A MODEL OF BOUNDEDLY RATIONAL LEARNING IN DYNAMIC GAMES

A THESIS PRESENTED BY HAKAN AKGOY
TO
THE INSTITUTE OF
ECONOMICS AND SOCIAL SCIENCES
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS
FOR THE DEGREE OF MASTER OF ARTS
IN ECONOMICS

BILKENT UNIVERSITY
AUGUST, 1997

A MODEL OF BOUNDEDLY RATIONAL LEARNING IN DYNAMIC GAMES

A THESIS PRESENTED BY HAKAN AKSOY

TO

THE INSTITUTE OF

ECONOMICS AND SOCIAL SCIENCES

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS

FOR THE DEGREE OF MASTER OF ARTS

IN ECONOMICS

Hakan Aksoy

BILKENT UNIVERSITY

AUGUST, 1997

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Economics.

Assist. Prof. Dr. Erdem Başçı

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Economics.

Assoc. Prof. Dr. Farhad Hüseyinov

I certify that I have read this thesis and in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Economics.

Dr. Nazmi Demir

Approved by the Institute of Economics and Social Sciences

Director:

ABSTRACT

A MODEL OF BOUNDEDLY RATIONAL LEARNING IN DYNAMIC GAMES

HAKAN AKSOY

MASTER OF ARTS IN ECONOMICS

Supervisor: Assist. Prof. Dr. Erdem Başçı

August, 1997

There are various computer-based algorithms about boundedly rational players' learning how to behave in dynamic games, including classifier systems, genetic algorithms and neural networks. Some examples of studies using boundedly rational players are Axelrod (1987), Miller (1989), Andreoni and Miller (1990) who use genetic algorithm and Marimon etal. (1990) and Arthur (1990) who use classifier systems. In this dissertation, a Two Armed Bandit Problem and the Kiyotaki-Wright (1989) Economic Environment are constructed and the learning behaviour of the boundedly rational players is observed by using classifier systems in computer programs. From the simulation results, we observe that experimentation and imitation enables faster convergence to the correct decision rules of players in both repeated static decision problems and dynamic games.

Key Words: Dynamic Games, Bounded Rationality, Classifier Systems, Two Armed Bandit Problem, Kiyotaki and Wright Model of Money, Learning, Experimentation, Imitation.

ÖZET

DİNAMİK OYUNLARDA SINIRLI RASYONEL BİR ÖĞRENME MODELİ

HAKAN AKSOY

Yüksek Lisans Tezi, Ekonomi Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Erdem Başçı

Ağustos, 1997

Dinamik oyunlarda sınırlı rasyonel oyuncuların öğrenirken nasıl davranacağı hakkında bilgisayar destekli birçok algoritma vardır. Bunlardan bazıları sınıflandırma sistemleri, genetik algoritmalar ve nöral sistemlerdir.Örneğin Axelrod (1987), Miller (1989), Andreoni ve Miller (1990) 'in makalelerinde genetik algoritma, Marimon etal. (1990) ve Arthur (1990) 'un makalelerinde de sınıflandırma sistemleri kullanılmıştır. Bu tezde İki Kollu Haydut Problemi ve Kiyotaki ve Wright (1989) 'ın makalesindeki ekonomi modeli oluşturulmuş ve bilgisayar programları ile sınıflandırma sistemleri kullanılarak oyuncuların öğrenebilirliği incelenmiştir. Dinamik oyunlarda deney yapmanın ve toplumun tecrübelerinden yararlanmanın oyunculara doğru ve daha hızlı yakınsama yaptırdığı simülasyon sonuçlarından gözlenmiştir.

Anahtar Kelimeler: Dinamik Oyunlar, Sınırlı Rasyonellik, Sınıflandırma Sistemleri, İki Kollu Haydut Problemi, Kiyotaki ve Wright Para Modeli, Öğrenme, Deney Yapma, İmitasyon.

Acknowledgements

I would like to express my gratitude to Assist. Prof. Dr. Erdem Başçı for his valuable supervision, for encouraging me and for providing me with the necessary background. I also would like to thank Assoc. Prof. Dr. Farhad Hüseyinov and Dr. Nazmi Demir for their valuable comments and suggestions.

Contents

1 Introduction

This dissertation is about learning how to behave. The economic environments studied here are quite complex, while the learning models considered are quite simplistic. Nevertheless, in three different example applications of increasing complexity, we observe convergence of behavior to the theoretical equilibrium.

An important class of economic decision problems are quite complex. The first type of complexity arises in situations where payoffs corresponding to some of the actions are random and moreover the players do not know much about the payoff probability distribution functions. A second type of complexity arises in dynamic problems where the actions taken now affect future conditions faced by the players. Yet, a third type of complexity arises due to the effects of player's actions on other players' payoffs, i.e. the game aspect. When all these three types of complexities are present in a model, it is usually called a *stochastic dynamic game.* If the same decison situations are faced again and again over time, the problem is said to have a recursive structure and may be called a *recursive stochastic dynamic game.*

In such a complex environment it may be too much to expect from the individual players to explain what happens in the whole system and hence, their behavior in general may be suboptimal. In an *adaptive learning system*, however, the actions of each player are assigned a value, such as performance, utility or strength and each player behaves so as to choose the higher valued actions, while updating these values with the help of experience.

A wide range of computer-based adaptive algorithms exists for exploring such behavior, including classifier systems, genetic algorithms, neural networks, and similar reinforcement learning mechanisms. Some examples of studies using adaptive players are Axelrod (1987), Miller (1989), Andreoni and Miller (1990) who use genetic

algorithm and Marimon etal. (1990) and Arthur (1990) who use classifier systems.

In this thesis, we concentrate on classifier systems as a boundedly rational, adaptive learning device in three different economic environments. The classifier systems have been introduced in the artificial intelligence literature by Holland (1975) as a model of human brain that could enable machines to "decide" and "learn". A classifier system is a collection of rules together with their strengths. Each rule is a "condition-action" or an "if-then" statement. In a decision situation, the rules whose condition parts are satisfied are said to be activated. Out of all the activated rules, each may give contradictory advices on how to act. Therefore to be selected, they compete by bidding their strengths. The classifiers with higher bids are selected more often, however the selection criteria for winning clasifiers have varied in the literature (See Arthur, 1990 and Marimon etal, 1990 for two different selection criteria, also see Sargent, 1994).

In the next section we review the literature on money as a medium of exchange with emphasis on the Kiyotaki and Wright (1989) paper, which will be the underlying economic environment in section four.

In section 3, we consider the so called *two armed bandit problem* as a demonstration of the performance of classifier systems in repeated static decison situations. In a two armed bandid problem, there are two arms A and B which a player can pull. In our setup, arm A yields a deterministic utility payoff of 5 units, while arm B yields a utility payoff of 1000 units with probability 1 percent and 0 units with probability 99 percent. Clearly a rational and fully informed player would choose always arm B, since it provides higher expected utility. There is also a Bayesian literature on how a rational player, who does not know the payoff structure but needs to experiment according to his priors, should act. In the first part, we present the recursive

least squares learning technique and will use this approach in all of the computer programs. In the second part, we construct the algorithm for the two armed bandit problem by using boundedly rational players with classifier systems as a learning device. The contribution here is that we allow both imitation and experimentation and observe the speed of learning for different imitation probabilities.

In section 4, we consider the Kiyotaki and Wright (1989) model of money as a medium of exchange and make theoretical calculations of three classes of rational players' behaviours. Our calculations show that the good with lowest storage cost plays the role of medium of exchange under "fundamental equilibrium" parameters. Similarly, the lowest and highest storage cost goods play the role of medium of exchange under "speculative equilibrium" parameters. We first try to observe the performance of classifier systems in a simple dynamic optimization problem. In this model we consider a finite number of unexperienced players facing an infinite number of learned players. Again, experimentation and various degrees of imitation of the finite reference group are allowed for. Next we construct the Kiyotaki-Wright economic environment as a dynamic game with a finite number of players, all unexperienced in the begining. Finally, we present the algorithm of the game and study the presence of convergence to the so called "fundamental" and "speculative" equilibria again by allowing experimentation and imitation in classifier systems. In this section, also a 100 independent simulations are run with no imitation, half imitation and full imitation, to observe the convergence rates.

The last section summarizes and interprets the results in light of the previous results in the literature, a brief account of which is given below.

2 Models of Money as a Medium of Exchange

The Kiyotaki and Wright (1989) paper is an attempt to capture the transactions demand for money. Due to the nature of trading arrangements, when two players meet, there is a 'mutual inconsistency of needs' and hence at least one of them has to accept an item, which will not be consumed immediately, in a trade round. In this model, media of exchange out of the three storable commodities are determined endogenously as part of the noncooperative equilibrium. When a commodity is accepted in trade, not for consumption purposes, but for facilitating future trade, that commodity acts as money in the model.

The Kiyotaki and Wright (KW) environment introduces trading frictions by its market structure. There is no central clearing house, but players in each period are matched randomly with one other player from the population. If they mutually agree to trade, they swap their endowments, which is always one unit of one of the three commodities available. The three commodities have different storage costs. Under a certain range of parameters, the unique Nash-Markov equilibrium turns out to be the one in which only the the commodity with the lowest storage cost acts as a medium of exchange. This equilibrium is called "the fundamental equilibrium". For an other range of parameters, the unique equilibrium consists of both the lowest and highest storage cost commodities being accepted and circulated as media of exchange. This one is called "the speculative equilibrium".

Related previous models in the literature include the framework of Jones (1976) where many commodities circulate as media of exchange, the paper by Iwai (1988), where expectations are fully rational but players are able to choose only simple trading patterns, studies in sequential bargaining theory, such as Mortensen (1982), Rubinstein and Wolinsky (1985), and Gale (1986) which have a similar mathcing

structure to the KW paper.

Kehoe, Kiyotaki and Wright (1993) extend the KW model to allow mixed strategy equilibria and dynamic equilibria. They show that steady-state equilibria always exist in mixed-strategies and new mixed-strategy steady-state equilibria can arise when there is a unique pure strategy equilibrium.

Yiting Li (1995) analyzes the informational frictions on the choice of commodity money in an economy with specialized production and consumption.People do not recognize the commodity, when they do not produce or consume it. He also analyzes the effect of private information on the interaction between players' willingness. There exists an equilibrium in which the lowest storage cost commodity serves as the unique commodity money. A related study by Cuadras-Morato (1994) also introduces private information into the Kiyotaki-Wright model, however he considers different questions.

Marimon, McGrattan and Sargent (1990) use artificially intelligent players to examine the equilibria that arise in Kiyotaki-Wright economic environment. Their players use a common classifier system to make their decisions. In their paper they try two different classifier systems. A complete enumaration of all possible rules is carried along in the first kind of classifier system. In the second kind of classifier system, a modified version of the genetic algorithm of Holland is used. In this case, some rules are eliminated and new rules are injected into the population of rules. They do not allow for experimentation. As a result, in the case of "fundamental equilibrium" they observe convergence to Nash strategies, while in the case of "speculative equilibrium" convergence is not attained.

Lettau and Uhlig (1993) study the relationships between classifier systems and dynamic programming. For solving stochastic dynamic optimazation problems, there

are similarities and differences between classifier systems and dynamic programming. In some situations a classifier system may not converge to the optimal solution, although it is reachable and may not settle down on a strict ranking of its classifiers. These results are robust and hint the differences between dynamic programing and classifier systems. However, their setting does not allow for experimentation or any kind of randomness in selecting among the activated classifiers.

Brown (1995) is an interesting experimental examination of Kiyotaki and Wright's model (1989) of how money can arise as a medium of exchange. His laboratory results with economics students indicate that individuals tend to utilize one of the goods as a medium of exchange however, there are some deviations from the predicted patterns, even after 50 rounds of trade. He does not allow for communication and experience sharing between his subjects.

In our simulations a high proportion (usually 100 percent) of each type of players converge to a stationary equilibrium even if players start with random rules. When we get a stationary equilibrium in Kiyotaki-Wright environment, our simulations show that each good with low storage costs play the role of medium of exchange under "fundamental equilibrium" parameters. Similarly, the lowest and highest storage cost play the role of medium of exchange under "speculative equilibrium" parameters, which is consistent with the theoretical equilibrium. However, the speed of convergence in the latter case is rather low.

Our simulations, in general, show that in order to reach the theoretical equilibria, experimentation, where each individual player makes a mistake, is neccessary. Moreover, imitation, where each individual player considers the value of accumulated social experience, speeds up the convergence process.

3 A Two Armed Bandit Problem

The simplest repeated static decision problem with unkown random payoffs is perhaps the two armed bandit problem. In our two armed bandit problem, players will have two choices, choosing action A or choosing action B. When players choose action A, they get utility 5. Choosing action B yields either a utility of 1000 with 1 percent probability or a utility of 0 with 99 percent probability. Clearly action B provides a higher expected utility. In this section we study the performance of classifier systems as a boundedly rational learning device. The insights developed in this section will help in understanding the performance of classifier systems under more complex environments.

3.1 Recursive Least Squares Learning via Classifier Systems

Let us consider an economic player, facing observations of a random return u_t, who does not know its mean, μ. How would the player estimate μ? One popular method would be just to take the equally weighted average of the observed returns. This gives nothing but the familiar ordinary least squares estimate of the mean. It is well known that this will be an optimal thing to do, provided that the returns are independent and identically distributed over time, no reliable prior information on μ, is available i.e. the prior is flat. In case a reliable prior probability density function for μ is available, the prior mean should be combined with the mean of observations with 'appropriate' weights. But what if the player lacks this Bayesian training, or does not have an idea about the data generating process?

As an alternative ad-hoc learning rule, the player can use the recursive least

squares formulation suggested in Sargent (1994):

$$\bar{u}_t = \bar{u}_{t-1} + (1/t)(u_t - \bar{u}_{t-1})$$

where $\bar{u}_t$ is the estimate of mean μ at time t. It should be noted that if t starts from 1, the initial condition, $\bar{u}_0$, is cancelled out at time 1, and hence is not important. In this case one can easily verify that at each time,

$$\bar{u}_t = (1/t)\sum_{s=1}^{t} u_s$$

i.e., just the sample mean.

If, however t starts from 2, the initial value, $\bar{u}_1$, only exponentially goes to zero, hence it matters. In this case, it stands for the prior mean in the Bayesian approach, hence it will be consistent with Bayesian update starting with some prior density. [1] Throughout the thesis, this approach of starting at $t = 2$ will be adopted, in order to allow for heterogeneity of initial values across players.

The two armed problem, however, has a further complication. There are two different actions with different returns and learning goes together with performance. This aspect makes the problem interesting and difficult.

By using the recursive least squares learning approach, strength of chosing action A will be updated by the formula,

$$SA_{t_A} = SA_{t_A-1} - \frac{1}{t_A}(SA_{t_A} - 5)$$

and experience counter of choosing action A will be updated as,

[1] This update scheme is a special case of the general stochastic approximation method of Robbins and Manro (1951) and hence convergence to the population mean will be attained. See Sargent (1994, pp. 39-42) for details.

$t_A = t_A + 1.$

For players who choose action B, the strength will be updated with 1 percent probability as,

$$SB_{t_B} = SB_{t_B-1} - \frac{1}{t_B}(SB_{t_B} - 1000)$$

or, with 99 percent probability as,

$$SB_{t_B} = SB_{t_B-1} - \frac{1}{t_B}(SB_{t_B})$$

and experience counter of choosing action B will be updated as,

$$t_B = t_B + 1.$$

3.2 Computer Algorithm and Simulation Results

The algorithm below involves three possible cases for decision making; normal, imitation and experimentation. In the normal case, each player decides using their own strengths of actions A and B. In the imitation case each player decides by using the social strengths for that time. Social strengths are calculated by using previous period's experience weighted average strength values of the society. In the experimentation case each individual player makes their decision randomly.The algorithm is,

```
Repeat until the last period.
  Repeat until all the players chose their actions.
    If there is no experimentation,
      If there is no imitation,
        If Strength of A is greater than Strength of B,
        choose action A otherwise choose action B.
      If there is imitation,
        if Social Strength of A is greater than Social Strength of B,
        choose action action A otherwise choose action B.
    If there is  experimentation,
      with 50 percent probability,
        choose action A,
      with 50 percent probability,
        choose action B.
  If action A is chosen,
    update the strength and experience counter of choosing action A,
    otherwise
    update the strength and experience counter of choosing action B.
  Update the Social Strength of choosing action A
    with respect to the previous period.
  Update the Social Strength of choosing action B
    with respect to the previous period.
```

(see Appendix A for the full Gauss Program)

Without experimentation players may not ever try an action. For example, if initial strength of choosing action A is greater than initial strength of choosing action B and initial strength of choosing action B is smaller than 5, player will always choose action A which has the lower return. Experimentation gives a chance to investigate different tastes and imitation helps for rapid convergence to the correct mode of behavior. We use the strength update scheme,

$$\bar{z}_t = \bar{z}_{t-1} + (\tfrac{1}{t})(z_t - \bar{z}_{t-1})$$

or,

$$\bar{z}_t = (1 - \tfrac{1}{t})\bar{z}_{t-1} + (\tfrac{1}{t} z_t)$$

$$\bar{z}_{t+1} = (1 - \tfrac{1}{t+1})\bar{z}_t + (\tfrac{1}{t+1} z_{t+1})$$

$$\bar{z}_{t+2} = (1 - \tfrac{1}{t+2})\bar{z}_{t+1} + (\tfrac{1}{t+2} z_{t+2})$$

$$\bar{z}_{t+2} = (1 - \tfrac{1}{t+2})((1 - \tfrac{1}{t+1})\bar{z}_t + (\tfrac{1}{t+1} z_{t+1})) + (\tfrac{1}{t+2} z_{t+2})$$

$$\bar{z}_{t+2} = (\tfrac{t}{t+2}\bar{z}_t) + (\tfrac{1}{t+2} z_{t+1}) + (\tfrac{1}{t+2} z_{t+2})$$

$$\bar{z}_{t+n} = (\tfrac{t}{t+n}\bar{z}_t) + (\tfrac{1}{t+n} \textstyle\sum_{i=1}^{n} z_{t+i}).$$

It should be noted that as n increases, the importance of initial strength $\bar{z}_t$ on $\bar{z}_{t+n}$ diminishes.

As n goes to infinity, convergence to the correct mean return parameter will be attained. For example assume that,

t is equal to 100,

$\bar{z}_t = 6$

$z_{t+i} = 5$ for $i \geq 1$

$5.05 \geq \bar{z}_{t+n}$ if $n \geq 2,000$

$5.005 \geq \bar{z}_{t+n}$ if $n \geq 20,000$

or if the initial strength t were larger, e.g. t is equal to 1000,

$5.05 \geq \bar{z}_{t+n}$ if $n \geq 20,000$

$5.005 \geq \bar{z}_{t+n}$ if $n \geq 200,000$.

With experimentation, whatever the initial strengths will be, convercenge to correct strength values will be satisfied. With imitation, this will happen rather fast. Expected return from choosing action A is equal to,

$$\begin{aligned} E(cA) &= 5 * 1 \\ &= 5 \end{aligned}$$

and expected return of choosing action B is equal to,

$$\begin{aligned} E(cB) &= (1000 * .01) + (0 * .99) \\ &= 10. \end{aligned}$$

In our simulations, in the beginning 50 percent of players choose action A and 50 percent of players choose action B because of their given initial strengths. Even-

tually, all the players end up choosing action B to get the highest utility. Social strength of choosing action A converges to 5 and social strength of choosing action B converges to 10. If we change the utility of choosing action B under good state of nature from 1000 to 100, expected return from choosing action B will fall to,

$$\begin{aligned} E(cB) &= (100 * .01) + (0 * .99) \\ &= 1. \end{aligned}$$

In this case the players choose action A to get the highest utility. Imitation also increases the speed of convergence. From the simulation results of the program, low imitation is observed to result in slow convergence, while high imitation leads to fast convergence.(see Appendix A for the Output)

4 Learning to use Money as a Medium of Exchange

We construct a physical environment like the model of Kiyotaki and Wright (1989) and observe the performance of classifier systems in a simple dynamic optimization problem. We consider a finite number of unexperienced players facing an infinite number of learned players. Then we let all the players be unexperienced. But first we recall the Kiyotaki and Wright (1989) model by using their own notation.

Time is discrete and continues forever, and at each date there are three indivisible commodities called goods 1, 2, and 3. β denotes the discount factor which is any nonnegative number smaller than or equal to one. Infinitely lived players are in equal proportions of type 1, type 2, and type 3, and they are randomly matched in each trade round. Type i players get their utility only from the consumption of good i. When they consume, type 1 player produces good 2, type 2 player produces good 3 and type 3 player produces good 1. Players can store all the goods at a cost and one good at a time. Storage cost of good 3 is the highest and that of good 1 is the lowest for all types of players. The storage cost of good k for type i player will be c_{ik}. Under our assumption, we can say that,

$$c_{i3} \geq c_{i2} \geq c_{i1}.$$

When type i player consumes good i and produces his good, he obtains utility U_i and disutility D_i. Utility is always greater than disutility. So each player has a trading strategy and wants to get his own type of good for consumption and therefore get highest utility. Otherwise he would prefer to choose any good which has a low storage cost. For example, type 2 player always prefers good 2 first, and then

good 1, and good 3 last or type 3 player always prefers good 3 first, and then good 1, and good 2 last, disregarding future trading opportunities. When type i player consumes good i and produces good k, the indirect utility will be,

$$V_{ii} = U_i - D_i + V_{ik}$$

or,

$$V_{ii} = u_i + V_{ik}$$

where $u_i = U_i - D_i$ and V_{ik} is the indirect utility of holding good k before the next trade round.

By using the Bellman equation of dynamic programming (Bertsekas 1976), the indirect utility for type i of storing good $k \neq i$ will be $-c_{ik}$ plus maximum of the expected value of next period's indirect utility multiplied with discount factor.

In Kiyotaki-Wright (1989), p_{ij} is defined as the density of type i player storing good j. Since type i player will always consume good i, type i player will not store good i before a trade round, so that we can write,

$$p_{12} + p_{13} = 1,$$

$$p_{21} + p_{23} = 1 \text{ and}$$

$$p_{31} + p_{32} = 1.$$

Type 1, type 2 and type 3 players are in equal propotions. The probability of type i player matching with any type of player is 1/3. If the players were concerned only with the storage costs and utilities, we would observe,

type 1 player storing good 2 trades for good 1 only,

type 1 player storing good 3 trades for good 1 or good 2,

type 2 player storing good 1 trades for good 2 only,

type 2 player storing good 3 trades for good 2 or good 1,

type 3 player storing good 1 trades for good 3 only,

type 3 player storing good 2 trades for good 3 or good 1.

