1} \Lambda_k^{-1} A_k x_k^* \tag{8}$$

The unique saddle-point worst-case disturbance trajectory is

$$w_k^* = r^{-1} D_k^T M_{k+1} \Lambda_k^{-1} A_k x_k^* \tag{9}$$

and the state trajectory is

$$x_{k+1}^* = \Lambda_k^{-1} A_k x_k^*; \quad x_1^* = x_1; \tag{10}$$

The resulting saddle-point value of the game is calculated as

$$J_\gamma^*(u, w) = x_1^T M_1 x_1 \tag{11}$$

Next, consider the case where both players have access to closed-loop state information with memory. Reference [1] shows that a unique feedback saddle-point solution exists if, and only if,

$$rI - D_k^T M_{k+1} D_k > 0, \quad k = 1, 2, \dots, K \quad (12)$$

If the condition in (12) is satisfied, then the matrices  $\Lambda_k, k \in [1, K]$ , are invertible, and the unique saddle-point control, disturbance, and state trajectories are given by

$$u_k^* = -B_k^T M_{k+1} \Lambda_k^{-1} A_k x_k \quad (13)$$

$$w_k^* = r^{-1} D_k^T M_{k+1} \Lambda_k^{-1} A_k x_k \quad (14)$$

$$x_{k+1}^* = \Lambda_k^{-1} A_k x_k^*; \quad x_1^* = x_1 \quad (15)$$

The saddle-point value of the game under closed-loop information is given by Equation (11), since it is the same as it was under open-loop information. If the matrix given by Equation (12) is not positive definite, and thus has one or more negative eigenvalues, then the game does not have a saddle-point solution, and its upper value is unbounded [1].

For disturbance attenuation, the value  $\gamma^*$  is chosen for  $\gamma$ , where  $r = \gamma^2 > 0$ , such that  $\gamma^*$  is the infimum, *i.e.*, the smallest value of  $\gamma$  that still allows for a saddle-point solution of the game. Expressions (3), (7), and (12) show that  $\gamma$  must be large enough to meet the conditions for the existence of a solution. Disturbance attenuation uses the minimum for  $\gamma$  that will satisfy these conditions. The Matlab program developed in this paper allows the user the choice to input positive values of  $r = \gamma^2$ , and will test to see whether the saddle-point conditions are met for the user's chosen value of  $r = \gamma^2$ . Thus, the user can choose any value of  $r = \gamma^2$  that is large enough to satisfy these conditions, and can then choose a value that is arbitrarily close to the infimum of these values for disturbance attenuation.

Note that if the conditions in (3), (7), and (12) given above are satisfied, then the solution to the soft-constrained game defined in Equations (1) and (2) is unique and global since the game is strictly convex in  $u$  and strictly concave in  $w$ , which is proven in [1]. The computer program developed here gives the user an error message if the theorem's conditions are not satisfied. Thus, the model will not allow the user to generate simulations in situations where the optimal (global) saddle-point solution does not exist in the soft-constrained LQ game. The value of  $r = \gamma^2$  must be large enough to admit a solution (if a solution exists) that will satisfy the global optimum saddle-point condition. There is no claim that this method extends to problems that are not strictly convex-concave, which cannot occur under the restricted LQ specification used in this model, given that the saddle-point solution conditions are satisfied. The global unique solution will be obtained in all cases if there is a value of the disturbance attenuation parameter that allows for a solution to the saddle-point condition matrices.

### 3. Application Example 1: Financial Assets

Consider a modified form of the control problem adapted

from [14,15]. Suppose that an investor deposits  $u_k$  dollars at the beginning of year  $k$  into a bank that pays an annually compounded interest rate of  $i_k$ . Let  $x_{1,k}$  be the account balance at the ending instant of year  $k-1$ , which is the beginning instant of year  $k$  before any new deposits or withdrawals are made. The state Equation is

$$x_{1,k+1} = (1+i_k)x_{1,k} + (1+i_k)u_k \quad (16)$$

Let  $i_k = 7\%$ , and assume that the initial balance in the account is zero, so that  $x_{1,1} = 0$ . Suppose that the investor's goal is to achieve a balance of \$100,000 at the end of the 10<sup>th</sup> year, *i.e.*, the beginning of the 11<sup>th</sup> year, so that  $K+1 = 11$ , and  $x_{1,11} = \$100,000$ .

One strategy would be to deposit an initial amount of  $u_1 = \$50834.93$ , and then let this grow to a balance of \$100,000 at the end of the tenth year, as shown in **Table 1**. In this case, the control variable deposit amounts are  $u_2 = u_3 = \dots = u_{10} = \$0$ . Unfortunately, many investors would not possess the large initial deposit that would be required.