The above pattern indeed emerges as an equilibrium the so called fundemantal equilibrium for a certain set of parameter values. In speculative equilibrium, players may concern not only the present trade but also the future trades and may not behave myopic. To obtain these values we can carry out the analyses below which involves also the missing steps in the Kiyotaki-Wright theorems. Our indirect utilities will be, for type 1 player:

$$V_{12} = -c_{12} + \beta[\tfrac{1}{3}V_{12} + \tfrac{1}{3}(p_{21}(u_1 + V_{12}) + p_{23}max(V_{12}, V_{13})) + \tfrac{1}{3}V_{12}]$$

$$V_{13} = -c_{13} + \beta[\tfrac{1}{3}V_{13} + \tfrac{1}{3}V_{13} + \tfrac{1}{3}(p_{31}(u_1 + V_{12}) + p_{32}max(V_{12}, V_{13}))]$$

for type 2 player:

$$V_{21} = -c_{21}+\beta[\tfrac{1}{3}(p_{12}(u_2+V_{23})+p_{13}max(V_{21}, V_{23}))+\tfrac{1}{3}V_{21}+\tfrac{1}{3}(p_{31}V_{21}+p_{32}(u_2+V_{23}))]$$

$$V_{23} = -c_{23} + \beta[\tfrac{1}{3}V_{23} + \tfrac{1}{3}V_{23} + \tfrac{1}{3}(p_{31}max(V_{21}, V_{23}) + p_{32}(u_2 + V_{23}))]$$

for type 3 player:

$$V_{31} = -c_{31}+\beta[\tfrac{1}{3}(p_{12}max(V_{31}, V_{32})+p_{13}(u_3+V_{31}))+\tfrac{1}{3}(p_{21}V_{31}+p_{23}(u_3+V_{31}))+\tfrac{1}{3}V_{31}]$$

$$V_{32} = -c_{32} + \beta[\tfrac{1}{3}V_{32} + \tfrac{1}{3}(p_{21}max(V_{31}, V_{32}) + p_{23}(u_3 + V_{31})) + \tfrac{1}{3}V_{32}].$$

By using the equations for type 1 player written above, assume that V_{12} is greater than V_{13}. Then;

$$V_{12} = -c_{12} + \beta[\tfrac{1}{3}V_{12} + \tfrac{1}{3}(p_{21}(u_1 + V_{12}) + p_{23}V_{12}) + \tfrac{1}{3}V_{12}]$$

$$V_{12} = -c_{12} + \beta[\tfrac{1}{3}V_{12} + \tfrac{1}{3}(p_{21}u_1 + V_{12}) + \tfrac{1}{3}V_{12}]$$

$$V_{12} = -c_{12} + \beta[V_{12} + \frac{1}{3}(p_{21}u_1)] \tag{1}$$

$$V_{13} = -c_{13} + \beta[\tfrac{1}{3}V_{13} + \tfrac{1}{3}V_{13} + \tfrac{1}{3}(p_{31}(u_1 + V_{12}) + p_{32}V_{12})]$$

$$V_{13} = -c_{13} + \beta[\frac{2}{3}V_{13} + \frac{1}{3}(p_{31}u_1 + V_{12})]. \tag{2}$$

After subtracting the two equations above,

$$V_{12} - V_{13} = c_{13} - c_{12} + \beta[\tfrac{2}{3}(V_{12} - V_{13}) + \tfrac{1}{3}(p_{21} - p_{31})u_1]$$

$$(1 - \tfrac{2}{3}\beta)(V_{12} - V_{13}) + (\tfrac{1}{3}\beta(p_{31} - p_{21})u_1) = c_{13} - c_{12}$$

$$c_{13} - c_{12} \geq (\tfrac{1}{3}\beta(p_{31} - p_{21})u_1).$$

If we assume,

$$c_{13} - c_{12} \geq (\tfrac{1}{3}\beta(p_{31} - p_{21})u_1),$$

we get,

$$-c_{12} \geq (\tfrac{1}{3}\beta(p_{31} - p_{21})u_1) - c_{13}.$$

By using the equation (1),

$$V_{12} \geq (\tfrac{1}{3}\beta(p_{31} - p_{21})u_1) - c_{13} + \beta[V_{12} + \tfrac{1}{3}(p_{21}u_1)]$$

$$V_{12} \geq (\tfrac{1}{3}\beta p_{31}u_1) - c_{13} + \beta V_{12}$$

$$V_{12} - (\tfrac{1}{3}\beta p_{31}u_1) - \beta V_{12} \geq c_{13}.$$

By using equation (2),

$$V_{12} - (\tfrac{1}{3}\beta p_{31}u_1) + \beta V_{12} + \beta[\tfrac{2}{3}V_{13} + \tfrac{1}{3}(p_{31}u_1 + V_{12})] \geq V_{13}$$

$$V_{12} - \beta V_{12} + \beta[\tfrac{2}{3}V_{13} + \tfrac{1}{3}V_{12}] \geq V_{13}$$

$$V_{12} + \beta[\tfrac{2}{3}V_{13} - \tfrac{2}{3}V_{12}] \geq V_{13}$$

$$[1 - \tfrac{2}{3}\beta]V_{12} \geq [1 - \tfrac{2}{3}\beta]V_{13}$$

since,

$$[1 - \tfrac{2}{3}\beta] \geq 0$$

we get,

$V_{12} \geq V_{13}$.

Therefore we can conclude that,

$$c_{13} - c_{12} \geq (\tfrac{1}{3}\beta(p_{31} - p_{21})u_1) \text{ iff } V_{12} \geq V_{13}.$$

By using the equations for type 2 player,

$$V_{21} = -c_{21} + \beta[\tfrac{1}{3}(p_{12}(u_2 + V_{23}) + p_{13}max(V_{21}, V_{23})) + \tfrac{1}{3}V_{21} + \tfrac{1}{3}(p_{31}V_{21} + p_{32}(u_2 + V_{23}))]$$

$$V_{23} = -c_{23} + \beta[\tfrac{1}{3}V_{23} + \tfrac{1}{3}V_{23} + \tfrac{1}{3}(p_{31}max(V_{21}, V_{23}) + p_{32}(u_2 + V_{23}))].$$

If we subtract them from each other,

$$V_{21} - V_{23} = -c_{21} + \tfrac{1}{3}\beta[(p_{12}(u_2 + V_{23}) + p_{13}max(V_{21}, V_{23})) + V_{21} + (p_{31}V_{21}$$

$$+p_{32}(u_2 + V_{23}))] + c_{23} - \tfrac{1}{3}\beta[V_{23} - V_{23} - (p_{31}max(V_{21}, V_{23}) - p_{32}(u_2 + V_{23}))]$$

$$V_{21} - V_{23} = -c_{21} + \tfrac{1}{3}\beta[(p_{12}(u_2 + V_{23}) + p_{13}max(V_{21}, V_{23})) + V_{21} + (p_{31}V_{21})]$$

$$+c_{23} - \tfrac{1}{3}\beta[V_{23} - V_{23} - (p_{31}max(V_{21}, V_{23}))].$$

Since,

$$c_{23} - c_{21} \geq 0 \text{ and } \beta[\tfrac{1}{3}(p_{12}u_2)] \geq 0$$

$$V_{21} - V_{23} \geq \tfrac{1}{3}\beta[(p_{12}V_{23}) + p_{13}max(V_{21}, V_{23})) + V_{21} + (p_{31}V_{21})]$$

$$-\beta\tfrac{1}{3}[V_{23} - V_{23} - (p_{31}max(V_{21}, V_{23}))].$$

Assuming that V_{23} is greater than V_{21},

$$V_{21} - V_{23} \geq \tfrac{1}{3}\beta[V_{23} + V_{21} + p_{31}V_{21} - V_{23} - V_{23} - P_{31}V_{23}]$$

$$(1 - \tfrac{1}{3}\beta(1 + p_{31}))(V_{21} - V_{23}) \geq 0.$$

Since,

$$(1 - \tfrac{1}{3}\beta(1 + p_{31})) \geq 0$$

we get,

$$(V_{21} - V_{23}) \geq 0.$$

So our supposition is wrong. Assuming that V_{21} is greater than V_{23},

$$V_{21} - V_{23} \geq \tfrac{1}{3}\beta[p_{12}V_{23} + p_{13}V_{21} + V_{21} + p_{31}V_{21} - V_{23} - V_{23} - P_{31}V_{21}].$$

Since,

$$p_{12}V_{23} + p_{13}V_{21} \geq V_{23}$$

we get,

$$V_{21} - V_{23} \geq \tfrac{1}{3}\beta[V_{21} - V_{23}]$$

$$(1 - \tfrac{1}{3}\beta)(V_{21} - V_{23}) \geq 0.$$

We know that $(1 - \frac{1}{3}\beta) \geq 0$. Then our assumption holds;

$V_{21} \geq V_{23}$.

By using the equations for type 3 player,

$$V_{31} = -c_{31} + \beta[\tfrac{1}{3}(p_{12}max(V_{31}, V_{32}) + p_{13}(u_3 + V_{31})) + \tfrac{1}{3}(p_{21}V_{31} + p_{23}(u_3 + V_{31})) + \tfrac{1}{3}V_{31}]$$

$$V_{32} = -c_{32} + \beta[\tfrac{1}{3}V_{32} + \tfrac{1}{3}(p_{21}max(V_{31}, V_{32}) + p_{23}(u_3 + V_{31})) + \tfrac{1}{3}V_{32}].$$

If we subtract them from each other,

$$V_{31} - V_{32} = -c_{31} + \tfrac{1}{3}\beta[(p_{12}max(V_{31}, V_{32}) + p_{13}(u_3 + V_{31})) + (p_{21}V_{31} + p_{23}(u_3 + V_{31}))$$

$$+\tfrac{1}{3}V_{31}] + c_{32} - \tfrac{1}{3}\beta[(p_{12}V_{32} + p_{13}(u_3 + V_{31})) + (p_{21}max(V_{31}, V_{32}) + p_{23}(u_3 + V_{31})) + V_{32}]$$

$$V_{31} - V_{32} = -c_{31} + c_{32} + \tfrac{1}{3}\beta[(p_{12}max(V_{31}, V_{32}) + p_{13}(u_3 + V_{31}))$$

$$+(p_{21}V_{31}) + V_{31} - p_{12}V_{32} - p_{13}(u_3 + V_{31}) - (p_{21}max(V_{31}, V_{32})) - V_{32}].$$

Since,

$$c_{32} - c_{31} \geq 0$$

we get,

$$V_{31} - V_{32} \geq \frac{\beta}{3}[p_{12}max(V_{31}, V_{32}) + p_{21}V_{31} + V_{31} - p_{12}V_{32} - p_{21}max(V_{31}, V_{32}) - V_{32}].$$

Assuming that V_{32} is greater than V_{31},

$$V_{31} - V_{32} \geq \tfrac{1}{3}\beta[(p_{21}+1)(V_{31} - V_{32})]$$

$$(1 - \tfrac{1}{3}\beta(1+p_{21}))(V_{31} - V_{32}) \geq 0$$

$$(V_{31} - V_{32}) \geq 0.$$

So our supposition is wrong. But assuming that V_{31} is greater than V_{32},

$$V_{31} - V_{32} \geq \tfrac{1}{3}\beta[(p_{12}+1)(V_{31} - V_{32})]$$

$$(1 - \tfrac{1}{3}\beta(1+p_{12}))(V_{31} - V_{32}) \geq 0$$

we get,

$$V_{31} \geq V_{32}.$$

Now our assumption holds. As a result, when we use the equations of V_{21}, V_{23}, V_{31} and V_{32}, we get the inequalities;

$V_{21} \geq V_{23}$ and

$V_{31} \geq V_{32}$ for all p_{ij} s.

Therefore for the equations of V_{12} and V_{13}, to yield the fundamental equilibrium, the statement will be,

$V_{12} \geq V_{13}$ iff $c_{13} - c_{12} \geq (\frac{1}{3}\beta(p_{31} - p_{21})u_1)$.

If $(\frac{1}{3}\beta(p_{31} - p_{21})u_1) \geq c_{13} - c_{12}$, we will have the same inequalities except the inequality which is V_{13} is greater than V_{12}, so steady state equilibrium is changed and we get the speculative equilibrium.

In fundamental equilibrium, the steady state inventory distributions will be,

$(p_{12}, p_{13}, p_{21}, p_{23}, p_{31}, p_{32}) = (1, 0, .5, .5, 1, 0)$.

(see Kiyotaki and Wright, 1989) Type 2 players trade good 3 for good 1 and all types of player wants get its own good. Good 1 is the unique medium of exchange and type 2 player acts as a middleman. So we have a fundamental equilibrium iff $c_{13} - c_{12} \geq .5\beta u_1$.

In speculative equilibrium, the steady state inventory distributions will be,

$(p_{12}, p_{13}, p_{21}, p_{23}, p_{31}, p_{32}) = (\frac{1}{\sqrt{2}}, 1 - \frac{1}{\sqrt{2}}, 2 - \sqrt{2}, \sqrt{2} - 1, 1, 0)$

(see Kiyotaki and Wright, 1989). In this case, we get $(\frac{\sqrt{2}-1}{3}\beta u_1) \geq c_{13} - c_{12}$ for speculative equilibrium. The difference from the fundamental equilibrium is that type 1 player uses good 3 as a medium of exchange as well. So type 1 player trades the low storage cost good 2 for the high storage cost good 3.

4.1 Learning Dynamic Optimization in a Stationary Environment and in the Kiyotaki-Wright Model of Money

In our simulations of the Kiyotaki-Wright model of money as a medium of exchange, we consider two cases, first a finite number of unexperienced players facing an infinite number of learned players, second all unexperienced in the begining. We study the

presence of convergence for both fundamental and speculative equilibrium parameters and observe the effects of experimentation and imitation in classifier systems. By using the assumptions above to get the fundamental equilibria, the proportions of the learned players like in the theoretical part must be,

$(p_{12}, p_{13}, p_{21}, p_{23}, p_{31}, p_{32}) = (1, 0, .5, .5, 1, 0)$,

or we can write the following table.

DIST. OF INV.	GOOD 1	GOOD 2	GOOD 3
TYPE 1 PLAYER	0	1	0
TYPE 2 PLAYER	0.5	0	0.5
TYPE 3 PLAYER	1	0	0

In the speculative equilibria, the proportions of the learned players like in the theoretical part must be,

$(p_{12}, p_{13}, p_{21}, p_{23}, p_{31}, p_{32}) = (\frac{1}{\sqrt{2}}, 1 - \frac{1}{\sqrt{2}}, 2 - \sqrt{2}, \sqrt{2} - 1, 1, 0)$,

or we also can write the following table.

DIST. OF INV.	GOOD 1	GOOD 2	GOOD 3
TYPE 1 PLAYER	0	0.707	0.293
TYPE 2 PLAYER	0.586	0	0.414
TYPE 3 PLAYER	1	0	0

As in the theoretical paper of Kiyotaki and Wright (1989), the pairs of players who decide to trade, will exchange their good, afterwards if consumption does not take place, each player will pay the storage costs of their goods and wait for the

next period's matching and for the different trading offer, if consumption takes place, each player will produce their own good and be ready for the next period's matching and trading round.

The logic of classifier systems is very simple, *to do* or *not to do* like in the two armed bandit example. If *do* strength is greater than *do not* strength, players will decide to carry out that action otherwise they will not.

In our simulation program, we have nine different possible trade states for each type of players. These are:

trade state 1 : trade good 1 in return to good 1,

trade state 2 : trade good 2 in return to good 1,

trade state 3 : trade good 3 in return to good 1,

trade state 4 : trade good 1 in return to good 2,

trade state 5 : trade good 2 in return to good 2,

trade state 6 : trade good 3 in return to good 2,

trade state 7 : trade good 1 in return to good 3,

trade state 8 : trade good 2 in return to good 3,

trade state 9 : trade good 3 in return to good 3.

For each type of players, there are nine different do trade strengths and nine different no trade strengths. In each trade round, if do trade strength is greater than the no trade strength for each player, we update the do trade strengths with the experience of trading, because both players decided to trade in this case. Otherwise we update the no trade strengths under the experience of non-trading.

For each type of players, we have three different possible consumption states. These are:

consumption state 1 : consume good 1,

consumption state 1 : consume good 2,

consumption state 1 : consume good 3.

For each type of players, there are three different do consumption strengths and three different no consumption strengths. In each consumption state, if do consumption strength is greater than no consumption strength, we update the do consumption strength and experience counter of consumption, otherwise we update the no consumption strength and experience counter of non-consumption. Like the trade case, players decide to consume or not by using their consumption strengths.

In our programs we have 20 players of each type. Hence there are 60 times 18 trade and 60 times 6 consumption classifiers which is equal to 1440 classifiers altogether. Initial strengths are taken randomly from normal density with mean zero and standard deviation one. At the end of the program, after 1000 periods, players have some experience and can learn how to behave. From the results of the output, one can observe that the players who consider the social strengths learn faster.

In the fundamental equilibrium case, type 1 players have only good 2 in their stores. Hence they may not have enough experience in trading good 3. Since there is no good 3 in type 1 players' storages, they also may not have enough experience in consuming or not consuming good 3. Type 2 players may not have enough experience for trading good 3 for good 2 because only type 1 players have good 2 and type 1 players do not accept the offer of type 2 because of the differences in storage costs. Type 3 players do not have any good 2, so they may not have enough experience for trading good 2 and not consuming good 2. Each type of players do not want

to trade their own kind of goods because they will consume and get the highest satisfaction without storing them. To have the convergence, disutiliy is important in consumption. If the players consume the wrong good, players can not get the the utility but will give the production cost named as disutility (compare with Marimon etal 1990 where they do not impose the disutility). As a result trade tables for fundamental equilibrium will be,

TRADE	GOOD1	GOOD1	GOOD2	GOOD2	GOOD3	GOOD3
FOR	GOOD2	GOOD3	GOOD1	GOOD3	GOOD1	GOOD2
TYPE1 Player	-	-	yes	no	don't care	don't care
TYPE2 Player	yes	no	-	-	yes	don't care
TYPE3 Player	no	yes	don't care	don't care	-	-

and consumption table will look like,

CONSUMPTION	GOOD 1	GOOD 2	GOOD 3
TYPE 1 PLAYER	yes	no	don't care
TYPE 2 PLAYER	no	yes	no
TYPE 3 PLAYER	no	don't care	yes

In the speculative equilibrium case, type 1 players may not have enough experience for trading good 3 for good 2 because the other type of players do not have any good 2 in their stores. Type 3 players do not have any good 2, so they may not have enough experience for trading good 2 and not consuming good 2. Again, no player will consume its own type of good. As a result trade table for speculative equilibrium will be,

TRADE	GOOD1	GOOD1	GOOD2	GOOD2	GOOD3	GOOD3
FOR	GOOD2	GOOD3	GOOD1	GOOD3	GOOD1	GOOD2
TYPE1 Player	-	-	yes	yes	yes	don't care
TYPE2 Player	yes	no	-	-	yes	yes
TYPE3 Player	no	yes	don't care	don't care	-	-

and consumption table will be,

CONSUMPTION	GOOD 1	GOOD 2	GOOD 3
TYPE 1 PLAYER	yes	no	no
TYPE 2 PLAYER	no	yes	no
TYPE 3 PLAYER	no	don't care	yes

By using the settings described above, we have written the corresponding programs in Gauss.

4.2 Computer Algorithm and Simulation Results

The algorithm of this program simulates the dynamic optimiztion problem in a stationary environment and in Kiyotaki-Wright Economic Environment. In imitation case players make their decisions by using social strength for that time but do not adopt the social strength. Social strengths are calculated by using previous period's experience weighted social average strength values over the same type of players. In the experimentation case each individual player makes the decision randomly. In the program, recursive least squares technique adopted to a dynamic programming setting is used for learning the correct strength values by update mechanism. At the first step, consumption strength update is calculated according to:

$$CONS_{t+1} = CONS_t - \frac{1}{TCONS_{t+1}}(CONS_t - U + DisU + SCOST - \beta * TRADE_t)$$

and

$$TCONS_{t+1} = TCONS_t + 1.$$

At the second step, trade strength update is calculated with the given consumption strength:

$$TRADE_{t+1} = TRADE_t - \frac{1}{TTRADE_{t+1}}(TRADE_t - CONS_{t+1})$$

and

$$TTRADE_{t+1} = TTRADE_t + 1.$$

The updating proceeds and may converge to some values after enough iterations. The technique is used for both do and not do strengths together. The algorithm then can be written as,

```
Repeat until the final round.
  Repeat until all the players are matched with an player.
      If there is imitation (trade),
        If Social Do Trade Strength of Each Type of Player is greater
           than Social No Trade Strength of Each Type of Player,
           decide to offer trade otherwise decide not to offer trade
      else if there is no imitation (trade),
```

```
  If Do Trade Strength of Player is greater
     than No Trade Strength of Player,
     decide to offer trade otherwise decide not to offer trade.
If there is  experimentation (trade),
  with 50 percent probability decide to do trade,
  with 50 percent probability decide not to trade.
If Consumption Flag is 1,(i.e. consumed the last period)
  update the strength of do consume classifier and
  experience counter
else if Consumption Flag is 0,
  update the strength of no consume classifier and
  experience counter.
If each pair of players decide to offer trade,
  exchange the stored goods of individual pairs,
  Trade Flag is 1.
If there is imitation (consumption),
  If Social Do Consumption Strength of Each Type of Player is
  greater than Social No Consumption Strength of Each Type of Player,
  decide to consume otherwise decide not to consume
else if there is no imitation (consumption),
  If Do Consumption Strength of Player is greater than
  No consumption Strength of Player,
  decide to consume otherwise decide not to consume.
If there is  experimentation,
  with 50 percent probability decide to do consume,
```

```
        with 50 percent probability decide not to consume.
      If decided to consume,
        type 1 player produces good 2,
        type 2 player produces good 3,
        type 3 player produces good 1.
        Consumption Flag is 1
      else decided not to consume,
        Consumption Flag is 0.
      If Trade Flag is 1,
        update the strength of do trade classifier and
        experience counter
      else if Trade Flag is 0,
        update the strength of no trade classifier and
        experience counter.
Report the results.
```

(see Appendix B and C for the full GAUSS Program)

Theoretically, we would expect the decision of a type 1 player in trading good 2 for good 3, be "NO" in fundamental equilibrium but "YES" in speculative equilibrium. In fundamental equilibrium type 1 players do not want to accept good 3 for good 2 because the storage costs' difference is large relative to utility from consuming own good. In speculative equilibrium type 1 players starts to exchange good 3 for good 1 with type 3 players. Good 1 is highly desirable for type 1 players and in order to enable this exchange good 3 becomes also desired for high utility values. For this

reason, type 1 players use good 3 as a medium of exchange and transfers form type 2 players to type 3 players. In fundamental equilibrium however, good 1 is the only medium of exchange as expected because good 1 has the lowest storage cost. (see Appendix B and Appendix C for the Output)

Our simulation results indicate that the higher the imitation rate, the faster is the convergence. The inventory stok distributions also become more similar to the theoretical distributions when the imitation rate is increased.

The finite number of unexperienced players facing an infinite number of learned players are observed to converge to correct decisions faster than their counterparts in the program with all unexperienced players. (compare the results in Appendix D and in Appendix E) The players who face with the learned players obviously can learn the system better than the players who face with the unexperienced players.

Regarding experimentation, taking actions that may look like mistakes indeed may help discovering higher rewards. That is why even without imitation, we observe convergence, despite slow to the expected equilibrium behavior. Imitation is the tool for bringing together the collective experience and hence introducing it speeds up the rate of convergence in all of our simulations. However it is also observed that increasing imitation rate to 100 percent may also hurt in some cases as far as speed of convergence is concerned.

5 Summary and Discussion of the Results

From the two armed bandit problem, we observe that the mistakes of each player cause finding better opportunities and lead to convergence to the correct decisions which give the highest utility. Since the percentage of players' mistakes is bounded by a small number, no instability in the system is caused. We also observe that imitation increases the speed of convergence. In the imitation case, the higher the probability of imitation, the faster is the convergence. With the results in light of this section, we try to observe the behavior of the bounded rational players in a more complicated system which is Kiyotaki and Wright (1989) model of money as a medium of exchange.

First we construct a model in which a finite number of players face an infinite number of learned players. This makes the model simple in the first step, so we can construct and test our simulation program in a less complicated environment. In this case, we see that imitation increases the speed of convergence and the players learn how to behave and converge to the expected decisons from a low proportion of players' mistakes. To observe the robustness of the convergence results, 100 independent Monte-Carlo simulations are run over a 1000 time periods. By using the parameter values in line with theoretical part, we get the fundamental and speculative equilibria settings. All types of players learn how to consume their type of goods and not to consume other types of goods. In both of the equilibria, players first want to trade for their own good, if this is not possible, they prefer the low storage cost good. The simulation results provide us approximately the same inventory distributions as expected from the decisions of players at the theoretical fundamental and speculative equilibria. (see Appendix D)

Finally, we constructed the final model, the real Kiyotaki-Wright economic envi-

ronment as a dynamic game with a finite number of players, all unexperienced in the begining. To observe the robustness of the convergence results as in the previous part, 100 independent Monte-Carlo simulations are run over a 1000 time periods. The speed of convergence is observed to decrease in most cases beacuse none of the players know how to behave in the beginning unlike in the previous case. We also observe that imitation increases the speed of convergence and the players learn how to behave and converge to the correct decisons under a bounded percentage of players' mistakes. Like in the dynamic stationary case, we get approximately the theoretical inventory distributions. In the fundamental equilibrium setting, type 2 player is observed to act as the middleman, transferring good 1 from type 3 player to type 1. Good 1 is the medium of exchange and acts as money. In the speculative case, type 1 players start to exchange good 2 for good 3 even if the storage cost of good 2 is lower than the good 3. Since the utility of consuming own good is very high with respect to the storage costs' differences in the speculative equilibria, type 1 players ignore the difference of the storage cost of good 2 to the storage cost of good 3 and start to transfer good 3 from player 2 to player 3. As a result type 1 players consume their own good more than before by transferring good 3. In this case, good 3 also acts as a medium of exchange. (see Appendix E)

In Brown's (1996) laboratary study, with human subjects where no imitation is allowed, students react but are far from equilibrium and exhibit convergence with more iterations. If we contrast our computer simulation with Brown (1996)'s, his experimental results are in line with our findings.

In summary, our simulations show that in order to approach the theoretical equilibria, a small probability of experimentation is necessary. Moreover imitation increases the speed of convergence process.

Bibliography

1. Andreoni, J. A., J. H. Miller, 1990, Auctions with adaptive artificially intelligent agents, Santa Fe Institute Working Paper no. 90-01-004.

2. Arthur, W. B., 1990, A learning algorithm that replicates human learning, Santa Fe Institute Working Paper no. 90-026.

3. Axelrod, R., 1987, The evolution of strategies in the iterated prisoner's dilemma, Genetic Algorithms and Simulated Annealing, London, Pitman.

4. Bertsekas, D. P., 1976, Dynamic programming and stochastic control, New York, Academic Press.

5. Brown, P. M., 1996, Experimental evidence on money as a medium of exchange, Journal of Economic Dynamics and Control 20, 583-600.

6. Booker, L. B., D. E. Goldberg and J. H. Holland, 1989, Classifier systems and genetic algorithms, Artificial Intelligence 40, 235-282.

7. Cuadras-Morato, X., 1994, Commodity money in the presence of goods of heterogeneous quality, Economic Theory 4, 579-591.

8. Gale, D. M., 1986, Bargaining and competition part I: Characterization, Econometrica 54, 785-806.

9. Gale, D. M., 1986, A strategic model of trading with money as the medium of exchange, Working Paper no. 86-04, University of Pennsylvannia, Center Analytic Res. Econ. and Soc. Sciences.

10. Holland, J. H., 1975, Adaptation in neural and artifical systems, University of Michigan Press, Ann Arbor.

11. Holland, J. H., A mathematical framework for studying learning in classifier systems, 1986, Physica 22D, 307-317.

12. Holland, J. H. and J. Miller, 1991, Artifical adaptive agents in economic theory, American Economic Review 81, 365-370.

13. Iwai, K., 1988, The evolution of money: A search theoretic foundation of monetary economics, Working Paper no. 83-03, University of Pennsylvannia, Center Analytic Res. Econ. and Soc. Sciences.

14. Jones, R. A., 1976, The origin and deveplopment of media of exchange, Journal of Political Economy 84, 757-775.

15. Kehoe, T., N. Kiyotaki and R. Wright, 1993, More on money as a medium of exchange, Journal of Economic Theory 3, 297-309.

16. Kiyotaki, N. and R. Wright, 1989, On money as a medium of exchange, Journal of Political Economy 97, 927-954.

17. Kiyotaki, N. and R. Wright, 1991, A contribution to the pure theory of money, Journal of Economic Theory 53, 215-235.

18. Kiyotaki, N. and R. Wright, 1993, A search-theoretic approach to monetary economics, American Economic Review 83, 63-77.

19. Lettau, M. and H. Uhlig, 1993, Classifier systems and dynamic programming, Princeton University.

20. Marimon, R., E. McGrattan and T. Sargent, 1990, Money as a medium of exchange in an economy with artificially intelligent agents, Journal of Economic Dynamics and Control 14, 329-373.

21. Miller, J. H., 1989, The coevolution of automata in the repeated prisoner's dilemma, Santa Fe Institute Working Paper no. 89-003.

22. Mortensen, D. T., 1982, The matching process as a noncooperative bargaining game, The Economics and Information and Uncertainty, edited by John J. McCall, University of Chicago Press.

23. Robins, Marro, 1951, A stochastic approximation method, Annals of Mathematical Statistics, 22:400-7.

24. Rubinstein, A., A. Wolinsky, 1985, Equilibrium in a market with sequential bargaining, Econometrica 53, 1133-1150.

25. Sargent, T., 1994, Bounded Rationality in Macroeconomics, Oxford, Clarendon Press, 39-42.

26. Valiant, L. G., 1984, A theory of learnable, Communications of the ACM 24, 1134-1142.

27. Yiting, Li, 1995, Commodity money under private information, Journal of Monetary Economics 36, 573-592.

APPENDIX A

Program and Results of Two Armed Bandit Part:

1. Computer Program
2. Output: Full Imitation Case
3. Output: No Imitation Case

```
/* This program simulates the repeated static decision problem called the */
/* two armed bandit problem */
/* In imitation case, each agent decides by using social
   strengths for the previous t but does not adopt the social strength
   for the future use */
TIMENUM=3000;  /* Number of periods*/
N1=30;         /* Number of agents who choose A or B */
N=2*N1;        /* Number of players */
uA=5;          /* Utility of action A */
uBg=1000;      /* Utility of action B under good state of nature
                                                (probability pgood) */
uB=0;          /* Utility of action B under bad state of nature
                                                (probability (1-pgood)) */
pgood=0.01;   /* probability of good state of nature */
pimit=1;    /* probability of imitation */
pexp=0.05;     /* probability of experimentation */
SAMPLE=20;    /* number of samples over the time period */
                                /* initialization */
SA=10*ones(N1,1)|6*ones(N1,1);  /* strength of A */
SAi=SA;                         /* initial strength of A is recorded */
TA=ones(N,1);                   /* number of times action A is chosen */
SB=6*ones(N1,1)|10*ones(N1,1);  /* strength of B */
SBi=SB;                         /* initial strength of B is recorded */
TB=ones(N,1);                   /* number of times action B is chosen */
SSA=(SA'*TA)/(sumc(TA));        /* social strength of A, avg. value */
SSB=(SB'*TB)/(sumc(TB));        /* social strength of B, avg. value */
tpropA=zeros(SAMPLE,1);         /* proportion over time of players playing A */
den=zeros(N,1);                 /* difference of SA-SB */
t=1;
h=1;                            /* within subsample counter */
t2=0;                           /* subsample counter */
do while t<TIMENUM+1;
 t=t+1;
 h=h+1;
 i=1;
 do while i<N+1;
  if rndu(1,1)<(1-pexp);        /* no experimentation */
    if rndu(1,1)<(1-pimit);     /* no imitation */
      if SA[i] > SB[i];
       choseA=1;
      else;
       choseA=0;
      endif;                  /* end of if loop for choosing an action */
    else;                                       /* imitate */
      if SSA > SSB;
       choseA=1;
      else;
       choseA=0;
      endif;
    endif;                  /* end of imitation */
  else;                                         /* experiment */
     if rndu(1,1)<0.5;
      choseA=1;
     else;
      choseA=0;
```

```
    endif;                    /* end of if loop for choosing an action */
 endif;                       /* end of experimentation */
 if choseA==1;
  SA[i]=SA[i] - ( (1/(TA[i]+1))*(SA[i]-uA) );  /* strength update */
  TA[i]=TA[i]+1;                               /* if A is chosen */
 else;
  if rndu(1,1) < pgood;                        /* if B is chosen */
   SB[i]=SB[i] - ( (1/(TB[i]+1))*(SB[i]-uBg) );
  else;
   SB[i]=SB[i] - ( (1/(TB[i]+1))*(SB[i]-uB) );
  endif;                 /* end of strength update in case B is chosen */
  TB[i]=TB[i]+1;
 endif;
 i=i+1;                       /* next agent */
endo;                         /* end of while loop  for players */
SSA=(SA'*TA)/(sumc(TA));      /* social strength of A, avg. value */
SSB=(SB'*TB)/(sumc(TB));      /* social strength of B, avg. value */
SABdiff=SA[.,1]-SB[.,1];
propA=sumc(SABdiff.>=0);
if h==TIMENUM/SAMPLE;
   t2=t2+1;
   tpropA[t2]=propA/N;
   h=0;
   print t;
endif;
endo;                         /*end of while loop  for time*/
```

```
WITH IMITATION
 In imitation case, each agent make their decision by using social
 strength for that time but do not change their strength.
Final Date        3000.0000
Initial number of players choosing each action (A or B)   30.000000
Number of players        60.000000
Utility of choosing action A        5.0000000
Utility of choosing action good state of nature        1000.0000
Utility of choosing action bad state of nature        0.0000000
probabilty of good state of nature      0.010000000
probability of imitation        1.0000000
probability of experimentation      0.050000000
number of sampling period        20.000000
difference of SA-SB at final time       -2.1296444       -3.7860205
      -4.5056391       -6.8979836       -5.1759589       -8.8486555
      -5.8634434       -6.5610478       -5.8680788       -5.1625414
      -5.5398646       -9.6143338       -6.1905805       -5.2107810
      -1.4235527       -6.5658564       -4.8081670       -2.4120043
      -1.7130796       -5.8587858       -7.9179592       -3.1495039
      -6.8678620       -4.8345454       -1.7934761       -5.9089477
      -8.0010494       -5.8999833       -3.1778632       -3.8181984
      -5.6303659       -6.9987973       -2.8220419       -6.6060722
     -0.46051726       -1.5028314       -5.2871950       -3.5001074
      -3.5250518       -4.5699624       -3.5682656       -6.6350652
      -6.6226480       -4.8799844       -5.6454878       -7.6489908
      -4.8512179       -4.2078787       -3.8825283       -2.5445485
      -2.8849988       -5.5430083       -5.2025248       -7.0030495
      -3.8761107       -5.2173518       -5.9306027       -3.8985376
      -5.9188635       -11.742569
social strength of A at final time        5.0393013
social strength of B at final time        10.108693
time plot of proportion of players who choose action A at some intervals
       0.30000000       0.21666667       0.21666667       0.11666667
      0.050000000      0.050000000      0.083333333      0.033333333
      0.033333333      0.050000000      0.050000000      0.050000000
      0.016666667        0.0000000        0.0000000        0.0000000
        0.0000000        0.0000000        0.0000000        0.0000000
```

```
WITHOUT IMITATION
 In imitation case, each agent make their decision by using social
 strength for that time but do not change their strength.
Final Date        3000.0000
Initial number of players choosing each action (A or B) 30.000000
Number of players        60.000000
Utility of choosing action A        5.0000000
Utility of choosing action good state of nature        1000.0000
Utility of choosing action bad state of nature        0.0000000
probabilty of good state of nature      0.010000000
probability of imitation        0.0000000
probability of experimentation      0.050000000
number of sampling period        20.000000
difference of SA-SB at final time       -11.791525        4.9107939
      -6.1768357        4.9237873       -10.971713       -9.6060207
       4.9247869       -9.5082631        4.9121513       -6.1554314
      -5.4661359        4.9294238        4.9134689       -5.3275524
      -4.1948712       -4.4981965        4.9093947        4.9217082
      -1.5237694        4.9257612       -9.9293472       -9.6024324
       4.9000040        4.9237873       0.21131464       -4.7776604
      0.32272428        4.9357836       -3.8563881        4.9016995
      -4.7301709       -11.536632       -5.1976191        4.8390498
      -4.2605807        4.8670083       -3.2593265        4.8826958
       4.8783912       -5.6097792       -9.0541665        4.8854005
      -5.5506624        4.8670083       -7.7278558        4.8532820
      -6.3957138       -6.5717864        4.8737598       0.27726070
     0.010460660       -6.6780990       -6.4988753       -5.4307473
      -7.7364113       -2.5214485        4.8826958       -4.7691550
       4.8721369        4.8783912
social strength of A at final time        5.0014981
social strength of B at final time        10.413728
time plot of proportion of players who choose action A at some intervals
      0.91666667       0.88333333       0.86666667       0.86666667
      0.81666667       0.80000000       0.75000000       0.71666667
      0.68333333       0.66666667       0.65000000       0.60000000
      0.58333333       0.56666667       0.51666667       0.50000000
      0.48333333       0.48333333       0.48333333       0.46666667
```

APPENDIX B

Programs and Results of Dynamic Optimization Part:

When we look at the table of outputs for trade and consumption, we get the proportion of players who decide to do action at the last time period. In the trade table, i-j means trade for good i in return to good j. For example, when we look at the row for type 1 player and column for 2-3, we get the proportion of type 1 players who decide to do trade good 2 in return to good 3. In the consumption table, the rows give us the type of players and columns give us the type of goods. In the stok table, the distribution of holding goods for each type of players is given.

1. Computer Program: Fundamental Equilibrium
2. Output: Fundamental Equilibrium and Full Imitation Case
3. Output: Fundamental Equilibrium and No Imitation Case
4. Computer Program: Speculative Equilibrium
5. Output: Speculative Equilibrium and Full Imitation Case
6. Output: Speculative Equilibrium and No Imitation Case

```
/*  This program simulates the Kiyotaki-Wright Economy with artificially
intelligent players
    Distinguishing features:
      - Production costs present (cf. Marimon etal.1990)
      - Mistake or experimentation present at 2.5% probability.
      - Imitation as occationally acting according to
        social strengths (with probability pimit).
      - Social strengths are experience weighted average individual
        strengths over a type of players calculated before every
        trade decision.
      - In strength updates if agent has imitated, social strengths
        of the following decision are used rather than individual strengths.
      - Imitation in consumption classifiers as well
      - Incorporates only the fundamental equilibrium (i.e. only good
        one is used as medium of exchange.
    Reporting facilities:
      - At each time, for each type, pre-trade stock distribution is recorded
        in matrix SDIST (T by 9) (1-3: first  type, goods 1-3,
                                  4-6: second type, goods 1-3,
                                  7-9: third  type, goods 1-3)
      - At each time, for each type and each state, consumption
        decision distribution is recorded in matrix
        CDIST (T by 9)  (1-3: first  type, goods 1-3, do consume perc.
                         4-6: second type, goods 1-3, do consume perc.
                         7-9: third  type, goods 1-3, do consume perc.)
      - At each time, for each type and each state, trade
        decision distribution is recorded in matrix
        TDIST (T by 27)  ( 1- 9: first  type, states 1-9, do trade perc.
                          10-18: second type, states 1-9, do trade perc.
                          19-27: third  type, states 1-9, do trade perc.)*/
N1=20;            /* Number of players of each type (select as even)  */
N=3*N1;          /* The total number of players   */
ref=ones(N1,1)|(N1+1)*ones(N1,1)|(2*N1+1)*ones(N1,1);
                                  /* for referencing purposes */
beta=0.6;       /* The discount factor (select between zero and one) */
pr=0.05;        /* The probability of mistake */
pim=1;                 /Imitation*/
pimit=pim*ones(1,N);  /* Homogeneous */
u11= 200;           /* The utilities from consuming own type of good  */
u22= 200;
u33= 200;
D1=-100;              /* The disutilities from production for each agent */
D2=-100;
D3=-100;
U1=(u11+D1)*ones(N1,1)~D1*ones(N1,1)~D1*ones(N1,1);
U2=D2*ones(N1,1)~(u22+D2)*ones(N1,1)~D1*ones(N1,1);
U3=D3*ones(N1,1)~D3*ones(N1,1)~(u11+D3)*ones(N1,1);
U=U1|U2|U3;      /* The N by 3 matrix of utilities */
c11=0.1;           /* The inventory holding costs  */
c12=10;           /* Type 1 agent, good 2        */
c13=25;
c21=0.1;
c22=10;
c23=25;
c31=0.1;
```

```
c32=10;
c33=25;
C1=c11*ones(N1,1)~c12*ones(N1,1)~c13*ones(N1,1);
C2=c21*ones(N1,1)~c22*ones(N1,1)~c23*ones(N1,1);
C3=c31*ones(N1,1)~c32*ones(N1,1)~c33*ones(N1,1);
C=C1|C2|C3;                    /* The N by 3 matrix of costs  */
/* Initial strengths will be random  */
SDOTR=rndn(N,9);      /* The strengths of players' do trade
                         classifiers (3x3=9 states) */
TDOTR=1*ones(N,9);      /* Times used */
SNOTR=rndn(N,9);      /* The strengths of players' don't trade
                         classifiers  */
TNOTR=1*ones(N,9);        /* Times used */
SDOCNS=rndn(N,3);      /* The strengths of players' do consume
                         classifiers (3 states) */
TDOCNS=1*ones(N,3);      /* Times used */
SNOCNS=rndn(N,3);     /* The strengths of players' don't consume
                         classifiers   */
TNOCNS=1*ones(N,3);      /* Times used */
STOK=2*ones(N1,1)|3*ones(N1,1)|ones(N1,1); /* The beginning pre-trade
                                              inventory vector  */
STOKC=ones(N,1);     /* Pre-consumption inventory held */
CNS=zeros(N,1);   /* Most recent consumption indicator. Consumed=1 Not=0 */
FINALD=1000;    /* Final date -- Number of times market opens */
AVR=100;       /* Averages of last markets which are openned */
SDIST=zeros(FINALD,9);     /* The stock distribution recorder */
SDIST[1,2]=1;              /* Starting distribution of inventories */
SDIST[1,6]=1;
SDIST[1,7]=1;
CDIST=zeros(FINALD,9);     /* The do consume percentage recorder */
TDIST=zeros(FINALD,27);    /* The do trade percentage recorder  */
/* Initialization part complete.  Now the market opens  */
/*  The main loop running over time index t.  */

t=0;
do while t<(FINALD);
 t=t+1;
 i=1;
 do while i<N+1;   /* i will be matched with a 'learned' agent  */
  rtype=rndu(1,1);      /* (FUNDAMENTAL EQM) */
  if rtype<1/3;
   ltype=1;         /* first type always comes with good 2 */
   lstock=2;
  elseif rtype<2/3;
     ltype=2;         /* second type comes with either good 1 (50%) */
     if rndu(1,1)<.5;
        lstock=1;
     else;            /* or good 3 (50%)   */
        lstock=3;
     endif;
  else;
    ltype=3;   /* third type always comes with good 1 */
    lstock=1;
  endif;
  TRADE1=0;                          /* Initialization of trade  */
```

```
TRADE2=0;                              /*  offer decisions.        */
STATE1=3*(STOK[i]-1)+lstock;           /* Pre-trade */
if ltype==1;         /* he will always carry good 2 (pre-trade) */
  if STOK[i]==1;   /* say 'yes' only if partner comes with good 1 */
      TRADE2=1;
  endif;
endif;
if ltype==2;         /* he will either carry (pre-trade) */
  if lstock==1;        /* good 1 */
    if STOK[i]==2;   /* (says 'yes' only if partner comes with good 1) */
      TRADE2=1;
    endif;
  elseif lstock==3;   /* or good 3 */
    if ( (STOK[i]==1) or (STOK[i]==2) );
       TRADE2=1;        /* (says 'yes' to everything) */
    endif;
  endif;
endif;
if ltype==3;         /* he will always carry good 1 (pre-trade) */
  if STOK[i]==3;   /* say 'yes' only if partner comes with good 3 */
      TRADE2=1;
  endif;
endif;
/* First, the possibility of imitation  */
if rndu(1,1)<pimit[i];     /* If decides to look around */
    /* For this state, he emulates the weighted average
       of strengths among his types for this state  */
    SOCDT=SUMC(SDOTR[ref[i]:ref[i]+N1-1,STATE1].*
         (TDOTR[ref[i]:ref[i]+N1-1,STATE1]-1))/
         (SUMC(TDOTR[ref[i]:ref[i]+N1-1,STATE1]-1)+1);
    SOCNT=SUMC(SNOTR[ref[i]:ref[i]+N1-1,STATE1].*
         (TNOTR[ref[i]:ref[i]+N1-1,STATE1]-1))/
         (SUMC(TNOTR[ref[i]:ref[i]+N1-1,STATE1]-1)+1);
   STRADE1=SOCNT;     /*  Trade strength -- default: social no trade */
   if SOCDT>=SOCNT;   /* Decision according to social values */
      TRADE1=1;
      STRADE1=SOCDT;  /* Trade strength -- change: social do trade */
   endif;
else;  /* No imitation -- own strengths */
   STRADE1=SNOTR[i,STATE1];     /*  Trade strength  -- default: no trade */
/*  Now trade offer decisons will be made. Note the bias towards
    trade in case of equality of strengths */
  if SDOTR[i,STATE1]>= SNOTR[i,STATE1];
     TRADE1=1;                       /* First agent decides */
     STRADE1=SDOTR[i,STATE1];     /* Trade strength -- change : do trade */
  endif;
endif;
/* Now the possibility of non-deliberate decision */
if rndu(1,1)<pr;  /* Random choice: Agent i */
  if rndu(1,1)<0.5;
    TRADE1=1;         /* 50 percent trade */
    STRADE1=SDOTR[i,STATE1];
  else;
    TRADE1=0;         /* 50 percent no trade */
    STRADE1=SNOTR[i,STATE1];
```

```
   endif;
endif;
 /* Now strength update of consumption classifiers will be made */
 if t>1;   /* Start update at time 2, since no consumption
              decision is made for t=1 yet. */
           /* First agent first  */
  if CNS[i]==1;
    SDOCNS[i,STOKC[i]]=SDOCNS[i,STOKC[i]]
    -(1/(TDOCNS[i,STOKC[i]]+1))
    *(SDOCNS[i,STOKC[i]]
    -U[i,STOKC[i]]
    +C[i,STOK[i]]
    -beta*STRADE1);
    TDOCNS[i,STOKC[i]]=TDOCNS[i,STOKC[i]]+1;
  else;
    SNOCNS[i,STOKC[i]]=SNOCNS[i,STOKC[i]]
    -(1/(TNOCNS[i,STOKC[i]]+1))
    *(SNOCNS[i,STOKC[i]]
    +C[i,STOK[i]]
    -beta*STRADE1);
    TNOCNS[i,STOKC[i]]=TNOCNS[i,STOKC[i]]+1;
  endif;
 endif;
 /* Now back to the trade round */
 TRADE=TRADE1*TRADE2;
 if TRADE==1;
  STOKC[i]=lstock;          /* If exchange takes place */
 else;                             /* else, prev. stocks   */
  STOKC[i]=STOK[i];      /* are kept.           */
 endif;
 /* Now the consumption decision will be made  */
 /* (Note the bias towards consuming in case of equal strengths */
 /* First, the possibility of imitation (consumption strengths) */
if rndu(1,1)<pimit[i];     /* If decides to look around */
  /* For this state (commodity), he acts according to the weighted average
        of strengths among his types for this state  */
   GOOD1=STOKC[i];
   SOCDC=SDOCNS[ref[i]:ref[i]+N1-1,GOOD1]'*
         (TDOCNS[ref[i]:ref[i]+N1-1,GOOD1]-1)/
         (SUMC(TDOCNS[ref[i]:ref[i]+N1-1,GOOD1]-1)+1);
   SOCNC=SNOCNS[ref[i]:ref[i]+N1-1,GOOD1]'*
         (TNOCNS[ref[i]:ref[i]+N1-1,GOOD1]-1)/
         (SUMC(TNOCNS[ref[i]:ref[i]+N1-1,GOOD1]-1)+1);
   SCONS1=SOCNC;   /*  Consumption strength -- default: social no consume */
   CNS[i]=0;
   STOK[i]=STOKC[i];           /* Same stock  */
   if SOCDC>=SOCNC;  /* Decision according to social values */
       SCONS1=SOCDC; /* Consumption strength -- change: social no consume */
       CNS[i]=1;
       if i<=20;               /* first type  */
       STOK[i]=2;       /* type 1 produces 2 */
       elseif i<=40;          /* second type */
         STOK[i]=3;   /* type 2 produces 3 */
       else;                  /* third type  */
         STOK[i]=1;   /* type 3 produces 1 */
```

```
        endif;
     endif;
else;  /* no imitation */
  if SDOCNS[i,STOKC[i]]>=SNOCNS[i,STOKC[i]];
     CNS[i]=1;                          /* do consume  */
     SCONS1=SDOCNS[i,STOKC[i]];              /* its value (strength) */
     if i<=20;                /* first type  */
     STOK[i]=2;       /* type 1 produces 2 */
      elseif i<=40;           /* second type */
        STOK[i]=3;   /* type 2 produces 3 */
      else;                   /* third type  */
        STOK[i]=1;   /* type 3 produces 1 */
     endif;
  else;
     CNS[i]=0;
     STOK[i]=STOKC[i];           /* Same stock  */
     SCONS1=SNOCNS[i,STOKC[i]];  /* its value (strength) */
  endif;
endif;
  /* Now the possibility of non-deliberate decision */
if rndu(1,1)<pr;  /* Random choice: Agent i */
    if rndu(1,1)<0.5;
      CNS[i]=0;                          /* don't consume  */
      SCONS1=SNOCNS[i,STOKC[i]];  /* its value (strength) */
      STOK[i]=STOKC[i];           /* Same stock  */
    else;
      CNS[i]=1;                          /* do consume */
      SCONS1=SDOCNS[i,STOKC[i]];  /* its value (strength) */
      if i<=20;                /* first type  */
      STOK[i]=2;       /* type 1 produces 2 */
       elseif i<=40;           /* second type */
        STOK[i]=3;   /* type 2 produces 3 */
       else;                   /* third type  */
        STOK[i]=1;   /* type 3 produces 1 */
      endif;
    endif;
endif;
  /* Now, update of trade classifier strengths  */
  if TRADE1==1;              /* Agent i */
    SDOTR[i,STATE1]=SDOTR[i,STATE1]
         -(1/(TDOTR[i,STATE1]+1))*(SDOTR[i,STATE1]-SCONS1);
    TDOTR[i,STATE1]=TDOTR[i,STATE1]+1;
   else;
    SNOTR[i,STATE1]=SNOTR[i,STATE1]
       -(1/(TNOTR[i,STATE1]+1))*(SNOTR[i,STATE1]-SCONS1);
    TNOTR[i,STATE1]=TNOTR[i,STATE1]+1;
  endif;
  i=i+1;   /* Next agent...  */
 endo;
 /* Now records for reporting purposes */
  /* Stock distribution */
 SDIST[t,1]=SUMC(STOK[1:N1].==1)/N1;  /* first type */
 SDIST[t,2]=SUMC(STOK[1:N1].==2)/N1;
 SDIST[t,3]=SUMC(STOK[1:N1].==3)/N1;
 SDIST[t,4]=SUMC(STOK[N1+1:2*N1].==1)/N1;  /* second type */
```

```
SDIST[t,5]=SUMC(STOK[N1+1:2*N1].==2)/N1;
SDIST[t,6]=SUMC(STOK[N1+1:2*N1].==3)/N1;
SDIST[t,7]=SUMC(STOK[2*N1+1:3*N1].==1)/N1;  /* third type */
SDIST[t,8]=SUMC(STOK[2*N1+1:3*N1].==2)/N1;
SDIST[t,9]=SUMC(STOK[2*N1+1:3*N1].==3)/N1;
 /* Do consume decision percentage over goods and types */
 NSDOCNS=SDOCNS-SNOCNS;       /* Net strengths of do consume classifiers */
 CDIST[t,1]=SUMC(NSDOCNS[1:N1,1].>=0)/N1;         /* first type */
 CDIST[t,2]=SUMC(NSDOCNS[1:N1,2].>=0)/N1;
 CDIST[t,3]=SUMC(NSDOCNS[1:N1,3].>=0)/N1;
 CDIST[t,4]=SUMC(NSDOCNS[N1+1:2*N1,1].>=0)/N1;     /* second type */
 CDIST[t,5]=SUMC(NSDOCNS[N1+1:2*N1,2].>=0)/N1;
 CDIST[t,6]=SUMC(NSDOCNS[N1+1:2*N1,3].>=0)/N1;
 CDIST[t,7]=SUMC(NSDOCNS[2*N1+1:3*N1,1].>=0)/N1;   /* third type */
 CDIST[t,8]=SUMC(NSDOCNS[2*N1+1:3*N1,2].>=0)/N1;
 CDIST[t,9]=SUMC(NSDOCNS[2*N1+1:3*N1,3].>=0)/N1;
 /* Do trade decision percentage over goods and types */
 NSDOTR=SDOTR-SNOTR;       /* Net strengths of do consume classifiers */
 TDIST[t,1]=SUMC(NSDOTR[1:N1,1].>=0)/N1;          /* first type */
 TDIST[t,2]=SUMC(NSDOTR[1:N1,2].>=0)/N1;
 TDIST[t,3]=SUMC(NSDOTR[1:N1,3].>=0)/N1;
 TDIST[t,4]=SUMC(NSDOTR[1:N1,4].>=0)/N1;
 TDIST[t,5]=SUMC(NSDOTR[1:N1,5].>=0)/N1;
 TDIST[t,6]=SUMC(NSDOTR[1:N1,6].>=0)/N1;
 TDIST[t,7]=SUMC(NSDOTR[1:N1,7].>=0)/N1;
 TDIST[t,8]=SUMC(NSDOTR[1:N1,8].>=0)/N1;
 TDIST[t,9]=SUMC(NSDOTR[1:N1,9].>=0)/N1;
 TDIST[t,10]=SUMC(NSDOTR[N1+1:2*N1,1].>=0)/N1;   /* second type */
 TDIST[t,11]=SUMC(NSDOTR[N1+1:2*N1,2].>=0)/N1;
 TDIST[t,12]=SUMC(NSDOTR[N1+1:2*N1,3].>=0)/N1;
 TDIST[t,13]=SUMC(NSDOTR[N1+1:2*N1,4].>=0)/N1;
 TDIST[t,14]=SUMC(NSDOTR[N1+1:2*N1,5].>=0)/N1;
 TDIST[t,15]=SUMC(NSDOTR[N1+1:2*N1,6].>=0)/N1;
 TDIST[t,16]=SUMC(NSDOTR[N1+1:2*N1,7].>=0)/N1;
 TDIST[t,17]=SUMC(NSDOTR[N1+1:2*N1,8].>=0)/N1;
 TDIST[t,18]=SUMC(NSDOTR[N1+1:2*N1,9].>=0)/N1;
 TDIST[t,19]=SUMC(NSDOTR[2*N1+1:3*N1,1].>=0)/N1;   /* third type */
 TDIST[t,20]=SUMC(NSDOTR[2*N1+1:3*N1,2].>=0)/N1;
 TDIST[t,21]=SUMC(NSDOTR[2*N1+1:3*N1,3].>=0)/N1;
 TDIST[t,22]=SUMC(NSDOTR[2*N1+1:3*N1,4].>=0)/N1;
 TDIST[t,23]=SUMC(NSDOTR[2*N1+1:3*N1,5].>=0)/N1;
 TDIST[t,24]=SUMC(NSDOTR[2*N1+1:3*N1,6].>=0)/N1;
 TDIST[t,25]=SUMC(NSDOTR[2*N1+1:3*N1,7].>=0)/N1;
 TDIST[t,26]=SUMC(NSDOTR[2*N1+1:3*N1,8].>=0)/N1;
 TDIST[t,27]=SUMC(NSDOTR[2*N1+1:3*N1,9].>=0)/N1;
endo;
/*Get the last 100 averages of inventory stock distribution*/
SON1=ZEROS(3,9);
SON1[1,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,1:9]))'/AVR;
SON1[2,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,10:18]))'/AVR;
SON1[3,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,19:27]))'/AVR;
SON2=ZEROS(3,3);
SON2[1,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,1:3]))'/AVR;
SON2[2,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,4:6]))'/AVR;
SON2[3,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,7:9]))'/AVR;
```

```
SON3=ZEROS(3,3);
SON3[1,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,1:3]))'/AVR;
SON3[2,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,4:6]))'/AVR;
SON3[3,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,7:9]))'/AVR;
```