As an alternative, the investor could follow an optimal control strategy where the performance index is given by

$$\min_u J = \frac{1}{2} \sum_{k=1}^K R_k u_k^2; \quad x_{1,K+1} \text{ is fixed}; \quad (17)$$

In this fixed terminal state case, the investor seeks to minimize the loss function in (17) subject to the state Equation

$$x_{1,k+1} = ax_{1,k} + bu_k \quad (18)$$

Using Equation (16), the parameters in Equation (18)

**Table 1. Deterministic control with fixed final state.**

Lump-Sum Deposit		Optimal Control Strategy			
$k$	$u$	$x_{1,k}$	$k$	$u$	$x_{1,k}$
1	\$ 50834.93	\$ -	1	\$ 8675.70	\$ -
2	\$ -	\$ 54393.38	2	\$ 8108.13	\$ 9283.00
3	\$ -	\$ 58200.91	3	\$ 7577.69	\$ 18608.51
4	\$ -	\$ 62274.98	4	\$ 7081.95	\$ 28019.23
5	\$ -	\$ 66634.22	5	\$ 6618.65	\$ 37558.27
6	\$ -	\$ 71298.62	6	\$ 6185.65	\$ 47269.30
7	\$ -	\$ 76289.52	7	\$ 5780.98	\$ 57196.80
8	\$ -	\$ 81629.79	8	\$ 5402.79	\$ 67386.23
9	\$ -	\$ 87343.87	9	\$ 5049.34	\$ 77884.25
10	\$ -	\$ 93457.95	10	\$ 4719.01	\$ 88738.94
11	\$ -	\$ 100000.00	11	\$ -	\$ 100,000.00

a.  $a = 1.07, b = 1.07, D = 1, K = 10, R = 1, Q_k = 0, x_1 = 0, x_{K+1} = \$100000.00$ .

are  $a = b = (1 + i)$ . Let the interest rate be fixed, and let the control parameter weight be fixed so that  $R_k = R$  for all  $k = 1, 2, \dots, K$ . The optimal control rule can be written as

$$u_k^* = \frac{(1 - a^2)}{b(1 - a^{2K})} (x_{1,K+1} - a^K x_0) a^{K-k} \quad (19)$$

A more flexible situation would allow the investor to have a free terminal state. Suppose that the investor is interested in optimally tracking a final account balance of \$100,000, but that this final value in the account at the end of the tenth year, *i.e.*, the beginning instant of the 11<sup>th</sup> year, is variable depending on an optimal performance index. Rather than using the index in Equation (17), the investor's objective is now to minimize the expression in (20) subject to Equation (18).

$$\min_u J = \frac{1}{2} Q_f (x_{1,K+1} - x_{1,K+1}^*)^2 + \frac{1}{2} \sum_{k=1}^K R_k u_k^2, \quad (20)$$

where  $x_{1,K+1}$  is free;  $x_{1,K+1}^* = \$100,000$ . The solution to the optimal LQ tracking problem is given by

$$u_k = -K_k x_{1,k} + g_k v_k \quad (21)$$

where the recursive Equations are

$$\begin{aligned} S_{K+1} &= Q_f; v_{K+1} = Q_f x_{1,K+1}^*; \\ h_k &= R_k + b^2 S_{k+1}; K_k = \frac{abS_{k+1}}{h_k}; \\ c_k &= a - bK_k; S_k = ac_k S_{k+1}; \\ v_k &= c_k v_{k+1}; g_k = \frac{b}{h_k}; \end{aligned} \quad (22)$$

**Table 2** shows the simulations of the tracking problem for different parameter values of  $Q_f$ , where the control weight is constant at  $R_k = 1$  for all  $k$ .

Next, consider an altered form of this problem. Suppose that an investor begins to purchase a stock, and wishes to grow the value of the portfolio to \$100,000 by the ending instant of the 10<sup>th</sup> year, *i.e.* the beginning instant of year 11. Unlike the bank deposit, the value of the stock held in the account could increase or decrease. One option would be to model the problem as a free final state, linear-quadratic Gaussian (LQG), where the stock value fluctuations are random errors. This would provide an optimal feedback control strategy  $u_k^*$  for the stock purchase amount in each period. With perfect state information, the feedback control rule would parallel the above tracking problem, where the actual value of the account balance  $x_{1,k}$  would be used in each period, rather than the deterministic, non-stochastic values in the above simulation.