WITH IMITATION
FUNDAMENTAL EQUILIBRIUM CASE
Number of players of each type (select as even) 20.000000
The total number of players 60.000000
The discount factor (select between zero and one) 0.60000000
The probability of mistake 0.050000000
Probability of Imitation 1.0000000
Final date -- Number of times market opens 1000.0000
The utilities from consuming own type of good
u11 200.00000
u22 200.00000
u33 200.00000
The disutilities from production for each agent
D1 -100.00000
D2 -100.00000
D3 -100.00000
The inventory holding costs
c11 0.10000000
c12 10.000000
c13 25.000000
c21 0.10000000
c22 10.000000
c23 25.000000
c31 0.10000000
c32 10.000000
c33 25.000000
AVERAGES OF LAST 100.00000 MARKETS OPENNED.
TRADE OF

	1_1	1_2	1_3
TYPE I	0.85000000	0.24150000	0.15000000
TYPE II	0.47050000	1.0000000	0.0000000
TYPE III	0.58850000	0.0000000	1.0000000
	2_1	2_2	2_3
TYPE I	1.0000000	0.41950000	0.066500000
TYPE II	0.10000000	0.86450000	0.10000000
TYPE III	0.66700000	0.72300000	1.0000000
	3_1	3_2	3_3
TYPE I	1.0000000	0.63800000	0.58250000
TYPE II	1.0000000	0.45950000	0.59350000
TYPE III	1.0000000	0.90850000	0.60000000

CONSUMPTION

	GOOD1	GOOD2	GOOD3
TYPE I	1.0000000	0.0000000	0.50900000
TYPE II	0.0000000	1.0000000	0.0000000
TYPE III	0.0000000	0.35000000	1.0000000

STOK

	GOOD1	GOOD2	GOOD3
TYPE I	0.0035000000	0.98450000	0.012000000
TYPE II	0.45000000	0.0035000000	0.54650000
TYPE III	0.97350000	0.022000000	0.0045000000

```
WITHOUT IMITATION
FUNDAMENTAL EQUILIBRIUM CASE
Number of players of each type (select as even)        20.000000
The total number of players        60.000000
The discount factor (select between zero and one)       0.60000000
The probability of mistake      0.050000000
Probability of Imitation        0.0000000
Final date -- Number of times market opens        1000.0000
The utilities from consuming own type of good
u11         200.00000
u22         200.00000
u33         200.00000
The disutilities from production for each agent
D1        -100.00000
D2        -100.00000
D3        -100.00000
The inventory holding costs
c11        0.10000000
c12         10.000000
c13         25.000000
c21        0.10000000
c22         10.000000
c23         25.000000
c31        0.10000000
c32         10.000000
c33         25.000000
AVERAGES OF LAST        100.00000  MARKETS OPENNED.
TRADE OF
                  1_1             1_2             1_3
TYPE I         0.45000000      0.10000000      0.15000000
TYPE II        0.45000000       1.0000000       0.0000000
TYPE III       0.44950000     0.050000000       1.0000000
                  2_1             2_2             2_3
TYPE I          1.0000000      0.50000000      0.10000000
TYPE II        0.50000000      0.40000000      0.15000000
TYPE III       0.90000000      0.63750000       1.0000000
                  3_1             3_2             3_3
TYPE I          1.0000000      0.69100000      0.48900000
TYPE II         1.0000000      0.60150000      0.50500000
TYPE III       0.40000000      0.50000000      0.40000000
CONSUMPTION
                 GOOD1           GOOD2           GOOD3
TYPE I          1.0000000       0.0000000       0.0000000
TYPE II         0.0000000       1.0000000       0.0000000
TYPE III        0.0000000       0.0000000       1.0000000
STOK
                 GOOD1           GOOD2           GOOD3
TYPE I       0.0040000000      0.95150000     0.044500000
TYPE II        0.46800000     0.0020000000      0.53000000
TYPE III       0.94500000      0.049000000    0.0060000000
```

```
N1=20;            /* Number of players of each type (select as even)  */
N=3*N1;          /* The total number of players   */
ref=ones(N1,1)|(N1+1)*ones(N1,1)|(2*N1+1)*ones(N1,1);
                                  /* for referencing purposes */
beta=0.6;       /* The discount factor (select between zero and one) */
pr=0.05;        /* The probability of mistake */
pim=1;              /Imitation*/
pimit=pim*ones(1,N);  /* Homogeneous */
u11= 200;           /* The utilities from consuming own type of good  */
u22= 200;
u33= 200;
D1=-100;              /* The disutilities from production for each agent */
D2=-100;
D3=-100;
U1=(u11+D1)*ones(N1,1)~D1*ones(N1,1)~D1*ones(N1,1);
U2=D2*ones(N1,1)~(u22+D2)*ones(N1,1)~D1*ones(N1,1);
U3=D3*ones(N1,1)~D3*ones(N1,1)~(u11+D3)*ones(N1,1);
U=U1|U2|U3;      /* The N by 3 matrix of utilities */
c11=0.1;           /* The inventory holding costs  */
c12=10;           /* Type 1 agent, good 2        */
c13=10.1;
c21=0.1;
c22=10;
c23=10.1;
c31=0.1;
c32=10;
c33=10.1;
C1=c11*ones(N1,1)~c12*ones(N1,1)~c13*ones(N1,1);
C2=c21*ones(N1,1)~c22*ones(N1,1)~c23*ones(N1,1);
C3=c31*ones(N1,1)~c32*ones(N1,1)~c33*ones(N1,1);
C=C1|C2|C3;                    /* The N by 3 matrix of costs  */
/* Initial strengths will be random  */
SDOTR=rndn(N,9);      /* The strengths of players' do trade
                         classifiers (3x3=9 states) */
TDOTR=1*ones(N,9);      /* Times used */
SNOTR=rndn(N,9);      /* The strengths of players' don't trade
                         classifiers  */
TNOTR=1*ones(N,9);        /* Times used */
SDOCNS=rndn(N,3);      /* The strengths of players' do consume
                         classifiers (3 states) */
TDOCNS=1*ones(N,3);      /* Times used */
SNOCNS=rndn(N,3);     /* The strengths of players' don't consume
                         classifiers   */
TNOCNS=1*ones(N,3);      /* Times used */
STOK=2*ones(N1,1)|3*ones(N1,1)|ones(N1,1); /* The beginning pre-trade
                                              inventory vector  */
STOKC=ones(N,1);     /* Pre-consumption inventory held */
CNS=zeros(N,1);   /* Most recent consumption indicator. Consumed=1 Not=0*/
FINALD=1000;    /* Final date -- Number of times market opens */
AVR=100;       /* Averages of last markets which are openned */
SDIST=zeros(FINALD,9);     /* The stock distribution recorder */
SDIST[1,2]=1;              /* Starting distribution of inventories */
SDIST[1,6]=1;
SDIST[1,7]=1;
CDIST=zeros(FINALD,9);     /* The do consume percentage recorder */
```

```
TDIST=zeros(FINALD,27);    /* The do trade percentage recorder  */
/* Initialization part complete.  Now the market opens  */
/*  The main loop running over time index t.  */
t=0;
do while t<(FINALD);
 t=t+1;
 i=1;
 do while i<N+1;   /* i will be matched with a 'learned' agent  */
  rtype=rndu(1,1);      /* (SPECULATIVE EQM) */
  if rtype<1/3;
     ltype=1;         /* first type always comes with good 2 or good 3 */
     if rndu(1,1)<.707;
        lstock=2;
     else;
        lstock=3;
     endif;
  elseif rtype<2/3;
     ltype=2;         /* second type comes with either good 1 (41%) */
     if rndu(1,1)<.414;
        lstock=3;
     else;            /* or good 3 (59%)   */
        lstock=1;
     endif;
  else;
    ltype=3;   /* third type always comes with good 1 */
    lstock=1;
  endif;
  TRADE1=0;                          /* Initialization of trade  */
  TRADE2=0;                          /*  offer decisions.        */
  STATE1=3*(STOK[i]-1)+lstock;       /* Pre-trade */
  if ltype==1;        /* he will always carry good 2 (pre-trade) */
    if lstock==2;
      if (STOK[i]==1) or (STOK[i]==3);
                    /* say 'yes' only if partner comes with good 1 */
        TRADE2=1;
      endif;
    elseif lstock==3;
      if (STOK[i]==1);
                    /* say 'yes' only if partner comes with good 1 */
        TRADE2=1;
      endif;
    endif;
  endif;
  if ltype==2;        /* he will either carry (pre-trade) */
    if lstock==1;       /* good 1 */
      if (STOK[i]==2);
                  /* (says 'yes' only if partner comes with good 1) */
        TRADE2=1;
      endif;
    elseif lstock==3;   /* or good 3 */
      if ( (STOK[i]==1) or (STOK[i]==2) );
         TRADE2=1;       /* (says 'yes' to everything) */
      endif;
    endif;
  endif;
```

```
 if ltype==3;        /* he will always carry good 1 (pre-trade) */
   if lstock==1;
     if STOK[i]==3;   /* say 'yes' only if partner comes with good 3 */
       TRADE2=1;
     endif;
   endif;
 endif;
 /* First, the possibility of imitation  */
if rndu(1,1)<pimit[i];     /* If decides to look around */
     /* For this state, he emulates the weighted average
        of strengths among his types for this state  */
     SOCDT=SUMC(SDOTR[ref[i]:ref[i]+N1-1,STATE1].*
          (TDOTR[ref[i]:ref[i]+N1-1,STATE1]-1))/
          (SUMC(TDOTR[ref[i]:ref[i]+N1-1,STATE1]-1)+1);
     SOCNT=SUMC(SNOTR[ref[i]:ref[i]+N1-1,STATE1].*
          (TNOTR[ref[i]:ref[i]+N1-1,STATE1]-1))/
          (SUMC(TNOTR[ref[i]:ref[i]+N1-1,STATE1]-1)+1);
    STRADE1=SOCNT;     /*  Trade strength -- default: social no trade */
    if SOCDT>=SOCNT;   /* Decision according to social values */
       TRADE1=1;
       STRADE1=SOCDT;  /* Trade strength -- change: social do trade */
    endif;
 else;  /* No imitation -- own strengths */
    STRADE1=SNOTR[i,STATE1];     /*  Trade strength  -- default: no trade */
 /*  Now trade offer decisons will be made. Note the bias towards
     trade in case of equality of strengths */
   if SDOTR[i,STATE1]>= SNOTR[i,STATE1];
      TRADE1=1;                      /* First agent decides */
      STRADE1=SDOTR[i,STATE1];     /* Trade strength -- change : do trade */
   endif;
endif;
 /* Now the possibility of non-deliberate decision */
if rndu(1,1)<pr;  /* Random choice: Agent i */
   if rndu(1,1)<0.5;
     TRADE1=1;         /* 50 percent trade */
     STRADE1=SDOTR[i,STATE1];
   else;
     TRADE1=0;         /* 50 percent no trade */
     STRADE1=SNOTR[i,STATE1];
   endif;
endif;
 /* Now strength update of consumption classifiers will be made */
 if t>1;   /* Start update at time 2, since no consumption
              decision is made for t=1 yet. */
           /* First agent first  */
  if CNS[i]==1;
    SDOCNS[i,STOKC[i]]=SDOCNS[i,STOKC[i]]
    -(1/(TDOCNS[i,STOKC[i]]+1))
    *(SDOCNS[i,STOKC[i]]
    -U[i,STOKC[i]]
    +C[i,STOK[i]]
    -beta*STRADE1);
    TDOCNS[i,STOKC[i]]=TDOCNS[i,STOKC[i]]+1;
  else;
    SNOCNS[i,STOKC[i]]=SNOCNS[i,STOKC[i]]
```

```
   -(1/(TNOCNS[i,STOKC[i]]+1))
   *(SNOCNS[i,STOKC[i]]
   +C[i,STOK[i]]
   -beta*STRADE1);
   TNOCNS[i,STOKC[i]]=TNOCNS[i,STOKC[i]]+1;
 endif;
endif;
/* Now back to the trade round */
TRADE=TRADE1*TRADE2;
if TRADE==1;
 STOKC[i]=1stock;          /* If exchange takes place */
else;                              /* else, prev. stocks   */
 STOKC[i]=STOK[i];      /* are kept.           */
endif;
/* Now the consumption decision will be made  */
/* (Note the bias towards consuming in case of equal strengths */
/* First, the possibility of imitation (consumption strengths) */
if rndu(1,1)<pimit[i];     /* If decides to look around */
/* For this state (commodity), he acts according to the weighted average
       of strengths among his types for this state  */
  GOOD1=STOKC[i];
  SOCDC=SDOCNS[ref[i]:ref[i]+N1-1,GOOD1]'*
        (TDOCNS[ref[i]:ref[i]+N1-1,GOOD1]-1)/
        (SUMC(TDOCNS[ref[i]:ref[i]+N1-1,GOOD1]-1)+1);
  SOCNC=SNOCNS[ref[i]:ref[i]+N1-1,GOOD1]'*
        (TNOCNS[ref[i]:ref[i]+N1-1,GOOD1]-1)/
        (SUMC(TNOCNS[ref[i]:ref[i]+N1-1,GOOD1]-1)+1);
   SCONS1=SOCNC;     /*  Consumption strength -- default: social no consume */
   CNS[i]=0;
   STOK[i]=STOKC[i];           /* Same stock  */
   if SOCDC>=SOCNC; /* Decision according to social values */
      SCONS1=SOCDC; /* Consumption strength -- change: social no consume */
      CNS[i]=1;
      if i<=20;               /* first type  */
      STOK[i]=2;       /* type 1 produces 2 */
      elseif i<=40;           /* second type */
        STOK[i]=3;   /* type 2 produces 3 */
      else;                   /* third type  */
        STOK[i]=1;   /* type 3 produces 1 */
      endif;
  endif;
else;  /* no imitation */
 if SDOCNS[i,STOKC[i]]>=SNOCNS[i,STOKC[i]];
    CNS[i]=1;                          /* do consume  */
    SCONS1=SDOCNS[i,STOKC[i]];              /* its value (strength) */
    if i<=20;               /* first type  */
    STOK[i]=2;       /* type 1 produces 2 */
     elseif i<=40;          /* second type */
       STOK[i]=3;   /* type 2 produces 3 */
     else;                  /* third type  */
       STOK[i]=1;   /* type 3 produces 1 */
    endif;
 else;
    CNS[i]=0;
    STOK[i]=STOKC[i];           /* Same stock  */
```

```
    SCONS1=SNOCNS[i,STOKC[i]];  /* its value (strength) */
 endif;
endif;
 /* Now the possibility of non-deliberate decision */
if rndu(1,1)<pr;  /* Random choice: Agent i */
   if rndu(1,1)<0.5;
     CNS[i]=0;                          /* don't consume  */
     SCONS1=SNOCNS[i,STOKC[i]];  /* its value (strength) */
     STOK[i]=STOKC[i];           /* Same stock  */
   else;
     CNS[i]=1;                          /* do consume */
     SCONS1=SDOCNS[i,STOKC[i]];  /* its value (strength) */
     if i<=20;                /* first type  */
     STOK[i]=2;       /* type 1 produces 2 */
      elseif i<=40;           /* second type */
       STOK[i]=3;   /* type 2 produces 3 */
      else;                    /* third type  */
       STOK[i]=1;   /* type 3 produces 1 */
     endif;
   endif;
endif;
 /* Now, update of trade classifier strengths  */
 if TRADE1==1;              /* Agent i */
   SDOTR[i,STATE1]=SDOTR[i,STATE1]
        -(1/(TDOTR[i,STATE1]+1))*(SDOTR[i,STATE1]-SCONS1);
   TDOTR[i,STATE1]=TDOTR[i,STATE1]+1;
  else;
   SNOTR[i,STATE1]=SNOTR[i,STATE1]
      -(1/(TNOTR[i,STATE1]+1))*(SNOTR[i,STATE1]-SCONS1);
   TNOTR[i,STATE1]=TNOTR[i,STATE1]+1;
 endif;
 i=i+1;   /* Next agent...  */
endo;
/* Now records for reporting purposes */
 /* Stock distribution */
SDIST[t,1]=SUMC(STOK[1:N1].==1)/N1;  /* first type */
SDIST[t,2]=SUMC(STOK[1:N1].==2)/N1;
SDIST[t,3]=SUMC(STOK[1:N1].==3)/N1;
SDIST[t,4]=SUMC(STOK[N1+1:2*N1].==1)/N1;  /* second type */
SDIST[t,5]=SUMC(STOK[N1+1:2*N1].==2)/N1;
SDIST[t,6]=SUMC(STOK[N1+1:2*N1].==3)/N1;
SDIST[t,7]=SUMC(STOK[2*N1+1:3*N1].==1)/N1;  /* third type */
SDIST[t,8]=SUMC(STOK[2*N1+1:3*N1].==2)/N1;
SDIST[t,9]=SUMC(STOK[2*N1+1:3*N1].==3)/N1;
 /* Do consume decision percentage over goods and types */
 NSDOCNS=SDOCNS-SNOCNS;       /* Net strengths of do consume classifiers */
 CDIST[t,1]=SUMC(NSDOCNS[1:N1,1].>=0)/N1;         /* first type */
 CDIST[t,2]=SUMC(NSDOCNS[1:N1,2].>=0)/N1;
 CDIST[t,3]=SUMC(NSDOCNS[1:N1,3].>=0)/N1;
 CDIST[t,4]=SUMC(NSDOCNS[N1+1:2*N1,1].>=0)/N1;     /* second type */
 CDIST[t,5]=SUMC(NSDOCNS[N1+1:2*N1,2].>=0)/N1;
 CDIST[t,6]=SUMC(NSDOCNS[N1+1:2*N1,3].>=0)/N1;
 CDIST[t,7]=SUMC(NSDOCNS[2*N1+1:3*N1,1].>=0)/N1;   /* third type */
 CDIST[t,8]=SUMC(NSDOCNS[2*N1+1:3*N1,2].>=0)/N1;
 CDIST[t,9]=SUMC(NSDOCNS[2*N1+1:3*N1,3].>=0)/N1;
```

```
  /* Do trade decision percentage over goods and types */
  NSDOTR=SDOTR-SNOTR;       /* Net strengths of do consume classifiers */
  TDIST[t,1]=SUMC(NSDOTR[1:N1,1].>=0)/N1;          /* first type */
  TDIST[t,2]=SUMC(NSDOTR[1:N1,2].>=0)/N1;
  TDIST[t,3]=SUMC(NSDOTR[1:N1,3].>=0)/N1;
  TDIST[t,4]=SUMC(NSDOTR[1:N1,4].>=0)/N1;
  TDIST[t,5]=SUMC(NSDOTR[1:N1,5].>=0)/N1;
  TDIST[t,6]=SUMC(NSDOTR[1:N1,6].>=0)/N1;
  TDIST[t,7]=SUMC(NSDOTR[1:N1,7].>=0)/N1;
  TDIST[t,8]=SUMC(NSDOTR[1:N1,8].>=0)/N1;
  TDIST[t,9]=SUMC(NSDOTR[1:N1,9].>=0)/N1;
  TDIST[t,10]=SUMC(NSDOTR[N1+1:2*N1,1].>=0)/N1;   /* second type */
  TDIST[t,11]=SUMC(NSDOTR[N1+1:2*N1,2].>=0)/N1;
  TDIST[t,12]=SUMC(NSDOTR[N1+1:2*N1,3].>=0)/N1;
  TDIST[t,13]=SUMC(NSDOTR[N1+1:2*N1,4].>=0)/N1;
  TDIST[t,14]=SUMC(NSDOTR[N1+1:2*N1,5].>=0)/N1;
  TDIST[t,15]=SUMC(NSDOTR[N1+1:2*N1,6].>=0)/N1;
  TDIST[t,16]=SUMC(NSDOTR[N1+1:2*N1,7].>=0)/N1;
  TDIST[t,17]=SUMC(NSDOTR[N1+1:2*N1,8].>=0)/N1;
  TDIST[t,18]=SUMC(NSDOTR[N1+1:2*N1,9].>=0)/N1;
  TDIST[t,19]=SUMC(NSDOTR[2*N1+1:3*N1,1].>=0)/N1;   /* third type */
  TDIST[t,20]=SUMC(NSDOTR[2*N1+1:3*N1,2].>=0)/N1;
  TDIST[t,21]=SUMC(NSDOTR[2*N1+1:3*N1,3].>=0)/N1;
  TDIST[t,22]=SUMC(NSDOTR[2*N1+1:3*N1,4].>=0)/N1;
  TDIST[t,23]=SUMC(NSDOTR[2*N1+1:3*N1,5].>=0)/N1;
  TDIST[t,24]=SUMC(NSDOTR[2*N1+1:3*N1,6].>=0)/N1;
  TDIST[t,25]=SUMC(NSDOTR[2*N1+1:3*N1,7].>=0)/N1;
  TDIST[t,26]=SUMC(NSDOTR[2*N1+1:3*N1,8].>=0)/N1;
  TDIST[t,27]=SUMC(NSDOTR[2*N1+1:3*N1,9].>=0)/N1;
endo;
/*Get the last 100 averages of inventory stock distribution*/
SON1=ZEROS(3,9);
SON1[1,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,1:9]))'/AVR;
SON1[2,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,10:18]))'/AVR;
SON1[3,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,19:27]))'/AVR;
SON2=ZEROS(3,3);
SON2[1,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,1:3]))'/AVR;
SON2[2,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,4:6]))'/AVR;
SON2[3,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,7:9]))'/AVR;
SON3=ZEROS(3,3);
SON3[1,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,1:3]))'/AVR;
SON3[2,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,4:6]))'/AVR;
SON3[3,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,7:9]))'/AVR;
```

WITH IMITATION
SPECULATIVE EQUILIBRIUM CASE
Number of players of each type (select as even) 20.000000
The total number of players 60.000000
The discount factor (select between zero and one) 0.60000000
The probability of mistake 0.050000000
Probability of Imitation 1.0000000
Final date -- Number of times market opens 1000.0000
The utilities from consuming own type of good
u11 200.00000
u22 200.00000
u33 200.00000
The disutilities from production for each agent
D1 -100.00000
D2 -100.00000
D3 -100.00000
The inventory holding costs
c11 0.10000000
c12 10.000000
c13 10.100000
c21 0.10000000
c22 10.000000
c23 10.100000
c31 0.10000000
c32 10.000000
c33 10.100000
AVERAGES OF LAST 100.00000 MARKETS OPENNED.
TRADE OF

	1_1	1_2	1_3
TYPE I	1.0000000	0.050000000	0.30000000
TYPE II	0.59200000	1.0000000	0.0000000
TYPE III	0.49400000	0.0000000	1.0000000
	2_1	2_2	2_3
TYPE I	1.0000000	0.23800000	1.0000000
TYPE II	0.90000000	0.95000000	0.67650000
TYPE III	0.90900000	0.20000000	1.0000000
	3_1	3_2	3_3
TYPE I	1.0000000	0.049000000	0.19500000
TYPE II	1.0000000	1.0000000	0.49550000
TYPE III	0.15000000	0.10000000	0.91600000

CONSUMPTION

	GOOD1	GOOD2	GOOD3
TYPE I	1.0000000	0.0000000	0.0000000
TYPE II	0.0000000	1.0000000	0.0000000
TYPE III	0.0000000	0.30000000	1.0000000

STOK

	GOOD1	GOOD2	GOOD3
TYPE I	0.0045000000	0.69350000	0.30200000
TYPE II	0.53750000	0.0085000000	0.45400000
TYPE III	0.96700000	0.028500000	0.0045000000

```
WITHOUT IMITATION
SPECULATIVE EQUILIBRIUM CASE
Number of players of each type (select as even)        20.000000
The total number of players        60.000000
The discount factor (select between zero and one)       0.60000000
The probability of mistake      0.050000000
Probability of Imitation        0.0000000
Final date -- Number of times market opens        1000.0000
The utilities from consuming own type of good
u11         200.00000
u22         200.00000
u33         200.00000
The disutilities from production for each agent
D1        -100.00000
D2        -100.00000
D3        -100.00000
The inventory holding costs
c11        0.10000000
c12         10.000000
c13         10.100000
c21        0.10000000
c22         10.000000
c23         10.100000
c31        0.10000000
c32         10.000000
c33         10.100000
AVERAGES OF LAST        100.00000  MARKETS OPENNED.
TRADE OF
                  1_1             1_2             1_3
TYPE I         0.75000000      0.050000000      0.10000000
TYPE II        0.40050000        1.0000000      0.050000000
TYPE III       0.50000000        0.0000000        1.0000000
                  2_1             2_2             2_3
TYPE I          1.0000000       0.55000000      0.90000000
TYPE II        0.48950000       0.25000000      0.32300000
TYPE III       0.80000000       0.49100000        1.0000000
                  3_1             3_2             3_3
TYPE I          1.0000000       0.27350000      0.70000000
TYPE II        0.95000000        1.0000000      0.61000000
TYPE III       0.55000000       0.25000000      0.40000000
CONSUMPTION
                 GOOD1           GOOD2           GOOD3
TYPE I          1.0000000       0.0000000       0.0000000
TYPE II         0.0000000       1.0000000       0.0000000
TYPE III        0.0000000       0.0000000       1.0000000
STOK
                 GOOD1           GOOD2           GOOD3
TYPE I       0.0050000000       0.75700000      0.23800000
TYPE II        0.48750000     0.0060000000      0.50650000
TYPE III       0.97900000      0.018000000     0.0030000000
```

APPENDIX C

Programs and Results of Kiyotaki and Wright Part:

When we look at the table of outputs for trade and consumption, we get the proportion of players who decide to do action at the last time period. In the trade table, i-j means trade for good i in return to good j. For example, when we look at the row for type 1 player and column for 2-3, we get the proportion of type 1 players who decide to do trade good 2 in return to good 3. In the consumption table, the rows give us the type of players and columns give us the type of goods. In the stok table, the distribution of holding goods for each type of players is given.

1. Computer Program: Fundamental Equilibrium
2. Output: Fundamental Equilibrium and Full Imitation Case
3. Output: Fundamental Equilibrium and No Imitation Case
4. Output: Speculative Equilibrium and Full Imitation Case
5. Output: Speculative Equilibrium and No Imitation Case

```
N1=20;            /* Number of players of each type (select as even)  */
N=3*N1;          /* The total number of players   */
ref=ones(N1,1)|(N1+1)*ones(N1,1)|(2*N1+1)*ones(N1,1);
                                /* for referencing purposes */
beta=0.6;       /* The discount factor (select between zero and one) */
pr=0.05;        /* The probability of mistake */
pim=1;
pimit=pim*ones(1,N);  /* Homogeneous */
u11= 200;           /* The utilities from consuming own type of good  */
u22= 200;
u33= 200;
D1=-100;              /* The disutilities from production for each agent */
D2=-100;
D3=-100;
U1=(u11+D1)*ones(N1,1)~D1*ones(N1,1)~D1*ones(N1,1);
U2=D2*ones(N1,1)~(u22+D2)*ones(N1,1)~D1*ones(N1,1);
U3=D3*ones(N1,1)~D3*ones(N1,1)~(u11+D3)*ones(N1,1);
U=U1|U2|U3;      /* The N by 3 matrix of utilities */
c11=0.1;           /* The inventory holding costs  */
c12=10;           /* Type 1 agent, good 2        */
c13=25;
c21=0.1;
c22=10;
c23=25;
c31=0.1;
c32=10;
c33=25;
C1=c11*ones(N1,1)~c12*ones(N1,1)~c13*ones(N1,1);
C2=c21*ones(N1,1)~c22*ones(N1,1)~c23*ones(N1,1);
C3=c31*ones(N1,1)~c32*ones(N1,1)~c33*ones(N1,1);
C=C1|C2|C3;                    /* The N by 3 matrix of costs  */
/* Initial strengths will be random  */
SDOTR=rndn(N,9);      /* The strengths of players' do trade
                         classifiers (3x3=9 states) */
TDOTR=1*ones(N,9);      /* Times used */
SNOTR=rndn(N,9);      /* The strengths of players' don't trade
                         classifiers  */
TNOTR=1*ones(N,9);        /* Times used */
SDOCNS=rndn(N,3);      /* The strengths of players' do consume
                         classifiers (3 states) */
TDOCNS=1*ones(N,3);      /* Times used */
SNOCNS=rndn(N,3);     /* The strengths of players' don't consume
                         classifiers   */
TNOCNS=1*ones(N,3);      /* Times used */
STOK=2*ones(N1,1)|3*ones(N1,1)|ones(N1,1); /* The beginning pre-trade
                                              inventory vector  */
STOKC=ones(N,1);     /* Pre-consumption inventory held */
CNS=ones(N,1);   /* Most recent consumption indicator. Consumed=1 Not=0*/
FINALD=1000;    /* Final date -- Number of times market opens */
AVR=100;
SDIST=zeros(FINALD,9);     /* The stock distribution recorder */
SDIST[1,2]=1;              /* Starting distribution */
SDIST[1,6]=1;
SDIST[1,7]=1;
CDIST=zeros(FINALD,9);     /* The do consume percentage recorder */
```

```
TDIST=zeros(FINALD,27);    /* The do trade percentage recorder  */
/* Initialization part complete.  Now the market opens  */
/*  The main loop running over time index t.  */
t=0;
do while t<FINALD;
 t=t+1;
/* First thing is the pairwise matching of the players.
     This will be done by first generating independent uniform random
numbers between zero and one for each agent.  This will determine the
rank of the agent. The highest valued agent will be assigned rank one,
the next high value rank 2, etc.The ranks will be stored in the vector
AGENT in the prescribed order.  Then following the order in AGENT,
the neighboring players will be matched.  */

 AGENT=ones(N,1);   /* Initialization  */
 R=RNDU(N,1);     /* Vector of random values for indices of players  */
 i=1;
 do while i<N+1;
        AGENT[i]=maxindc(R);
        R[maxindc(R)]=-1;   /* A number smaller than zero */
        i=i+1;
 endo;
 i=1;
 do while i<N;   /* AGENT[i] and AGENT[i+1] are matched  */
  TRADE1=1;                          /* Initialization of trade  */
  TRADE2=1;                          /*  offer decisions.        */
  STATE1=3*(STOK[AGENT[i]]-1)+STOK[AGENT[i+1]];  /* Pre-trade */
  STATE2=3*(STOK[AGENT[i+1]]-1)+STOK[AGENT[i]];  /* states.    */
  /* First, the possibility of imitation  */
  /* First agent first  */
  if rndu(1,1)<pimit[agent[i]];     /* If decides to look around */
      /* For this state, he adopts the weighted average
         of strengths among his types for this state  */
    SOCDT=SUMC(SDOTR[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,STATE1].*
          (TDOTR[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,STATE1]-1))/
          (SUMC(TDOTR[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,STATE1]-1)+1);
    SOCNT=SUMC(SNOTR[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,STATE1].*
           (TNOTR[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,STATE1]-1))/
           (SUMC(TNOTR[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,STATE1]-1)+1);
    STRADE1=SOCDT;
    if SOCDT<SOCNT;
         STRADE1=SOCNT;
         TRADE1=0;
    endif;
  endif;
  /* Second agent next  */
  if rndu(1,1)<pimit[agent[i+1]];     /* If decides to look around */
      /*  Similar update for the second agent */
    SOCDT=SUMC(SDOTR[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,STATE2].*
           (TDOTR[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,STATE2]-1))/
           (SUMC(TDOTR[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,STATE2]-1)+1);
    SOCNT=SUMC(SNOTR[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,STATE2].*
           (TNOTR[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,STATE2]-1))/
           (SUMC(TNOTR[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,STATE2]-1)+1);
    STRADE2=SOCDT;
```

```
  if SOCDT<SOCNT;
       STRADE2=SOCNT;
       TRADE2=0;
  endif;
 else;
 /* Preperations for trade  */
 STRADE1=SDOTR[AGENT[i],STATE1];     /*  Trade  */
 STRADE2=SDOTR[AGENT[i+1],STATE2];   /*  strengths for both players*/
 /*  Now trade offer decisons will be made. Note the bias towards
     trade in case of equality of strengths */
 if SDOTR[AGENT[i],STATE1]< SNOTR[AGENT[i],STATE1];
   TRADE1=0;                        /* First agent decides */
   STRADE1=SNOTR[AGENT[i],STATE1];
endif;
 if SDOTR[AGENT[i+1],STATE2]< SNOTR[AGENT[i+1],STATE2];
   TRADE2=0;                       /* Second agent decides  */
   STRADE2=SNOTR[AGENT[i+1],STATE2];
 endif;
endif;
 /* Now the possibility of non-deliberate decision */
 if rndu(1,1)<pr;  /* Random choice: Agent i */
   if rndu(1,1)<0.5;
     TRADE1=1;         /* 50 percent trade */
     STRADE1=SDOTR[AGENT[i],STATE1];
   else;
     TRADE1=0;         /* 50 percent no trade */
     STRADE1=SNOTR[AGENT[i],STATE1];
   endif;
 endif;
 if rndu(1,1)<pr;  /* Random choice: Agent i+1 */
   if rndu(1,1)<0.5;
     TRADE2=1;         /* 50 percent trade */
     STRADE2=SDOTR[AGENT[i+1],STATE2];
   else;
     TRADE2=0;         /* 50 percent no trade */
     STRADE2=SNOTR[AGENT[i+1],STATE2];
   endif;
 endif;
 /* Now strength update of consumption classifiers will be made */
 if t>1;   /* Start update at time 2, since no consumption
              decision is made for t=1 yet. */
           /* First agent first  */
  if CNS[AGENT[i]]==1;
    SDOCNS[AGENT[i],STOKC[AGENT[i]]]=SDOCNS[AGENT[i],STOKC[AGENT[i]]]
    -(1/(TDOCNS[AGENT[i],STOKC[AGENT[i]]]+1))
    *(SDOCNS[AGENT[i],STOKC[AGENT[i]]]
    -U[AGENT[i],STOKC[AGENT[i]]]
    +C[AGENT[i],STOK[AGENT[i]]]
    -beta*STRADE1);
    TDOCNS[AGENT[i],STOKC[AGENT[i]]]=TDOCNS[AGENT[i],STOKC[AGENT[i]]]+1;
  else;
    SNOCNS[AGENT[i],STOKC[AGENT[i]]]=SNOCNS[AGENT[i],STOKC[AGENT[i]]]
    -(1/(TNOCNS[AGENT[i],STOKC[AGENT[i]]]+1))
    *(SNOCNS[AGENT[i],STOKC[AGENT[i]]]
    +C[AGENT[i],STOK[AGENT[i]]]
```

```
    -beta*STRADE1);
    TNOCNS[AGENT[i],STOKC[AGENT[i]]]=TNOCNS[AGENT[i],STOKC[AGENT[i]]]+1;
   endif;
          /* Second agent next */
   if CNS[AGENT[i+1]]==1;
    SDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]=SDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]
     -(1/(TDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]+1))
     *(SDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]
     -U[AGENT[i+1],STOKC[AGENT[i+1]]]
     +C[AGENT[i+1],STOK[AGENT[i+1]]]
     -beta*STRADE2);
     TDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]
     =TDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]+1;
   else;
    SNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]=SNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]
     -(1/(TNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]+1))
     *(SNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]
     +C[AGENT[i+1],STOK[AGENT[i+1]]]-beta*STRADE2);
     TNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]
     =TNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]+1;
   endif;
  endif;
   /* Now back to the trade round */
  TRADE=TRADE1*TRADE2;
  if TRADE==1;
   STOKC[AGENT[i]]=STOK[AGENT[i+1]]; /* If exchange takes place */
   STOKC[AGENT[i+1]]=STOK[AGENT[i]];
  else;                              /* else, prev. stocks   */
   STOKC[AGENT[i]]=STOK[AGENT[i]];  /* are kept.           */
   STOKC[AGENT[i+1]]=STOK[AGENT[i+1]];
  endif;
  /* Now the consumption decision will be made  */
  /* (Note the bias towards consuming in case of equal strengths */
  /* First, the possibility of imitation (consumption strengths) */
  /* First agent first  */
 if rndu(1,1)<pimit[agent[i]];     /* If decides to look around */
    /* For this state (commodity), he acts according to the weighted average
         of strengths among his types for this state  */
    GOOD1=STOKC[AGENT[i]];
    GOOD2=STOKC[AGENT[i+1]];
    SOCDC=SUMC(SDOCNS[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,GOOD1].*
           (TDOCNS[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,GOOD1]-1))/
           (SUMC(TDOCNS[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,GOOD1]-1)+1);
    SOCNC=SUMC(SNOCNS[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,GOOD1].*
           (TNOCNS[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,GOOD1]-1))/
           (SUMC(TNOCNS[ref[AGENT[i]]:ref[AGENT[i]]+N1-1,GOOD1]-1)+1);
  if SOCDC<SOCNC;
     CNS[AGENT[i]]=0;                        /* don't consume  */
     SCONS1=SOCNC;  /* its value (social strength) */
  else;
     CNS[AGENT[i]]=1;                        /* do consume  */
     SCONS1=SOCDC;  /* its value (social strength) */
  endif;
else;               /* Consider own strengths --not social ones */
  if SDOCNS[AGENT[i],STOKC[AGENT[i]]]<SNOCNS[AGENT[i],STOKC[AGENT[i]]];
```

```
    CNS[AGENT[i]]=0;                          /* don't consume  */
    SCONS1=SNOCNS[AGENT[i],STOKC[AGENT[i]]];  /* its value (strength) */
   else;
    CNS[AGENT[i]]=1;                          /* do consume */
    SCONS1=SDOCNS[AGENT[i],STOKC[AGENT[i]]];  /* its value (strength) */
 endif;
endif;
 /* Second agent next  */
if rndu(1,1)<pimit[agent[i+1]];     /* If decides to look around */
     /*  Similar update for the second agent */
   SOCDC=SUMC(SDOCNS[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,GOOD2].*
         (TDOCNS[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,GOOD2]-1))/
         (SUMC(TDOCNS[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,GOOD2]-1)+1);
   SOCNC=SUMC(SNOCNS[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,GOOD2].*
         (TNOCNS[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,GOOD2]-1))/
         (SUMC(TNOCNS[ref[AGENT[i+1]]:ref[AGENT[i+1]]+N1-1,GOOD2]-1)+1);
 if SOCDC<SOCNC;
    CNS[AGENT[i+1]]=0;                        /* don't consume  */
    SCONS2=SOCNC;   /* its value (social strength) */
 else;
    CNS[AGENT[i+1]]=1;                        /* do consume  */
    SCONS2=SOCDC;   /* its value (social strength) */
 endif;
else;
 if SDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]<SNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]];
    CNS[AGENT[i+1]]=0;                          /* don't consume  */
    SCONS2=SNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]; /* its value (strength) */
   else;
    CNS[AGENT[i+1]]=1;                          /* do consume */
    SCONS2=SDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]]; /* its value (strength) */
 endif;
endif;
if CNS[AGENT[i]]==1;            /* If decided to consume */
    if AGENT[i]<=N1;               /* first type  */
    STOK[AGENT[i]]=2;       /* type 1 produces 2 */
     elseif AGENT[i]<=2*N1;          /* second type */
       STOK[AGENT[i]]=3;   /* type 2 produces 3 */
     else;                  /* third type  */
       STOK[AGENT[i]]=1;   /* type 3 produces 1 */
    endif;
else;                         /* If decided not to consume */
    STOK[AGENT[i]]=STOKC[AGENT[i]];          /* Same stock  */
endif;
if CNS[AGENT[i+1]]==1;            /* If decided to consume */
    if AGENT[i+1]<=N1;               /* first type  */
    STOK[AGENT[i+1]]=2;       /* type 1 produces 2 */
     elseif AGENT[i+1]<=2*N1;          /* second type */
       STOK[AGENT[i+1]]=3;   /* type 2 produces 3 */
     else;                  /* third type  */
       STOK[AGENT[i+1]]=1;   /* type 3 produces 1 */
    endif;
else;                         /* If decided not to consume */
    STOK[AGENT[i+1]]=STOKC[AGENT[i+1]];          /* Same stock  */
endif;
/* Now the possibility of non-deliberate decision */
```

```
if rndu(1,1)<pr;  /* Random choice: Agent i */
  if rndu(1,1)<0.5;
   CNS[AGENT[i]]=0;                          /* don't consume  */
   SCONS1=SNOCNS[AGENT[i],STOKC[AGENT[i]]];  /* its value (strength) */
   STOK[AGENT[i]]=STOKC[AGENT[i]];           /* Same stock  */
  else;
   CNS[AGENT[i]]=1;                          /* do consume */
   SCONS1=SDOCNS[AGENT[i],STOKC[AGENT[i]]];  /* its value (strength) */
   if AGENT[i]<=N1;               /* first type  */
   STOK[AGENT[i]]=2;       /* type 1 produces 2 */
    elseif AGENT[i]<=2*N1;          /* second type */
      STOK[AGENT[i]]=3;   /* type 2 produces 3 */
    else;                  /* third type  */
      STOK[AGENT[i]]=1;   /* type 3 produces 1 */
   endif;
  endif;
endif;
if rndu(1,1)<pr;  /* Random choice: Agent i+1 */
  if rndu(1,1)<0.5;
   CNS[AGENT[i+1]]=0;                            /* don't consume  */
   SCONS2=SNOCNS[AGENT[i+1],STOKC[AGENT[i+1]]];  /* its value (strength) */
   STOK[AGENT[i+1]]=STOKC[AGENT[i+1]];           /* Same stock  */
  else;
   CNS[AGENT[i+1]]=1;                            /* do consume */
   SCONS2=SDOCNS[AGENT[i+1],STOKC[AGENT[i+1]]];  /* its value (strength) */
   if AGENT[i+1]<=N1;               /* first type  */
   STOK[AGENT[i+1]]=2;       /* type 1 produces 2 */
    elseif AGENT[i+1]<=2*N1;          /* second type */
      STOK[AGENT[i+1]]=3;   /* type 2 produces 3 */
    else;                  /* third type  */
      STOK[AGENT[i+1]]=1;   /* type 3 produces 1 */
   endif;
  endif;
endif;
/* Now, update of trade classifier strengths  */
if TRADE1==1;             /* Agent i */
  SDOTR[AGENT[i],STATE1]=SDOTR[AGENT[i],STATE1]
       -(1/(TDOTR[AGENT[i],STATE1]+1))*(SDOTR[AGENT[i],STATE1]-SCONS1);
  TDOTR[AGENT[i],STATE1]=TDOTR[AGENT[i],STATE1]+1;
 else;
  SNOTR[AGENT[i],STATE1]=SNOTR[AGENT[i],STATE1]
     -(1/(TNOTR[AGENT[i],STATE1]+1))*(SNOTR[AGENT[i],STATE1]-SCONS1);
  TNOTR[AGENT[i],STATE1]=TNOTR[AGENT[i],STATE1]+1;
endif;
if TRADE2==1;             /* Agent i+1 */
  SDOTR[AGENT[i+1],STATE2]=SDOTR[AGENT[i+1],STATE2]
     -(1/(TDOTR[AGENT[i+1],STATE2]+1))*(SDOTR[AGENT[i+1],STATE2]-SCONS2);
  TDOTR[AGENT[i+1],STATE2]=TDOTR[AGENT[i+1],STATE2]+1;
 else;
  SNOTR[AGENT[i+1],STATE2]=SNOTR[AGENT[i+1],STATE2]
     -(1/(TNOTR[AGENT[i+1],STATE2]+1))*(SNOTR[AGENT[i+1],STATE2]-SCONS2);
  TNOTR[AGENT[i+1],STATE2]=TNOTR[AGENT[i+1],STATE2]+1;
endif;
i=i+2;   /* Skip one agent since that is already processed.  */
endo;
```

```
/* Now records for reporting purposes */
 /* Stock distribution */
SDIST[t,1]=SUMC(STOK[1:N1].==1)/N1;  /* first type */
SDIST[t,2]=SUMC(STOK[1:N1].==2)/N1;
SDIST[t,3]=SUMC(STOK[1:N1].==3)/N1;
SDIST[t,4]=SUMC(STOK[N1+1:2*N1].==1)/N1;  /* second type */
SDIST[t,5]=SUMC(STOK[N1+1:2*N1].==2)/N1;
SDIST[t,6]=SUMC(STOK[N1+1:2*N1].==3)/N1;
SDIST[t,7]=SUMC(STOK[2*N1+1:3*N1].==1)/N1;  /* third type */
SDIST[t,8]=SUMC(STOK[2*N1+1:3*N1].==2)/N1;
SDIST[t,9]=SUMC(STOK[2*N1+1:3*N1].==3)/N1;
 /* Do consume decision percentage over goods and types */
 NSDOCNS=SDOCNS-SNOCNS;       /* Net strengths of do consume classifiers */
 CDIST[t,1]=SUMC(NSDOCNS[1:N1,1].>=0)/N1;         /* first type */
 CDIST[t,2]=SUMC(NSDOCNS[1:N1,2].>=0)/N1;
 CDIST[t,3]=SUMC(NSDOCNS[1:N1,3].>=0)/N1;
 CDIST[t,4]=SUMC(NSDOCNS[N1+1:2*N1,1].>=0)/N1;     /* second type */
 CDIST[t,5]=SUMC(NSDOCNS[N1+1:2*N1,2].>=0)/N1;
 CDIST[t,6]=SUMC(NSDOCNS[N1+1:2*N1,3].>=0)/N1;
 CDIST[t,7]=SUMC(NSDOCNS[2*N1+1:3*N1,1].>=0)/N1;   /* third type */
 CDIST[t,8]=SUMC(NSDOCNS[2*N1+1:3*N1,2].>=0)/N1;
 CDIST[t,9]=SUMC(NSDOCNS[2*N1+1:3*N1,3].>=0)/N1;
 /* Do trade decision percentage over goods and types */
 NSDOTR=SDOTR-SNOTR;       /* Net strengths of do consume classifiers */
 TDIST[t,1]=SUMC(NSDOTR[1:N1,1].>=0)/N1;          /* first type */
 TDIST[t,2]=SUMC(NSDOTR[1:N1,2].>=0)/N1;
 TDIST[t,3]=SUMC(NSDOTR[1:N1,3].>=0)/N1;
 TDIST[t,4]=SUMC(NSDOTR[1:N1,4].>=0)/N1;
 TDIST[t,5]=SUMC(NSDOTR[1:N1,5].>=0)/N1;
 TDIST[t,6]=SUMC(NSDOTR[1:N1,6].>=0)/N1;
 TDIST[t,7]=SUMC(NSDOTR[1:N1,7].>=0)/N1;
 TDIST[t,8]=SUMC(NSDOTR[1:N1,8].>=0)/N1;
 TDIST[t,9]=SUMC(NSDOTR[1:N1,9].>=0)/N1;
 TDIST[t,10]=SUMC(NSDOTR[N1+1:2*N1,1].>=0)/N1;   /* second type */
 TDIST[t,11]=SUMC(NSDOTR[N1+1:2*N1,2].>=0)/N1;
 TDIST[t,12]=SUMC(NSDOTR[N1+1:2*N1,3].>=0)/N1;
 TDIST[t,13]=SUMC(NSDOTR[N1+1:2*N1,4].>=0)/N1;
 TDIST[t,14]=SUMC(NSDOTR[N1+1:2*N1,5].>=0)/N1;
 TDIST[t,15]=SUMC(NSDOTR[N1+1:2*N1,6].>=0)/N1;
 TDIST[t,16]=SUMC(NSDOTR[N1+1:2*N1,7].>=0)/N1;
 TDIST[t,17]=SUMC(NSDOTR[N1+1:2*N1,8].>=0)/N1;
 TDIST[t,18]=SUMC(NSDOTR[N1+1:2*N1,9].>=0)/N1;
 TDIST[t,19]=SUMC(NSDOTR[2*N1+1:3*N1,1].>=0)/N1;   /* third type */
 TDIST[t,20]=SUMC(NSDOTR[2*N1+1:3*N1,2].>=0)/N1;
 TDIST[t,21]=SUMC(NSDOTR[2*N1+1:3*N1,3].>=0)/N1;
 TDIST[t,22]=SUMC(NSDOTR[2*N1+1:3*N1,4].>=0)/N1;
 TDIST[t,23]=SUMC(NSDOTR[2*N1+1:3*N1,5].>=0)/N1;
 TDIST[t,24]=SUMC(NSDOTR[2*N1+1:3*N1,6].>=0)/N1;
 TDIST[t,25]=SUMC(NSDOTR[2*N1+1:3*N1,7].>=0)/N1;
 TDIST[t,26]=SUMC(NSDOTR[2*N1+1:3*N1,8].>=0)/N1;
 TDIST[t,27]=SUMC(NSDOTR[2*N1+1:3*N1,9].>=0)/N1;
endo;
/*Get the last 100 averages of inventory stock distribution*/
SON1=ZEROS(3,9);
SON1[1,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,1:9]))'/AVR;
```

```
SON1[2,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,10:18]))'/AVR;
SON1[3,.]=(SUMC(TDIST[(FINALD-AVR+1):FINALD,19:27]))'/AVR;
SON2=ZEROS(3,3);
SON2[1,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,1:3]))'/AVR;
SON2[2,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,4:6]))'/AVR;
SON2[3,.]=(SUMC(CDIST[(FINALD-AVR+1):FINALD,7:9]))'/AVR;
SON3=ZEROS(3,3);
SON3[1,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,1:3]))'/AVR;
SON3[2,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,4:6]))'/AVR;
SON3[3,.]=(SUMC(SDIST[(FINALD-AVR+1):FINALD,7:9]))'/AVR;
```