Another way of obtaining insight is to consider a worst-case design approach, and formulate the problem

**Table 2. Deterministic control with free final state.**

Optimal Tracking Strategy: $Q_f = 0.50$			Optimal Tracking Strategy: $Q_f = 5.0$		
$k$	$u$	$x_{1,k}$	$k$	$u$	$x_{1,k}$
1	\$ 7972.48	\$ -	1	\$ 8599.84	\$ -
2	\$ 7450.92	\$ 8530.55	2	\$ 8037.24	\$ 9201.83
3	\$ 6963.47	\$ 17100.17	3	\$ 7511.44	\$ 18445.80
4	\$ 6507.92	\$ 25748.10	4	\$ 7020.03	\$ 27774.25
5	\$ 6082.17	\$ 34513.94	5	\$ 6560.78	\$ 37229.88
6	\$ 5684.27	\$ 43437.84	6	\$ 6131.57	\$ 46856.01
7	\$ 5312.40	\$ 52560.65	7	\$ 5730.44	\$ 56696.70
8	\$ 4964.86	\$ 61924.17	8	\$ 5355.55	\$ 66797.04
9	\$ 4640.06	\$ 71571.26	9	\$ 5005.19	\$ 77203.27
10	\$ 4336.50	\$ 81546.11	10	\$ 4677.74	\$ 87963.05
11		\$ 91894.39	11		\$ 99125.66

a.  $a = 1.07, b = 1.07, D = 1, K = 10, R = 1, Q_k = 0, x_1 = 0, x_{K+1}^* = \$100000.00$ .

through the  $H^\infty$ -optimal control methods discussed above. Let  $w_k$  be a disturbance term that represents the change in the value of the stock held in the account during period  $k$ . Let  $x_{3,k}$  denote the target amount in the account at time instant  $k$  that investor is tracking, and let  $x_{3,k}$  be the difference between the actual and the desired value of the stock held in the account at time instant  $k$ . This is also a LQ tracking problem, and can be converted into a LQ regulator-type of  $H^\infty$ -optimal control problem by writing Equations (1) and (2) as follows:

$$\begin{aligned} \min_u \max_w J_\gamma(u, w) &= x_{K+1}^T Q_f x_{K+1} \\ &+ \sum_{k=1}^K [x_k^T Q_k x_k + u_k^2 - r w_k^2] \end{aligned} \quad (23)$$

where, for  $k = 1, 2, \dots, K$ ,

$$\begin{aligned} r = \gamma^2 > 0; Q_f &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_{3,f} \end{bmatrix}, \\ Q_k &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_{3,k} \end{bmatrix}, q_{3,f} > 0, q_{3,k} > 0 \end{aligned}$$

subject to

$$x_{k+1} = A_k x_k + B_k u_k + D_k w_k \quad (24)$$

where

$$x_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \end{bmatrix} A_k = \begin{bmatrix} 1+i_k & 0 & 0 \\ 0 & 1 & 0 \\ 1+i_k & -1 & 0 \end{bmatrix}$$

$$B_k = \begin{bmatrix} 1+i_k \\ 0 \\ 1+i_k \end{bmatrix} D_k = \begin{bmatrix} d_k \\ 0 \\ 0 \end{bmatrix}$$

Assume that the investor starts with a zero balance and is only concerned with getting near the final target amount of \$100,000 at the final time at the end of 10 years, which is the beginning of year 11. Also assume that there is no intermediate state weighting, so that  $q_{3,k} = 0$  for all  $k=1, \dots, K$ . As in the problem scenario above, assume that the long-run average rate of return is constant at  $i_k = 7\%$ , and that the disturbance terms represent the short-term loss in the stock value that occurs over period  $k$ , which causes the portfolio value to decline. The program allows the user to set the values for all parameters, and for the initial values in the state vector. For this example, let  $d_k = 1$  for all  $k=1, \dots, K$ , so that the disturbance directly represents the loss in the value of the account due to the fall in the stock's value over the period. In this case, the coefficient matrices are constant:

$$x_1 = \begin{bmatrix} 0 \\ 100,000 \\ -100,000 \end{bmatrix} A = \begin{bmatrix} 1.07 & 0 & 0 \\ 0 & 1 & 0 \\ 1.07 & -1 & 0 \end{bmatrix}$$

$$B_k = \begin{bmatrix} 1.07 \\ 0 \\ 1.07 \end{bmatrix} D_k = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

The investor is seeking to drive the shortfall in the stock account from an initial value of  $x_{3,1} = -\$100,000$  to  $x_{3,11} = \$0$  at the end of the time 10-year time horizon. The more important is the goal of having a portfolio value of \$100,000 at the end of 10 years, which means a reaching an account balance deficiency of  $x_{3,11} = \$0$ , the higher will be the weight expressed in the parameter  $q_{3,f}$ . The optimally simulated values using a closed-loop information scheme are given in **Table 3**.

Reference [16] uses the golden ratio as a search algorithm to compute the optimal bound for slow and fast subsystems in an  $H^\infty$ -suboptimal perturbation model. Reference [1] also calculates a golden ratio disturbance optimal attenuation value in a scalar model with imperfect information model where all of the performance index weights are unity, and all variables in the state and measurement Equations have unit coefficients. In order to be consistent with these previous studies, the simulations in **Table 3** follow this choice of relative weighting design, and thus use  $q_{3,f} = 1.618$ , the golden ratio, as the ratio between the parameters on the final state and control

**Table 3. Minimax control with higher risk exposure.**

$k$	$u$	$w$	$x_{1k}$	$x_{2k}$	$x_{3k}$
1	\$9062.50	-\$4234.80	\$0.00	\$100000.00	-\$100000.00
2	\$9092.90	-\$4249.00	\$5460.00	\$100000.00	-\$90300.00
3	\$9185.20	-\$4292.10	\$11320.00	\$100000.00	-\$84430.00
4	\$9356.10	-\$4372.00	\$17650.00	\$100000.00	-\$78050.00
5	\$9632.90	-\$4501.40	\$24530.00	\$100000.00	-\$71100.00
6	\$10062.00	-\$4701.70	\$32050.00	\$100000.00	-\$63450.00
7	\$10728.00	-\$5013.30	\$40360.00	\$100000.00	-\$54940.00
8	\$11816.00	-\$5251.70	\$49650.00	\$100000.00	-\$45340.00
9	\$13813.00	-\$6454.70	\$60250.00	\$100000.00	-\$34230.00
10	\$18736.00	\$0.00	\$72790.00	\$100000.00	-\$20750.00
11			\$97930.00	\$100000.00	-\$2070.00

a.  $q_{3,f} = 1.618, d = 1, r = 2$ .

variables in the performance index. Given these parameters, the disturbance attenuation value of  $r = 2$  is the lowest whole number value for  $r$  that admits a minimax saddle-point solution.

The results in the previous example that were shown in **Table 3** can be compared to an alternative case. Suppose that the investor decides to purchase shares in mutual fund that contained both stocks and bonds, and that the mutual fund has a much lower variance, or volatility, around its expected mean growth rate. This less-risky alternative provides an intermediate case between the examples in **Tables 2** and **3**. Assume that the rate of return is still constant at constant at  $i_k = 7\%$ , but that  $d_k = 0.5$  for all  $k=1, \dots, K$ , so that the potential loss in the value due to the disturbance is decreased. Further, assume that the final state weight in the performance index is increased from  $q_{3,f} = 1.618$  to  $q_{3,f} = 3.0$ .

In **Table 4**, the annual contributions in the  $u_k$  again have the decreasing pattern over time that was obtained in the deterministic case of **Table 2**. However, the minimax control investment strategy in **Table 4** is still more aggressive and requires higher contributions than in the deterministic case. When comparing the optimal two minimax designs in **Tables 3** and **4**, note that the optimal control annual contributions are much higher in **Table 3** when the risk exposure is greater. And, when the penalty weight on the final deviation of the state variable from its target is increased to  $q_{3,f} = 3.0$ , the final account balance is closer to the target amount in **Table 4** than it was in **Table 3**.

Note that the authors have run many sensitivity simulations with various values for all of the parameters, and the performance index weights above can be chosen

**Table 4. Minimax control with lower risk exposure.**

$k$	$u$	$w$	$x_{1k}$	$x_{2k}$	$x_{3k}$
1	\$8699.90	-\$2032.70	\$0.00	\$100000.00	-\$100000.00
2	\$8275.30	-\$1933.50	\$8290.00	\$100000.00	-\$90690.00
3	\$7885.70	-\$1842.50	\$16760.00	\$100000.00	-\$82270.00
4	\$7531.40	-\$1759.70	\$25450.00	\$100000.00	-\$73630.00
5	\$7213.70	-\$1685.50	\$34410.00	\$100000.00	-\$64710.00
6	\$6936.10	-\$1620.60	\$43700.00	\$100000.00	-\$55460.00
7	\$6706.10	-\$1566.80	\$53370.00	\$100000.00	-\$45820.00
8	\$6540.40	-\$1528.10	\$63490.00	\$100000.00	-\$35720.00
9	\$6482.50	-\$1514.60	\$74170.00	\$100000.00	-\$25060.00
10	\$6696.40	\$0.00	\$85540.00	\$100000.00	-\$13700.00
11			\$98700.00	\$100000.00	-\$1300.00

a.  $q_{3f} = 3, d = 0.5, r = 2.$

arbitrarily. The Matlab program is included in the Appendix, and users can run their own simulations. Portions of the program can be modified to handle other similar applications in economics and finance, and the authors have done this to handle other types of LQ tracking problems with higher dimensional matrices. Due to brevity and space considerations, only the simulations in **Tables 3 and 4** are reported here. Comparing these results to those in **Table 2** shows that the investor will optimally follow a strategy that requires a higher dollar input in each period in order to counteract the short-term disturbances when the stock or bundled portfolio underperforms its trend growth.