WITH IMITATION
FUNDAMENTAL EQUILIBRIUM CASE
Number of players of each type (select as even) 20.000000
The total number of players 60.000000
The discount factor (select between zero and one) 0.60000000
The probability of mistake 0.050000000
Probability of Imitation 1.0000000
Final date -- Number of times market opens 1000.0000
The utilities from consuming own type of good
u11 200.00000
u22 200.00000
u33 200.00000
The disutilities from production for each agent
D1 -100.00000
D2 -100.00000
D3 -100.00000
The inventory holding costs
c11 0.10000000
c12 10.000000
c13 25.000000
c21 0.10000000
c22 10.000000
c23 25.000000
c31 0.10000000
c32 10.000000
c33 25.000000
AVERAGES OF LAST 100.00000 MARKETS OPENNED.
TRADE OF

	1_1	1_2	1_3
TYPE I	0.87700000	0.050000000	0.27700000
TYPE II	0.15000000	1.0000000	0.0000000
TYPE III	0.58550000	0.0000000	1.0000000
	2_1	2_2	2_3
TYPE I	1.0000000	0.73650000	0.071000000
TYPE II	0.74700000	0.90000000	0.40000000
TYPE III	0.52050000	0.57350000	1.0000000
	3_1	3_2	3_3
TYPE I	1.0000000	0.24500000	0.54250000
TYPE II	1.0000000	0.35000000	0.50350000
TYPE III	0.91300000	0.80000000	0.80000000

CONSUMPTION

	GOOD1	GOOD2	GOOD3
TYPE I	1.0000000	0.0000000	0.65150000
TYPE II	0.0000000	1.0000000	0.0000000
TYPE III	0.0000000	0.25000000	1.0000000

STOK

	GOOD1	GOOD2	GOOD3
TYPE I	0.0035000000	0.99050000	0.0060000000
TYPE II	0.47250000	0.0045000000	0.52300000
TYPE III	0.96450000	0.031500000	0.0040000000

```
WITHOUT IMITATION
FUNDAMENTAL EQUILIBRIUM CASE
Number of players of each type (select as even)        20.000000
The total number of players        60.000000
The discount factor (select between zero and one)       0.60000000
The probability of mistake      0.050000000
Probability of Imitation        0.0000000
Final date -- Number of times market opens        1000.0000
The utilities from consuming own type of good
u11         200.00000
u22         200.00000
u33         200.00000
The disutilities from production for each agent
D1        -100.00000
D2        -100.00000
D3        -100.00000
The inventory holding costs
c11        0.10000000
c12         10.000000
c13         25.000000
c21        0.10000000
c22         10.000000
c23         25.000000
c31        0.10000000
c32         10.000000
c33         25.000000
AVERAGES OF LAST        100.00000  MARKETS OPENNED.
TRADE OF
                  1_1             1_2             1_3
TYPE I         0.50000000      0.050000000      0.25000000
TYPE II        0.40000000       1.0000000       0.0000000
TYPE III       0.70000000       0.10000000      1.0000000
                  2_1             2_2             2_3
TYPE I          1.0000000       0.50100000      0.0000000
TYPE II        0.40000000       0.55000000      0.40000000
TYPE III       0.80000000       0.42400000      1.0000000
                  3_1             3_2             3_3
TYPE I          1.0000000       0.57850000      0.38600000
TYPE II         1.0000000       0.99950000      0.44900000
TYPE III       0.60000000       0.25000000      0.50000000
CONSUMPTION
                 GOOD1           GOOD2           GOOD3
TYPE I          1.0000000       0.0000000       0.0000000
TYPE II         0.0000000       1.0000000       0.0000000
TYPE III        0.0000000       0.0000000       1.0000000
STOK
                 GOOD1           GOOD2           GOOD3
TYPE I       0.0025000000      0.98400000     0.013500000
TYPE II        0.46300000     0.0050000000      0.53200000
TYPE III       0.90550000      0.091000000     0.0035000000
```

WITH IMITATION
SPECULATIVE EQUILIBRIUM CASE
Number of players of each type (select as even) 20.000000
The total number of players 60.000000
The discount factor (select between zero and one) 0.60000000
The probability of mistake 0.050000000
Probability of Imitation 1.0000000
Final date -- Number of times market opens 1000.0000
The utilities from consuming own type of good
u11 200.00000
u22 200.00000
u33 200.00000
The disutilities from production for each agent
D1 -100.00000
D2 -100.00000
D3 -100.00000
The inventory holding costs
c11 0.10000000
c12 10.000000
c13 10.100000
c21 0.10000000
c22 10.000000
c23 10.100000
c31 0.10000000
c32 10.000000
c33 10.100000
AVERAGES OF LAST 100.00000 MARKETS OPENNED.
TRADE OF

	1_1	1_2	1_3
TYPE I	0.95000000	0.20000000	0.25000000
TYPE II	0.40500000	1.0000000	0.0000000
TYPE III	0.40000000	0.0000000	1.0000000
	2_1	2_2	2_3
TYPE I	1.0000000	0.46700000	1.0000000
TYPE II	0.82650000	0.80000000	0.15000000
TYPE III	0.72400000	0.51750000	0.95000000
	3_1	3_2	3_3
TYPE I	1.0000000	0.0000000	0.60000000
TYPE II	1.0000000	1.0000000	0.11250000
TYPE III	0.10000000	0.15000000	0.75150000

CONSUMPTION

	GOOD1	GOOD2	GOOD3
TYPE I	1.0000000	0.0000000	0.0000000
TYPE II	0.0000000	1.0000000	0.0000000
TYPE III	0.0000000	0.46100000	1.0000000

STOK

	GOOD1	GOOD2	GOOD3
TYPE I	0.0020000000	0.69850000	0.29950000
TYPE II	0.53550000	0.0080000000	0.45650000
TYPE III	0.96100000	0.033500000	0.0055000000

```
WITHOUT IMITATION
SPECULATIVE EQUILIBRIUM CASE
Number of players of each type (select as even)        20.000000
The total number of players     60.000000
The discount factor (select between zero and one)       0.60000000
The probability of mistake      0.050000000
Probability of Imitation        0.0000000
Final date -- Number of times market opens        1000.0000
The utilities from consuming own type of good
u11         200.00000
u22         200.00000
u33         200.00000
The disutilities from production for each agent
D1        -100.00000
D2        -100.00000
D3        -100.00000
The inventory holding costs
c11        0.10000000
c12         10.000000
c13         10.100000
c21        0.10000000
c22         10.000000
c23         10.100000
c31        0.10000000
c32         10.000000
c33         10.100000
AVERAGES OF LAST        100.00000  MARKETS OPENNED.
TRADE OF
                  1_1             1_2             1_3
TYPE I         0.60000000      0.11750000      0.25000000
TYPE II        0.50000000       1.0000000     0.050000000
TYPE III       0.35000000      0.20000000       1.0000000
                  2_1             2_2             2_3
TYPE I          1.0000000      0.58800000      0.75000000
TYPE II        0.45000000      0.70000000      0.15000000
TYPE III       0.82400000      0.48400000      0.95000000
                  3_1             3_2             3_3
TYPE I         0.95000000      0.65000000      0.46100000
TYPE II         1.0000000       1.0000000      0.68550000
TYPE III       0.45000000      0.15000000      0.65000000
CONSUMPTION
                 GOOD1           GOOD2           GOOD3
TYPE I          1.0000000       0.0000000       0.0000000
TYPE II         0.0000000       1.0000000       0.0000000
TYPE III        0.0000000       0.0000000       1.0000000
STOK
                 GOOD1           GOOD2           GOOD3
TYPE I       0.0045000000      0.75800000      0.23750000
TYPE II        0.49400000    0.0070000000      0.49900000
TYPE III       0.87750000      0.11400000    0.0085000000
```

APPENDIX D

Graphics of Dynamic Optimization Part: (100 RUNS)

In the graphics of the density functions, y-axis gives us the proportion of runs for each periods, x-axis gives us the acceptance ratio of players' decisions for the periods of 10, 40, 160, 640 and 1000. For example if the acceptance ratio is equal to 1, all the players will accept the decision, if it is between 0.9 and 1, between 90 percent and 100 percent of players in a game, will accept the decision.

In the graphics of the stok distribution functions, time plot of means and standard deviations of players holding goods are given for the periods of 10, 20, 40, 80, 160, 320, 640 and 1000.

1. Density of Type 1 Player Trading Good 2 For Good 1-fun. eq. full imit.
2. Density of Type 1 Player Trading Good 2 For Good 1-fun. eq. half imit.
3. Density of Type 1 Player Trading Good 2 For Good 1-fun. eq. no imit.
4. Density of Type 1 Player Trading Good 2 For Good 3-fun. eq. full imit.
5. Density of Type 1 Player Trading Good 2 For Good 3-fun. eq. half imit.
6. Density of Type 1 Player Trading Good 2 For Good 3-fun. eq. no imit.
7. Density of Type 1 Player For Consumption of Good 1-fun. eq. full imit.
8. Density of Type 1 Player For Consumption of Good 1-fun. eq. half imit.

9. Density of Type 1 Player For Consumption of Good 1-fun. eq. no imit.

10. Density of Type 1 Player Trading Good 2 For Good 1-spec. eq. full imit.

11. Density of Type 1 Player Trading Good 2 For Good 1-spec. eq. half imit.

12. Density of Type 1 Player Trading Good 2 For Good 1-spec. eq. no imit.

13. Density of Type 1 Player Trading Good 2 For Good 3-spec. eq. full imit.

14. Density of Type 1 Player Trading Good 2 For Good 3-spec. eq. half imit.

15. Density of Type 1 Player Trading Good 2 For Good 3-spec. eq. no imit.

16. Density of Type 1 Player Trading Good 3 For Good 1-spec. eq. full imit.

17. Density of Type 1 Player Trading Good 3 For Good 1-spec. eq. half imit.

18. Density of Type 1 Player Trading Good 3 For Good 1-spec. eq. no imit.

19. Density of Type 1 Player For Consumption of Good 1-spec. eq. full imit.

20. Density of Type 1 Player For Consumption of Good 1-spec. eq. half imit.

21. Density of Type 1 Player For Consumption of Good 1-spec. eq. no imit.

APPENDIX E

Graphics of Kiyotaki and Wright Part: (100 RUNS)

In the graphics of the density functions, y-axis gives us the proportion of runs for each periods, x-axis gives us the acceptance ratio of players' decisions for the periods of 10, 40, 160, 640 and 1000. For example if the acceptance ratio is equal to 1, all the players will accept the decision, if it is between 0.9 and 1, between 90 percent and 100 percent of players in a game, will accept the decision.

In the graphics of the stok distribution functions, time plot of means and standard deviations of players holding goods are given for the periods of 10, 20, 40, 80, 160, 320, 640 and 1000.

1. Density of Type 1 Player Trading Good 2 For Good 1-fun. eq. full imit.
2. Density of Type 1 Player Trading Good 2 For Good 1-fun. eq. half imit.
3. Density of Type 1 Player Trading Good 2 For Good 1-fun. eq. no imit.
4. Density of Type 1 Player Trading Good 2 For Good 3-fun. eq. full imit.
5. Density of Type 1 Player Trading Good 2 For Good 3-fun. eq. half imit.
6. Density of Type 1 Player Trading Good 2 For Good 3-fun. eq. no imit.
7. Density of Type 2 Player Trading Good 1 For Good 2-fun. eq. full imit.
8. Density of Type 2 Player Trading Good 1 For Good 2-fun. eq. half imit.
9. Density of Type 2 Player Trading Good 1 For Good 2-fun. eq. no imit.

10. Density of Type 2 Player Trading Good 1 For Good 3-fun. eq. full imit.

11. Density of Type 2 Player Trading Good 1 For Good 3-fun. eq. half imit.

12. Density of Type 2 Player Trading Good 1 For Good 3-fun. eq. no imit.

13. Density of Type 2 Player Trading Good 3 For Good 1-fun. eq. full imit.

14. Density of Type 2 Player Trading Good 3 For Good 1-fun. eq. half imit.

15. Density of Type 2 Player Trading Good 3 For Good 1-fun. eq. no imit.

16. Density of Type 3 Player Trading Good 1 For Good 2-fun. eq. full imit.

17. Density of Type 3 Player Trading Good 1 For Good 2-fun. eq. half imit.

18. Density of Type 3 Player Trading Good 1 For Good 2-fun. eq. no imit.

19. Density of Type 3 Player Trading Good 1 For Good 3-fun. eq. full imit.

20. Density of Type 3 Player Trading Good 1 For Good 3-fun. eq. half imit.

21. Density of Type 3 Player Trading Good 1 For Good 3-fun. eq. no imit.

22. Density of Type 1 Player For Consumption of Good 2-fun. eq. full imit.

23. Density of Type 1 Player For Consumption of Good 2-fun. eq. half imit.

24. Density of Type 1 Player For Consumption of Good 2-fun. eq. no imit.

25. Density of Type 2 Player For Consumption of Good 2-fun. eq. full imit.

26. Density of Type 2 Player For Consumption of Good 2-fun. eq. half imit.

27. Density of Type 2 Player For Consumption of Good 2-fun. eq. no imit.

28. Density of Type 3 Player For Consumption of Good 3-fun. eq. full imit.

29. Density of Type 3 Player For Consumption of Good 3-fun. eq. half imit.

30. Density of Type 3 Player For Consumption of Good 3-fun. eq. no imit.

31. Stok Distribution of Type 1 Player Holding Good 2-fun. eq. full imit.

32. Stok Distribution of Type 1 Player Holding Good 2-fun. eq. half imit.

33. Stok Distribution of Type 1 Player Holding Good 2-fun. eq. no imit.

34. Stok Distribution of Type 1 Player Holding Good 3-fun. eq. full imit.

35. Stok Distribution of Type 1 Player Holding Good 3-fun. eq. half imit.

36. Stok Distribution of Type 1 Player Holding Good 3-fun. eq. no imit.

37. Stok Distribution of Type 2 Player Holding Good 1-fun. eq. full imit.

38. Stok Distribution of Type 2 Player Holding Good 1-fun. eq. half imit.

39. Stok Distribution of Type 2 Player Holding Good 1-fun. eq. no imit.

40. Stok Distribution of Type 2 Player Holding Good 3-fun. eq. full imit.

41. Stok Distribution of Type 2 Player Holding Good 3-fun. eq. half imit.

42. Stok Distribution of Type 2 Player Holding Good 3-fun. eq. no imit.

43. Stok Distribution of Type 3 Player Holding Good 1-fun. eq. full imit.

44. Stok Distribution of Type 3 Player Holding Good 1-fun. eq. half imit.

45. Stok Distribution of Type 3 Player Holding Good 1-fun. eq. no imit.

61. Density of Type 2 Player Trading Good 1 For Good 3-spec. eq. full imit.

62. Density of Type 2 Player Trading Good 1 For Good 3-spec. eq. half imit.

63. Density of Type 2 Player Trading Good 1 For Good 3-spec. eq. no imit.

64. Density of Type 2 Player Trading Good 3 For Good 1-spec. eq. full imit.

65. Density of Type 2 Player Trading Good 3 For Good 1-spec. eq. half imit.

66. Density of Type 2 Player Trading Good 3 For Good 1-spec. eq. no imit.

67. Density of Type 2 Player Trading Good 3 For Good 2-spec. eq. full imit.

68. Density of Type 2 Player Trading Good 3 For Good 2-spec. eq. half imit.

69. Density of Type 2 Player Trading Good 3 For Good 2-spec. eq. no imit.

70. Density of Type 3 Player Trading Good 1́ For Good 2-spec. eq. full imit.

71. Density of Type 3 Player Trading Good 1 For Good 2-spec. eq. half imit.

72. Density of Type 3 Player Trading Good 1 For Good 2-spec. eq. no imit.

73. Density of Type 3 Player Trading Good 1 For Good 3-spec. eq. full imit.

74. Density of Type 3 Player Trading Good 1 For Good 3-spec. eq. half imit.

75. Density of Type 3 Player Trading Good 1 For Good 3-spec. eq. no imit.

76. Density of Type 1 Player For Consumption of Good 2-spec. eq. full imit.

77. Density of Type 1 Player For Consumption of Good 2-spec. eq. half imit.

78. Density of Type 1 Player For Consumption of Good 2-spec. eq. no imit.

79. Density of Type 2 Player For Consumption of Good 2-spec. eq. full imit.

80. Density of Type 2 Player For Consumption of Good 2-spec. eq. half imit.

81. Density of Type 2 Player For Consumption of Good 2-spec. eq. no imit.

82. Density of Type 3 Player For Consumption of Good 3-spec. eq. full imit.

83. Density of Type 3 Player For Consumption of Good 3-spec. eq. half imit.

84. Density of Type 3 Player For Consumption of Good 3-spec. eq. no imit.

85. Stok Distribution of Type 1 Player Holding Good 2-spec. eq. full imit.

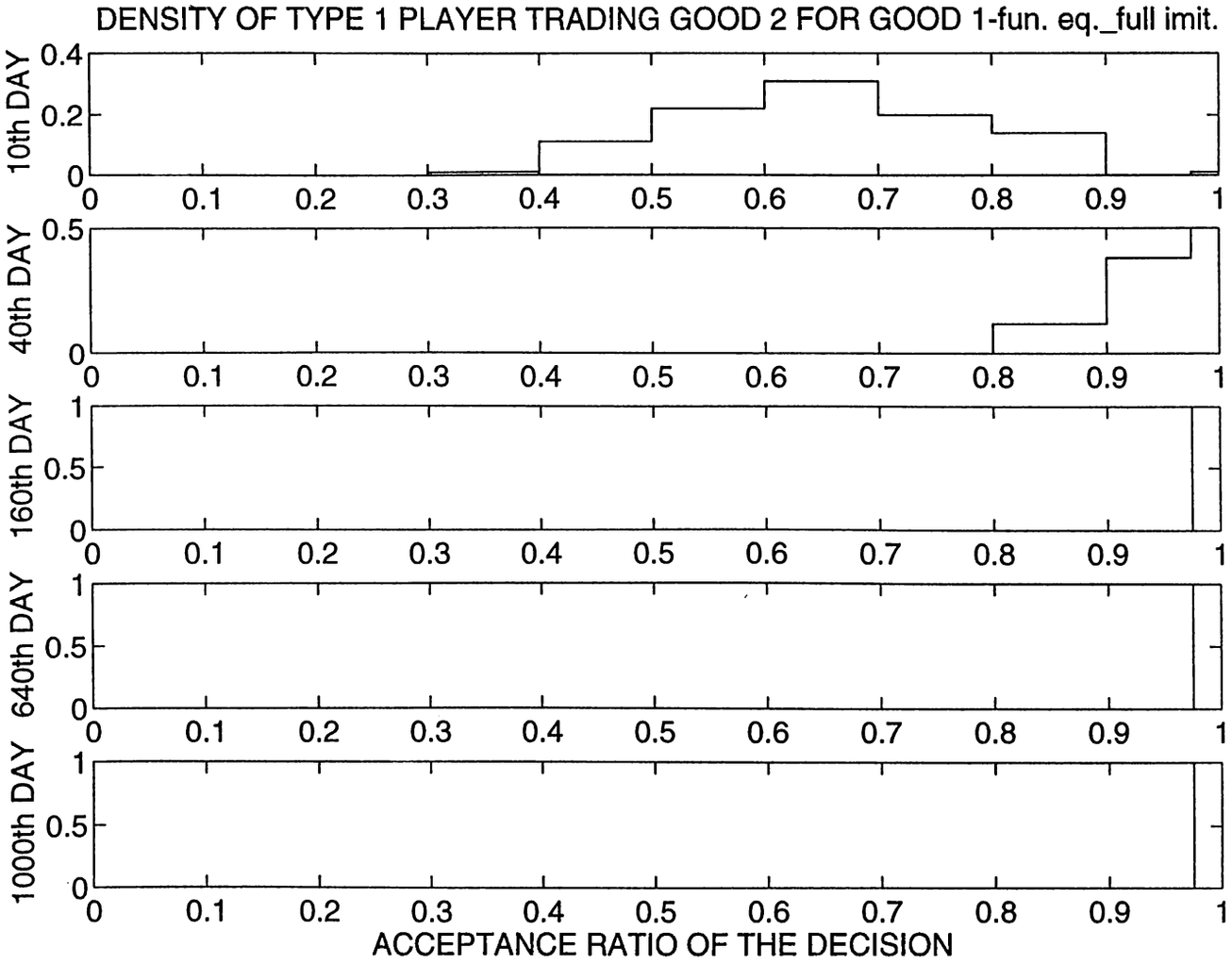
86. Stok Distribution of Type 1 Player Holding Good 2-spec. eq. half imit.