The minimax strategy can be employed in different ways. The investor could follow the simulated  $H^\infty$ -optimal control strategy when purchasing stock in each period  $k$ . Alternatively, the investor could treat the disturbance value losses as the shortfalls that need to be covered through additional deposits. These shortfalls could be added to the control inputs, *i.e.*, the investment amounts in each respective period  $k$ , within the previous deterministic LQ-tracking problem. The investor could also design a control rule that used a weighted average of the controls, or disturbances, from standard LQ-tracker and the  $H^\infty$ -optimal control analysis in order compromise between the two approaches. All of these options allow the user, or investor, to evaluate and incorporate robustness considerations when formulating the feedback control investment, and to develop a desired hedge strategy.

#### 4. Application Example 2: Input Purchasing

Consider the purchasing agent for a retail electricity supplier in a deregulated market. Wholesale electricity gen-

eration markets are generally characterized by fixed-price forward contracts, where the delivery price of some fixed quantity of electricity is determined during some period in advance of the hour of delivery, or transmission. Wholesale electricity sellers have an incentive to sign contracts in order to ensure demand, gain advantage over competitors, and deter market entry. Purchasers will sign contracts in order to reduce uncertainty over prices and quantities. Note that this analysis actually applies to any industry where inputs are commonly purchased through advanced forward contracts. This also includes electricity generator firms who are purchasing natural gas and coal inputs in the forward market to hedge against unexpected increases in input prices.

Assume that the demand for electricity for the retail utility firm has been growing at a constant average rate of  $i$  per period, so that  $A = 1 + i$ . Also assume that the retail plant manager would like to sign forward contracts to satisfy the firm's entire predicted demand. This can be modeled by a scalar version of Equations (1) and (2). Let  $u_k$  be the control variable that denotes net change in the forward contract position from the previous period. In other words, it is the quantity of electricity (measured in megawatt hours) that the plant manager purchases in the forward market that is in addition to the previous forward purchase amount in period  $k$  under a fixed-price forward contract to be delivered in period  $k + 1$ . Let  $x_k$  be the difference between the retail demand for electricity (measured in megawatt hours) during period  $k$ , and the total amount of fix-price forward contracts that are already in place in period  $k$ . Thus, the plant manager is seeking to steer the final value of the state variable to  $x_{K+1} = 0$ . Since each quantity purchased forward reduces the residual excess demand,  $B$  will be negative. For the simulations, let  $B = -1$ , and let  $D = 1$  since the disturbance term  $w_k$  represents unanticipated excess demand.

This problem lends itself to robust considerations, since the input purchaser will face exposure to potentially high market prices during the period of high demand. In many cases, these price fluctuations will not be characterized by a stochastic error term with a constant distribution. Thus, the information gained from minimax simulation analysis would prove helpful.

**Tables 5 and 6** show simulations for different parameters, where  $K = 12$ . **Table 5** uses a demand growth rate of  $i = 0.03$ , and **Table 5** uses a growth rate of  $i = 0.02$ . These tables use the minimum whole number disturbance attenuation values for  $\gamma$  that allow the saddle-point solution existence condition to be satisfied. Note that the authors have run many simulations, and are only including the following examples as illustrations. The Matlab program for this scalar system is available from the authors upon request, and can be applied toward

**Table 5. Scalar simulation with smaller state weights.**

$k$	$\lambda$	$M$	$u$	$w$	$x_k$	$S$
1	1.5867	0.6037	78.3908	2.2397	200.0000	10.3983
2	1.5867	0.6037	50.8942	1.4541	129.8489	7.5416
3	1.5866	0.6037	33.0420	0.9441	84.3044	5.7778
4	1.5866	0.6037	21.4514	0.6129	54.7355	4.5709
5	1.5864	0.6036	13.9258	0.3979	35.5391	3.6861
6	1.5861	0.6034	9.0392	0.2583	23.0773	3.0040
7	1.5852	0.6031	5.8654	0.1676	14.9887	2.4574
8	1.5831	0.6022	3.8032	0.1087	9.7405	2.0059
9	1.5783	0.6001	2.4618	0.0703	6.3382	1.6233
10	1.5671	0.5951	1.5868	0.0453	4.1369	1.2921
11	1.5414	0.5835	1.0125	0.0289	2.7196	1.0000
12	1.4861	0.5570	0.6301	0.0180	1.8176	0.7381
13		0.5000			1.2601	0.5000

a.  $A = 1.03, B = -1, D = 1, K = 12, R = 1, Q_k = 0.2, Q_f = 0.5, x_1 = 200, r = 36, \gamma = 6.$

**Table 6. Scalar simulation with larger state weights.**

$k$	$\lambda$	$M$	$u$	$w$	$x_k$	$S$
1	2.6357	1.6522	127.8788	1.2788	200.0000	61.1089
2	2.6357	1.6522	49.4891	0.4949	77.4000	36.6185
3	2.6357	1.6522	19.1523	0.1915	29.9538	25.5040
4	2.6357	1.6522	7.4119	0.0741	11.5921	19.0627
5	2.6357	1.6522	2.8684	0.0287	4.4861	14.7931
6	2.6356	1.6522	1.1101	0.0111	1.7361	11.7056
7	2.6356	1.6522	0.4296	0.0043	0.6719	9.3299
8	2.6352	1.6521	0.1662	0.0017	0.2600	7.4129
9	2.6326	1.6517	0.0643	0.0006	0.1006	5.8060
10	2.6155	1.6491	0.0248	0.0002	0.0390	4.4154
11	2.5076	1.6318	0.0094	0.0001	0.0152	3.1784
12	1.9900	1.5228	0.0032	0.0000	0.0062	2.0509
13		1.0000			0.0032	1.0000

a.  $A = 1.02, B = -1, D = 1, K = 12, R = 1, Q_k = 1, Q_f = 1, x_1 = 200, r = 100, \gamma = 10.$

finding the  $H^\infty$ -optimal solution for any scalar analysis that uses expressions (1) and (2).

When the state variable parameter weights  $Q_f$  and  $Q_k$  are larger, the purchaser will arrange more fix-price forward contracts during the earlier periods, and the worst-case disturbance term is driven down faster. As shown in

**Table 6**, the purchaser will have purchased almost all of the required input in the forward market by the end of the planning horizon. As in the previous financial application example, this type of analysis can be used to plan a more robust optimal managerial strategy.

### 5. Conclusions

This analysis has developed a method for converting the discrete time LQ-tracking problem into an LQ-regulator design, which can be simulated with a Matlab software application. The framework can be utilized to provide insight into a variety of finance and economics issues. The limited application of  $H^\infty$ -optimal control methods in economic analysis has been partly due to the recognition that many economic problems can be adequately addressed through robust control. This problem has been exaggerated by a failure to formulate alternative robust versions of ubiquitous economic problems where minimax control would have offered useful insight. The limited application of  $H^\infty$ -optimal control methods has also been partly due to the lack of efficient computational methods that can be readily employed to handle these problems. The above methodology has addressed all three of these  $H^\infty$ -optimal control robust application issues in financial economics: problem formulation, solution computation, and simulation.

The minimax control methods explored in this paper have been employed in order to design a clear objective to guide the policies in terms of both thrust and magnitude. The applications given above are straightforward, and represent only a small range of applications that could be addressed with this framework. They demonstrate how to formulate the applications as minimax problems, and then how to use software to simulate solutions that have good performance. The minimax LQ-tracking conversion method and the associated Matlab program presented here provide a user-friendly computational method that efficiently simulates the minimax design, and can be used in applications in finance, economics, or other disciplines. Further analysis would apply the method and the examples above to cases of imperfect state information. The authors have explored some of these comparisons, but this is not discussed here since it is beyond the scope of the paper.

### REFERENCES

- [1] T. Basar and P. Bernhard, "H<sup>∞</sup>-Optimal Control and Related Minimax Design Problems," Birkhauser, Boston, 1991.
- [2] G. Chow, "Analysis and Control of Dynamic Economic Systems," John Wiley and Sons, New York, 1975.
- [3] A. Sage and C. White, "Optimum Systems Control," 2nd Edition, E Prentice Hall, Englewood Cliffs, 1977.