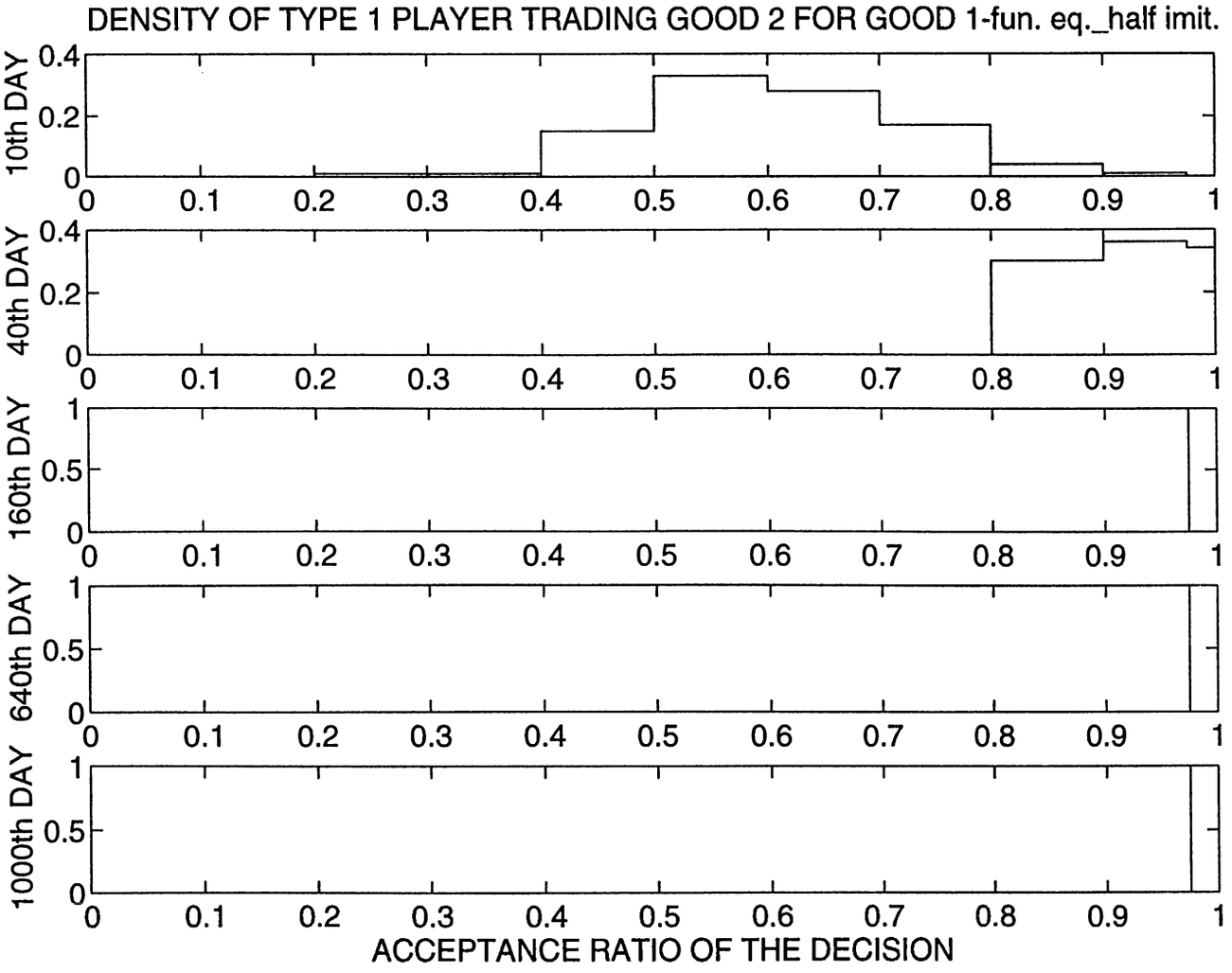
87. Stok Distribution of Type 1 Player Holding Good 2-spec. eq. no imit.

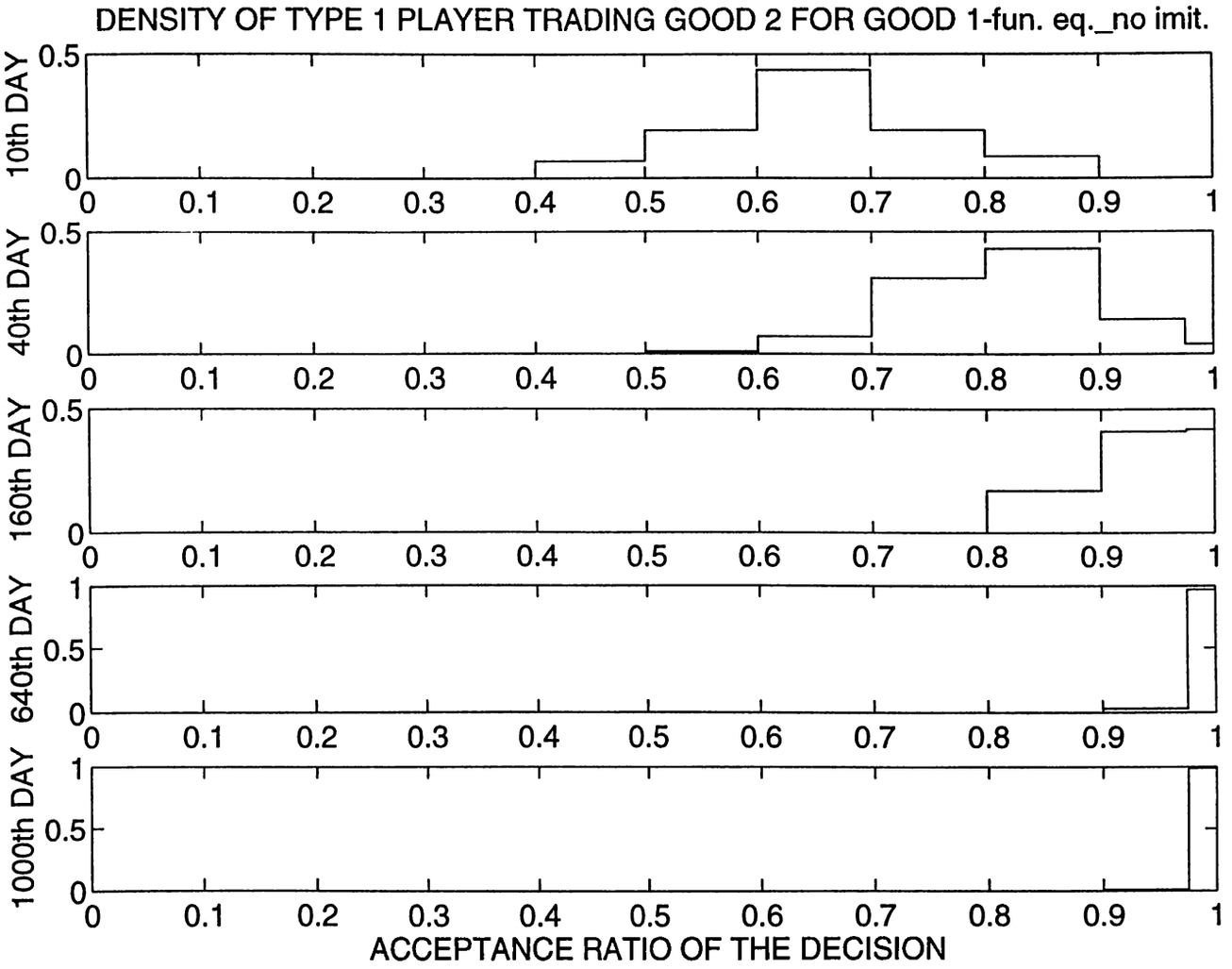
88. Stok Distribution of Type 1 Player Holding Good 3-spec. eq. full imit.

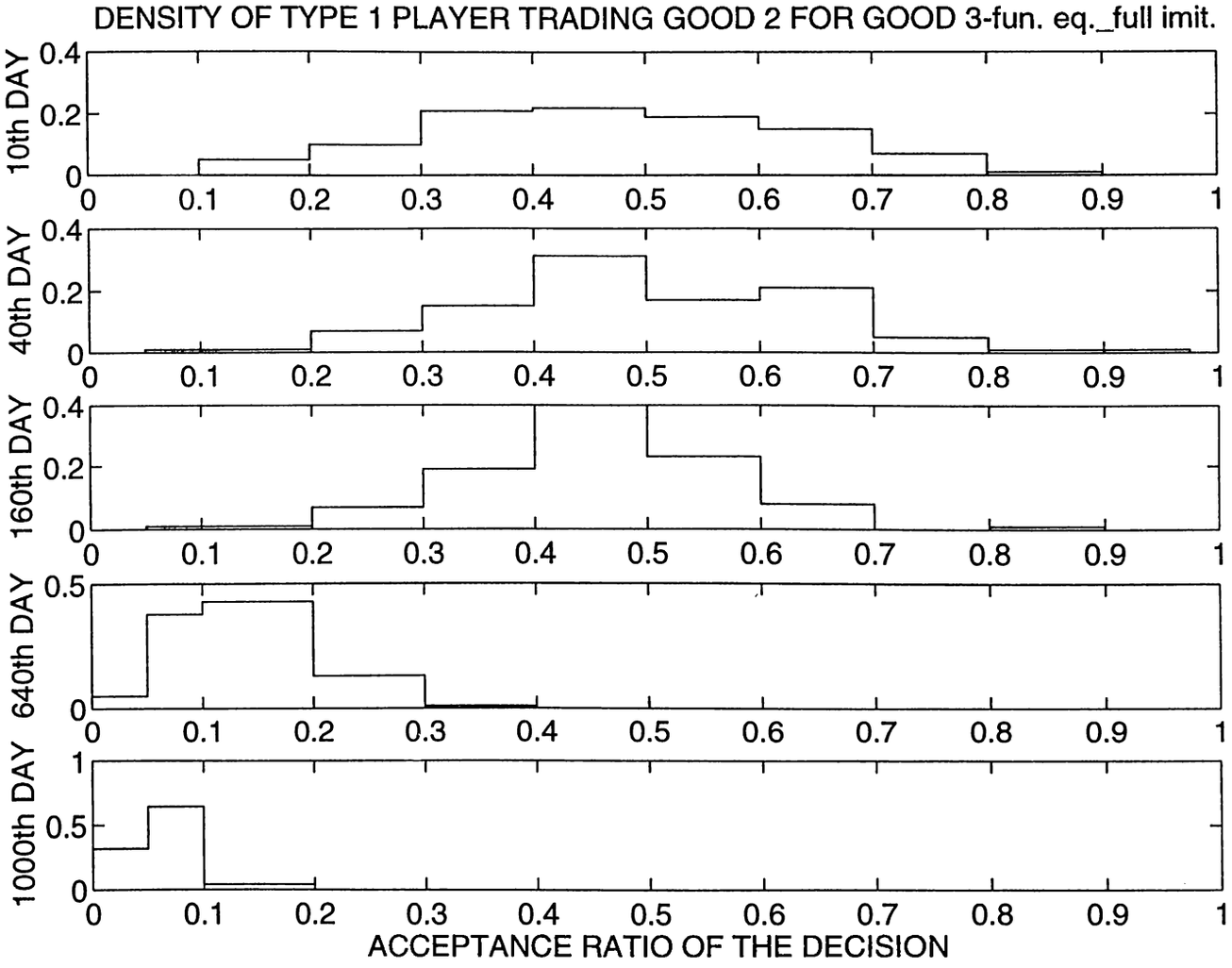
89. Stok Distribution of Type 1 Player Holding Good 3-spec. eq. half imit.

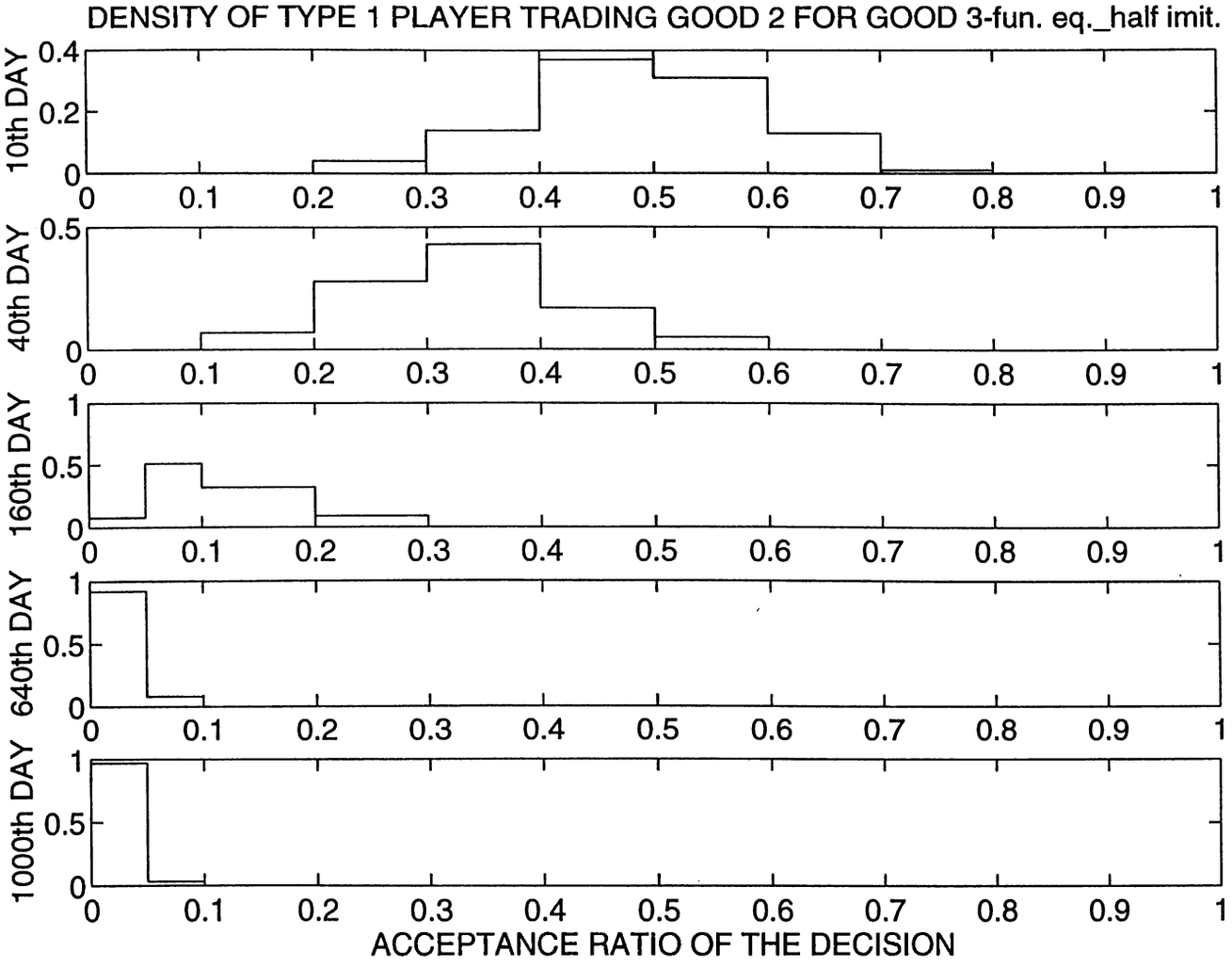
90. Stok Distribution of Type 1 Player Holding Good 3-spec. eq. no imit.

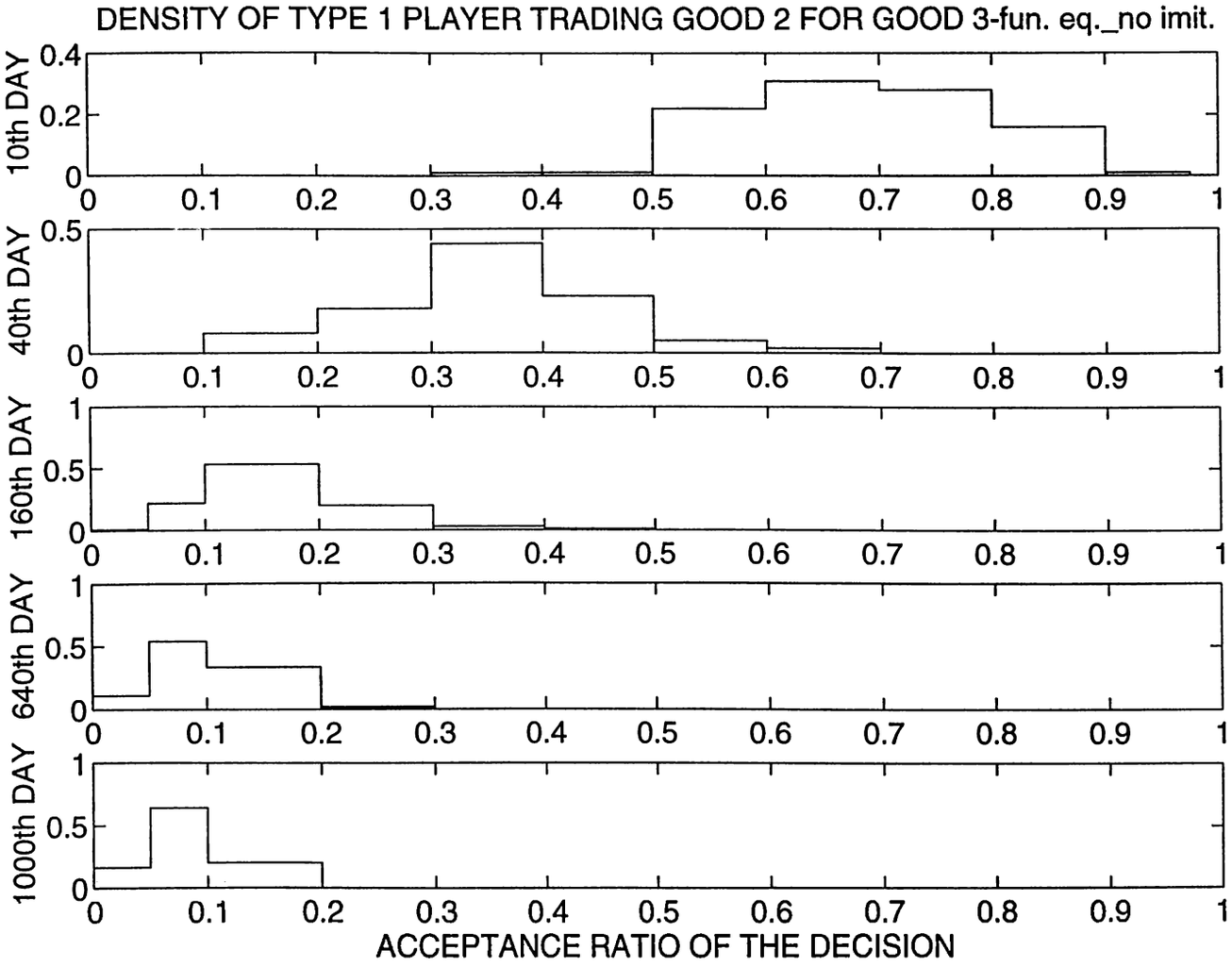

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

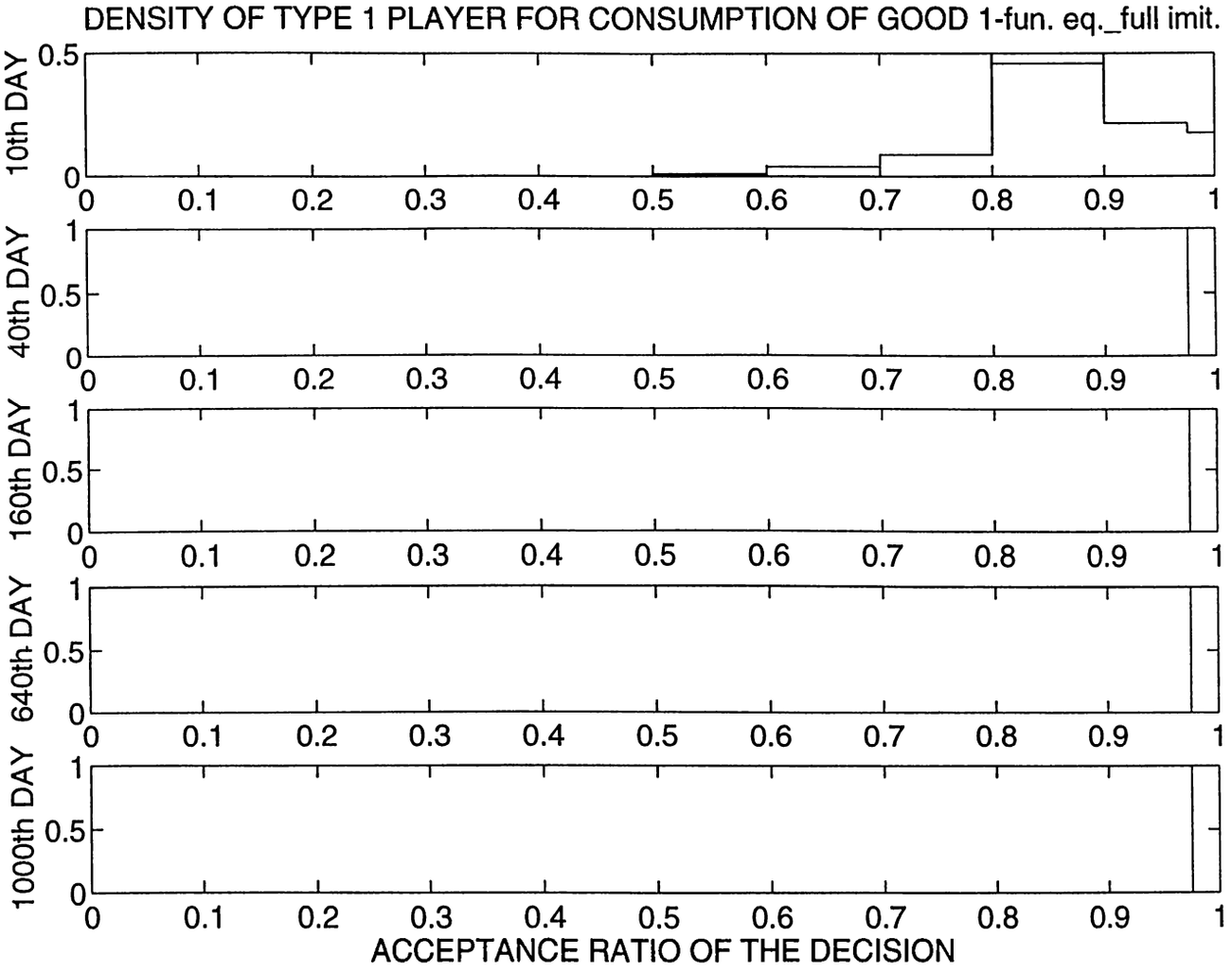

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

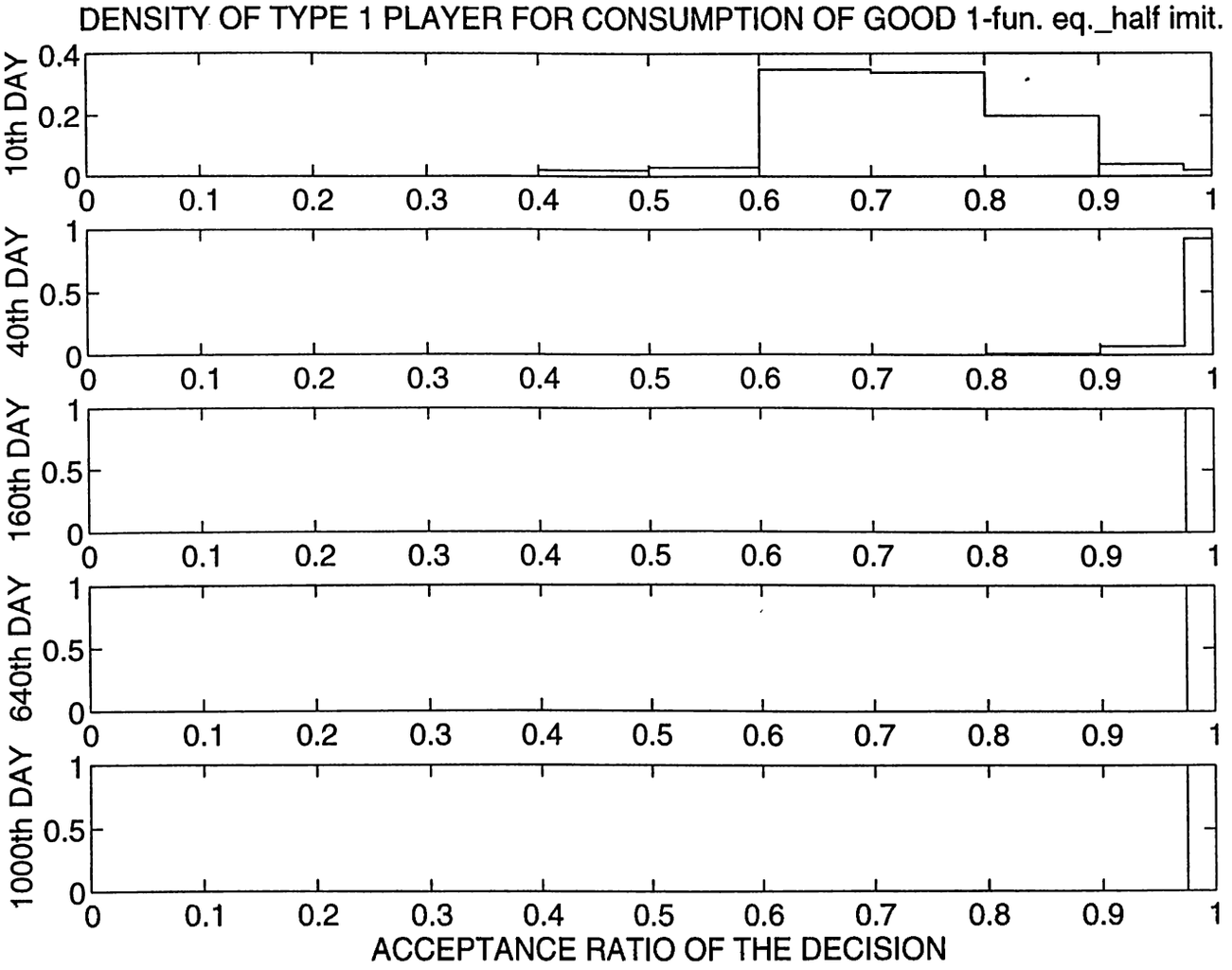

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
ACCEPTANCE RATIO OF THE DECISION