- [4] D. Kendrick, "Stochastic Control for Econometric Models," McGraw Hill, New York, 1981.
- [5] A. Tornell, "Robust-H-Infinity Forecasting and Asset Pricing Anomalies," *NBER Working Paper No. 7753*, 2000.
- [6] T. Basar, "On the Application of Differential Game Theory in Robust Controller Design for Economic Systems," In: G. Feichtinger, Ed., *Dynamic Economic Models and Optimal Control*, North-Holland, Amsterdam, 1992, pp. 269-278.
- [7] L. Hansen, T. Sargent and T. Tallarini, "Robust Permanent Income and Pricing," *Review of Economic Studies*, Vol. 66, No. 4, 1999, pp. 873-907.  
[doi:10.1111/1467-937X.00112](https://doi.org/10.1111/1467-937X.00112)
- [8] D. Hudgins and C. Chan, "Optimal Exchange Rate Policy under Unknown Pass-Through and Learning with Applications to Korea," *Computational Economics*, Vol. 32, No. 3, 2008, pp. 279-293.  
[doi:10.1007/s10614-008-9139-1](https://doi.org/10.1007/s10614-008-9139-1)
- [9] L. Hansen and T. Sargent, "Robustness," Princeton University Press, New York, 2008.
- [10] P. Bernhard, "Survey of Linear Quadratic Robust Control," *Macroeconomic Dynamics*, Vol. 6, No. 1, 2002, pp. 19-39. [doi:10.1017/S1365100502027037](https://doi.org/10.1017/S1365100502027037)
- [11] R. Dennis, K. Leitemo and U. Soderstrom, "Methods for Robust Control," *Journal of Economic Dynamics and Control*, Vol. 33, No. 8, 2009, pp. 1004-1616.  
[doi:10.1016/j.jedc.2009.02.011](https://doi.org/10.1016/j.jedc.2009.02.011)
- [12] G. Barlevy, "Robustness and Macroeconomic Policy," *Annual Review of Economics, Federal Reserve Bank of Chicago*, Vol. 3, No. 1, 2011, pp. 1-24.
- [13] T. Basar and G. Olsder, "Dynamic Noncooperative Game Theory," Academic Press, London, 1982.
- [14] D. Luenberger, "Introduction to Dynamic Systems," John Wiley & Sons, New York, 1979.
- [15] F. Lewis, "Optimal Control," John Wiley & Sons, New York, 1986.
- [16] Z. Pan and T. Basar, "Designing  $H^\infty$ -Optimal Controller for Singularly Perturbed Systems; Part I: Perfect State Measurements," *Automatica*, Vol. 29, No. 2, 1991, pp. 401-423. [doi:10.1016/0005-1098\(93\)90132-D](https://doi.org/10.1016/0005-1098(93)90132-D)

## Appendix: Matlab Computer Program

The Matlab computer problem for the matrix system problem for constant coefficients is given below. The user can set some of the system parameters, as described in the previous analysis, when prompted at the beginning of each simulation. The user can change the fixed matrix parameters within the code if desired in order use the program to solve other application problems.

```

clear all
clc
format short
run = 1;

%user gets a choice between open loop information structure or closed-loop information structure.
while (run == 1)
fprintf('1: Open Loop Information 2: Closed Loop Information \n');
method = input('Choice: ');
if method > 2
error('Invalid Input')
elseif method < 0
error('Invalid Input')
end

switch method
case 1% open loop information structure control

    %User input of desired iteration and other variables
    K = input('Input the number of iteration: ');
    ik = input('Input the desired i,k value: ');
    dk = input('Input the desired d,k value: ');
    q3f = input('Input the desired q3,f value: ');
    q3k = input('Input the desired q3,k value: ');

    %setting up the A,B,D,Qf,andQk matrix
    A = [1+ik 0 0; 0 1 0; 1+ik -1 0];
    B = [1+ik; 0; 1+ik];
    D = [dk; 0; 0];
    Qf = [0 0 0; 0 0 0; 0 0 q3f];
    Qk = [0 0 0; 0 0 0; 0 0 q3k];

    %user input of the r value
    r = input('Input the desired value of r: ');
    if r < 0
error('Invalid input: the value of gamma cannot be negative')
end

    %solving the Riccati Equation to check if the r value produces a unique saddle-point solution
    S = zeros(3,3,K);
    S(:, :, K+1) = Qf;

    for i = (K:-1:1) %Equation (4) above; Riccati Equation for S
        S(:, :, i) = Qk + (A'*S(:, :, i+1)*A)+(A'*S(:, :, i+1)*D)*(((r*eye(1))-D'*S(:, :, i+1)*D)^-1)*(D'*S(:, :, i+1)*A);
    end

    fprintf('S values: \n')

```

```

disp(S)

    %checking the condition given by Equation (3) above
check = zeros(K,1);
fprintf('Conditional Values: \n')
for j = 1:1:K+1
check(j) = (r*eye(1)) - (D'*S(:,j)*D);
if check(j) < 0
warning('Your r value does not satisfy the required condition')
fprintf('Press any key to continue')
pause
end
end
disp(check)

    %Creating matrices for lambda,M,u,w,and x (state trajectory)
lambda = zeros(3,3,K);
M = zeros(3,3,K);
M(:,K+1) = Qf;
u = zeros(3,1,K);
w = zeros(3,1,K);
x = zeros(3,1,K);
x_asterisk = zeros(3,1,K);

for n=(K:-1:1)
    %Equations (5) and (6) above.
lambda(:,n) = eye(size(A)) + (B*B' - (1/r)*(D'*D))*M(:,n+1);
M(:,n) = Qk + (A'*M(:,n+1)*(lambda(:,n)^-1)*A);
end

fprintf('lambda values are: \n')
disp(lambda)
fprintf('M values are: \n')
disp(M)

    %The initial value of x is given by the user
x(:,1) = input('Input the initial value of x: ');
    %[0; 100000; -100000];
x_asterisk(:,1) = x(:,1);
    %condition given by Equation 3.5c

    %producing u,w,x, and x* values as described by above, and by Basar and Bernhard (1991)
for k = 1:1:K %Equations (8), (9), and (10) above; equivalent to Equations 3.5a,3.5b,3.5c; 3.1 for x(k+1)=ax+bu+dw
u(:,k) = -B'*M(:,k+1)*(lambda(:,k)^-1)*A*x_asterisk(:,k);
w(:,k) = (1/r)*D'*M(:,k+1)*(lambda(:,k)^-1)*A*x_asterisk(:,k);
x(:,k+1) = A*x(:,k)+ B.*u(:,k)+ D.*w(:,k);
x_asterisk(:,k+1) = (lambda(:,k)^-1)*A*x_asterisk(:,k);
end

fprintf('u values are: \n')
disp(u)
fprintf('w values are: \n')
disp(w)
fprintf('State trajectories (x* values) are: \n')

```

```

disp(x_asterisk)

case 2 %closed loop perfect state information structure

    %Desired number of iteration
    K = input('Input the number of iteration: ');
if K < 1
error('Invalid input')
elseif K == 1
    fprintf('For a better result, choose higher number of K')
end

    %Parameters given by the user
ik = input('Input the desired i,k value: ');
dk = input('Input the desired d,k value: ');
q3f = input('Input the desired q3,f value: ');
q3k = input('Input the desired q3,k value: ');

    A = [1+ik 0 0;0 1 0;1+ik -1 0];
    B = [1+ik;0;1+ik];
    D = [dk;0;0];
Qf = [0 0 0;0 0 0; 0 0 q3f];
Qk = [0 0 0; 0 0 0; 0 0 q3k];

    %User desired r
    r = input('Input the desired value of r: ');
if r < 0
error('Invalid input: gamma cannot be negative')
end

lambda = zeros(3,3,K);
M = zeros(3,3,K);
M(:, :,K+1) = Qf;
u = zeros(3,1,K);
w = zeros(3,1,K);
x = zeros(3,1,K);

for n =(K:-1:1) %Equations 3.4a,3.4b
lambda(:, :,n) = eye(size(A)) + (B*B' - (1/r)*(D'*D))*M(:, :,n+1);
M(:, :,n) = Qk + (A'*M(:, :,n+1)*(lambda(:, :,n)^-1)*A);
end

fprintf('lambda values are: \n')
disp(lambda)
fprintf('M values are: \n')
disp(M)

%checking the condition given by Equation (12) above.
xi = zeros(K,1);
fprintf('Conditional Values: \n')
for j = 1:1:K
xi(j) = (r*eye(1)) - (D'*M(:, :,j+1)*D);
if xi(j) < 0
warning('Your r value does not admit an unique saddle-point solution')

```

```

fprintf('Press any key to continue')
pause
end
end
disp(xi)

%The initial value of x is given by the user
x(:,1) = input('Input the initial value of x: '); %[-100000; 100000];

%producing x values
for k = 1:1:K
    %Equations (13), (14), (15) above, for x(k+1)=ax+bu+dw
    u(:,k) = -B'*M(:,k+1)*(lambda(:,k)^-1)*A*x(:,k);
    w(:,k) = (1/r)*D'*M(:,k+1)*(lambda(:,k)^-1)*A*x(:,k);
    x(:,k+1) = A*x(:,k)+ B.*u(:,k)+ D.*w(:,k);
end

fprintf('u values are: \n')
disp(u)
fprintf('w values are: \n')
disp(w)
fprintf('State trajectories (x values) are: \n')
disp(x)

end

%option to exit the program
fprintf('1 : for more computation 2: exit\n')
run = input('Choice: ');
end

```