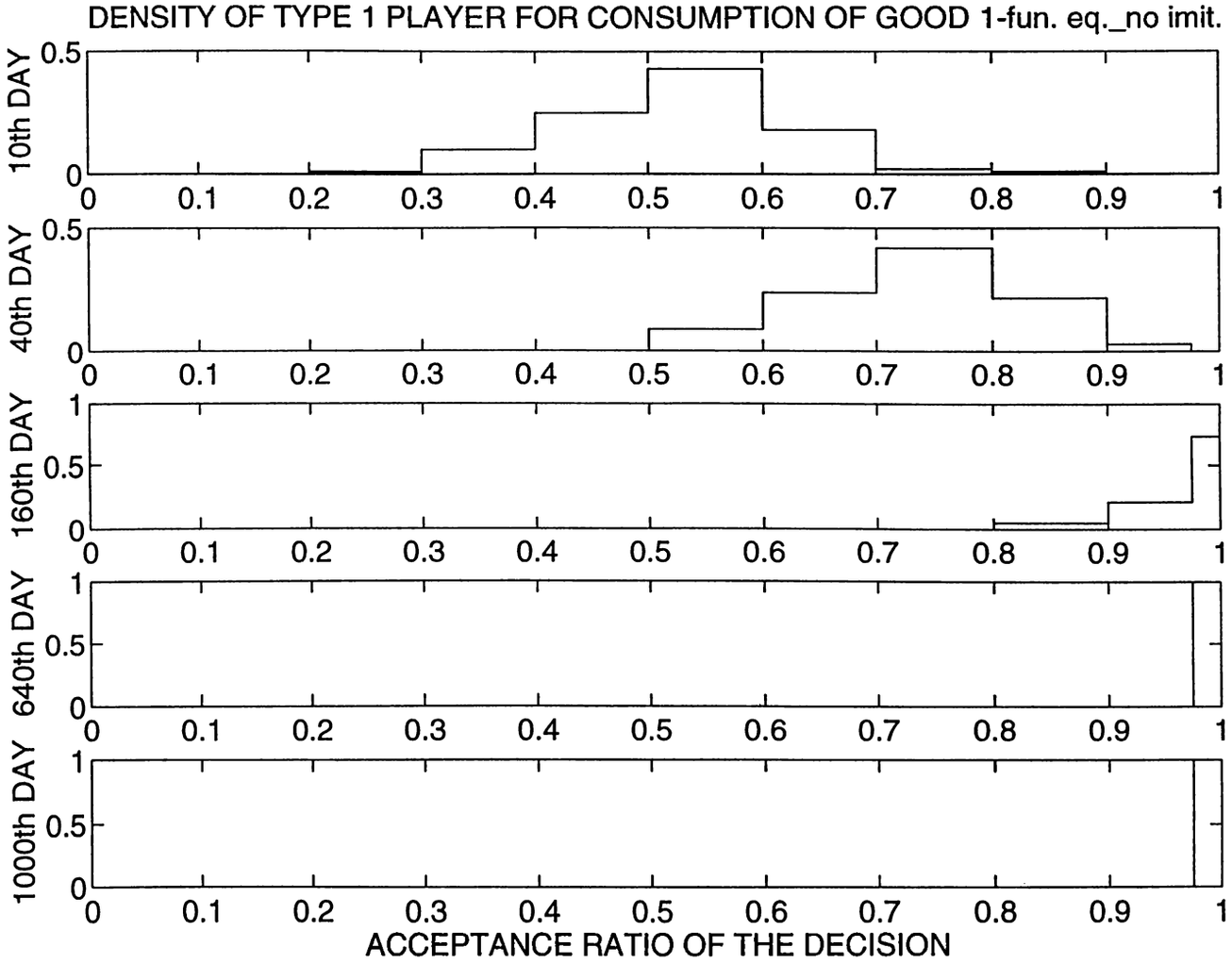

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
0.5
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

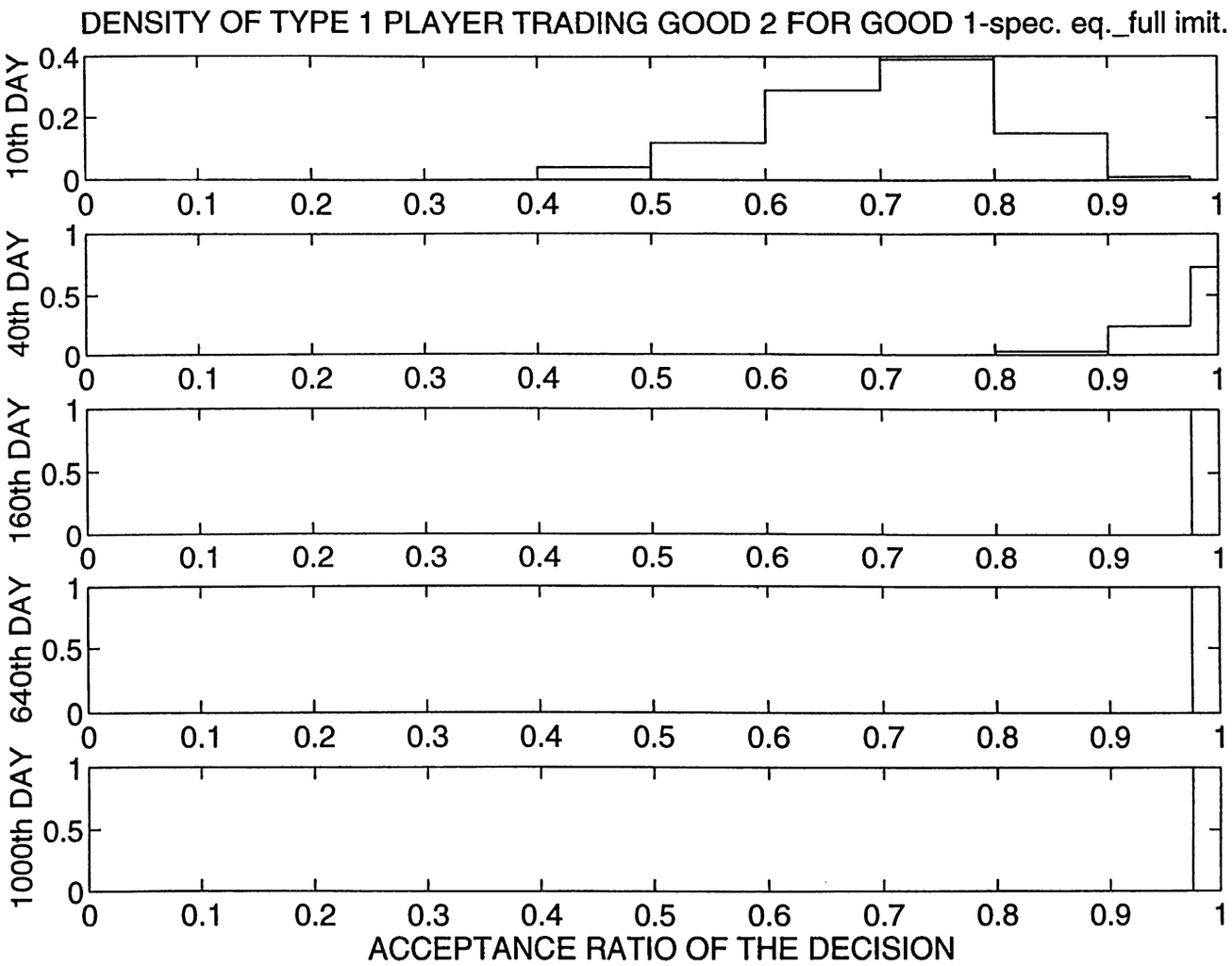

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

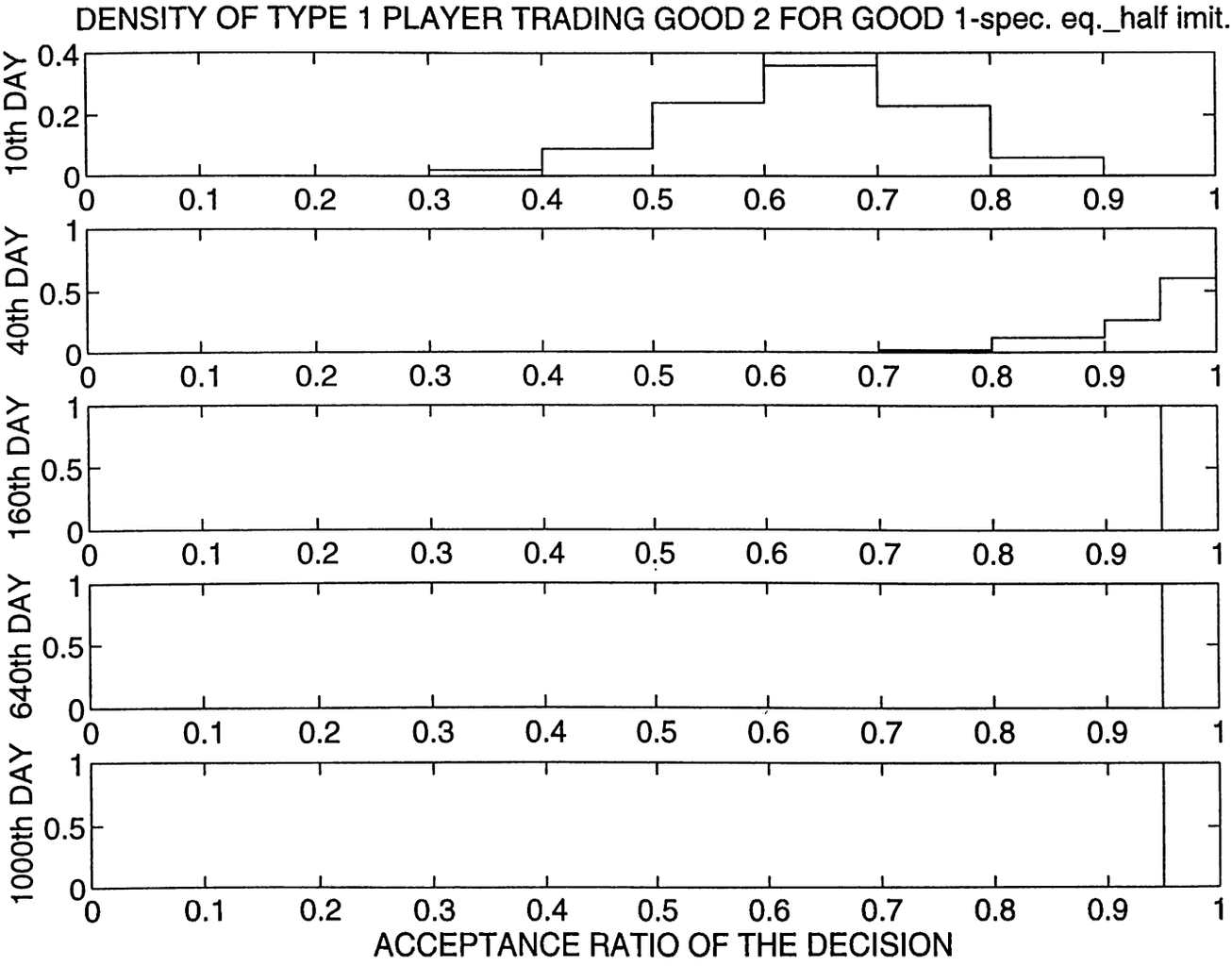

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

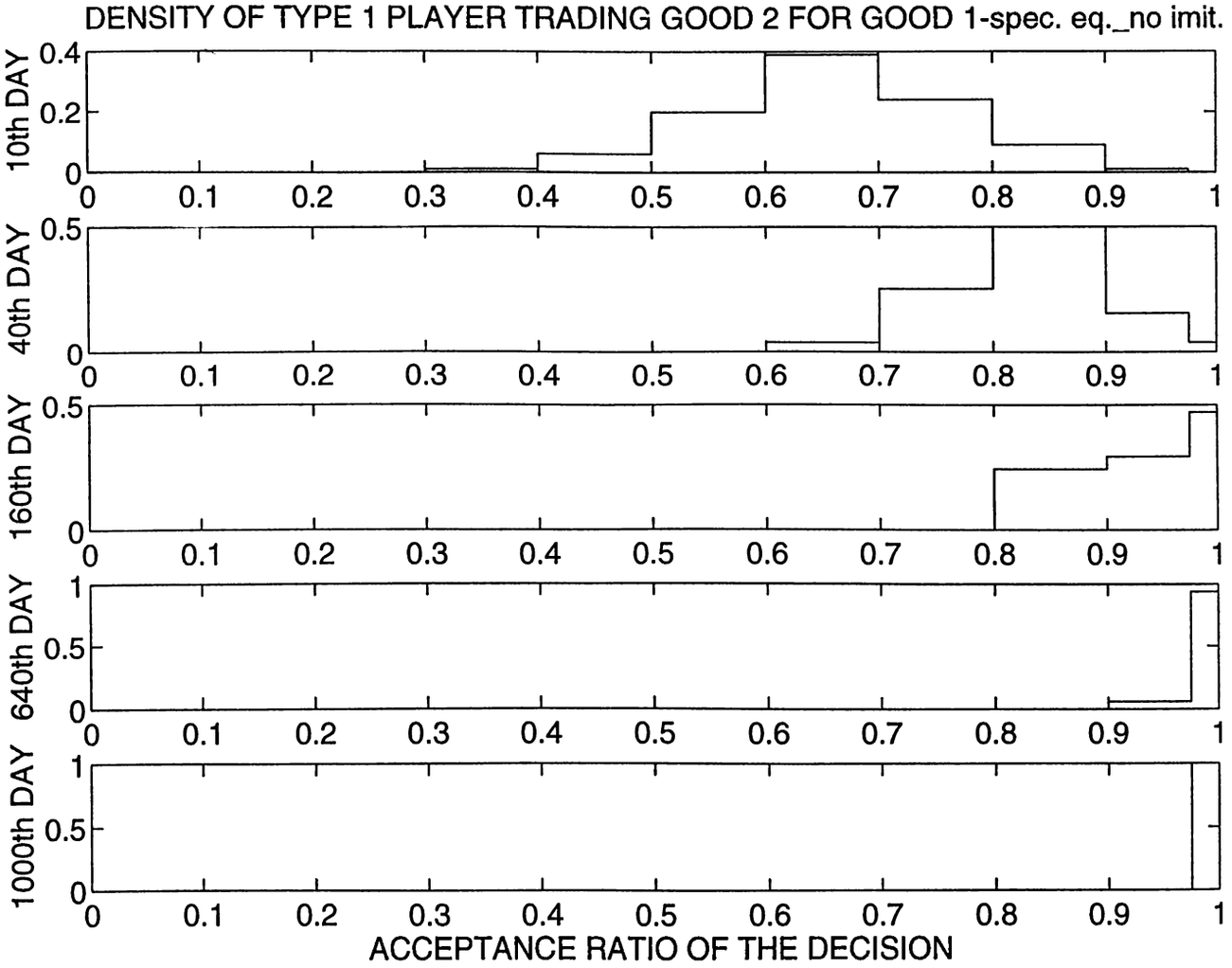

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 1-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.5
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

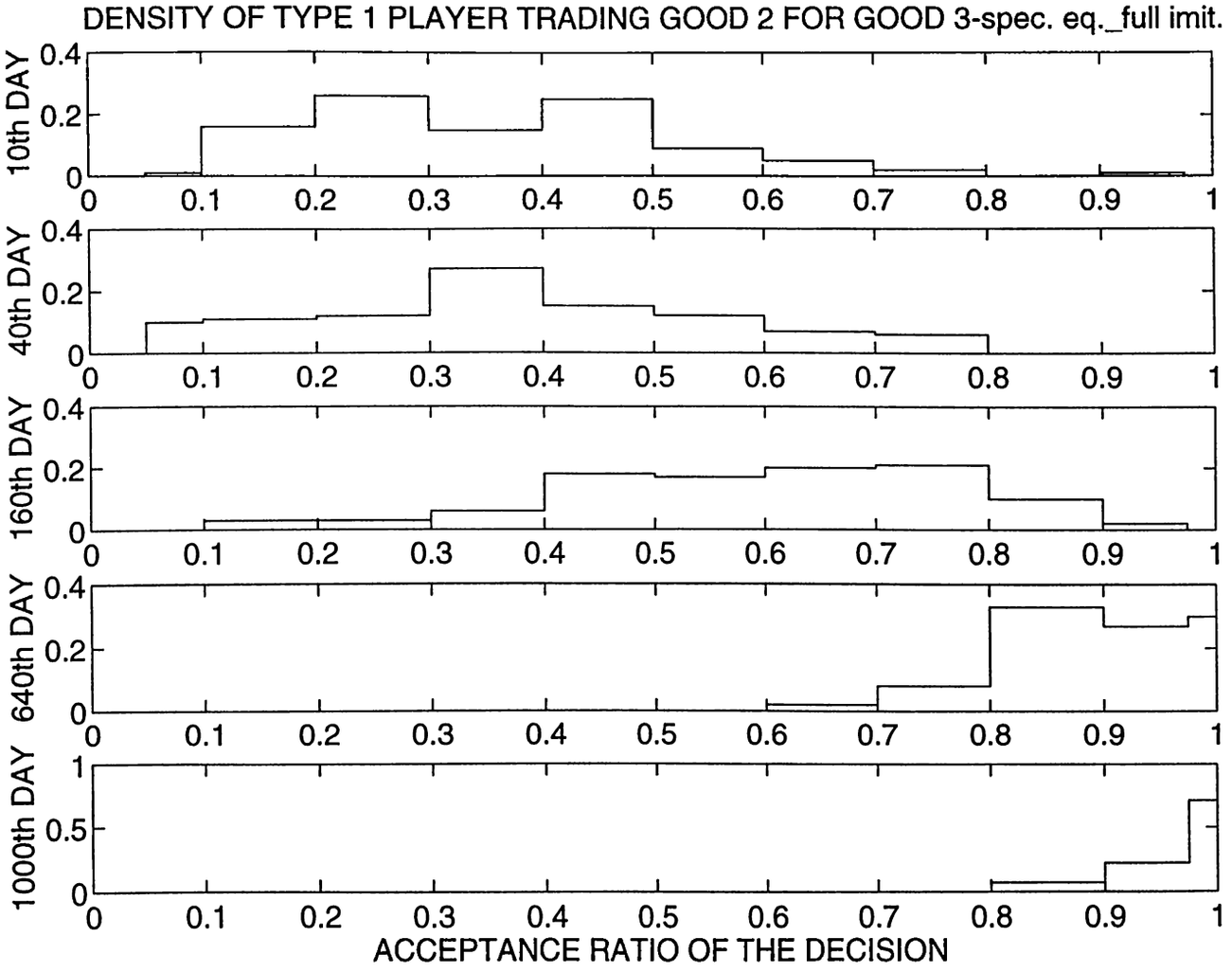

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 1-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

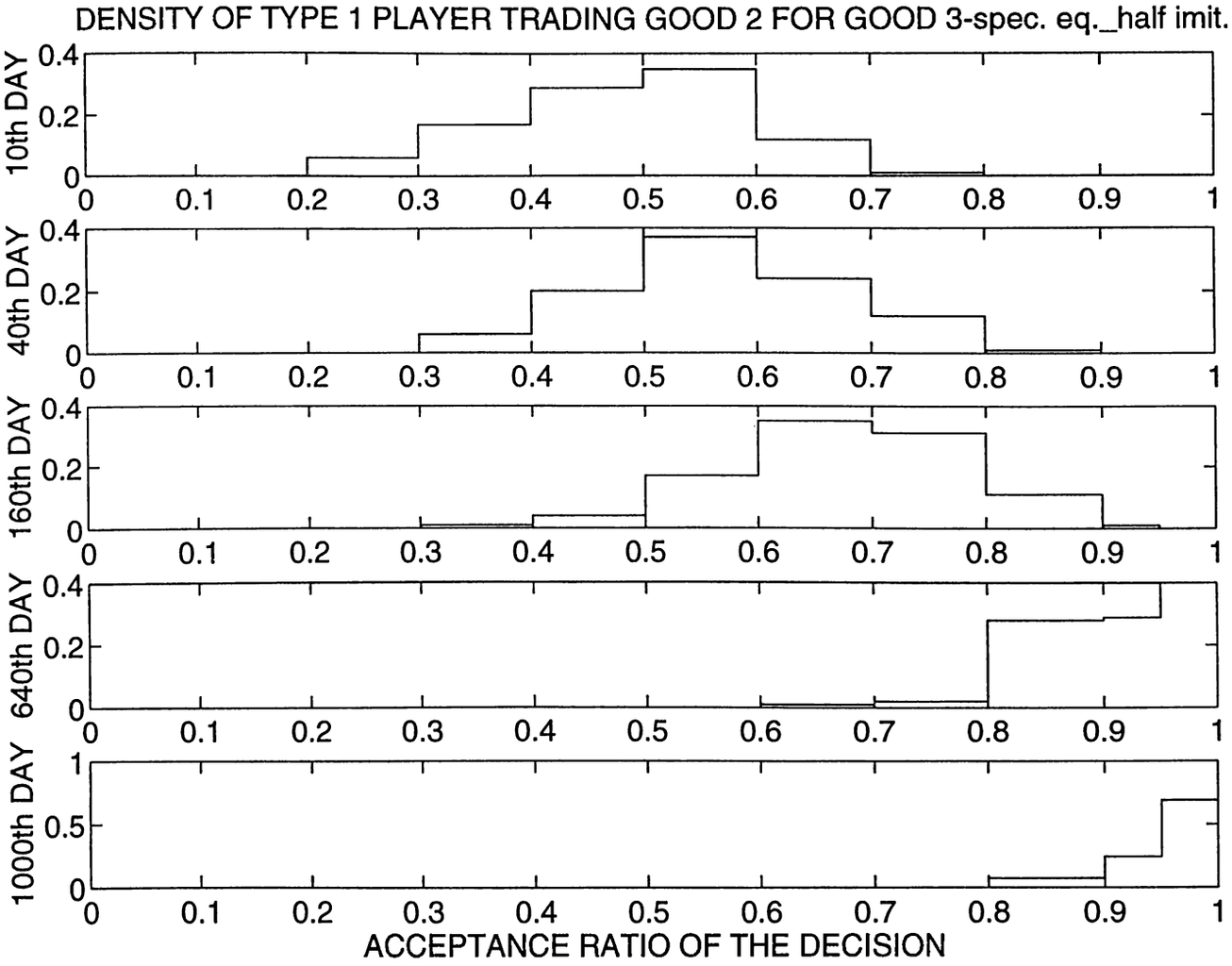

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 1-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

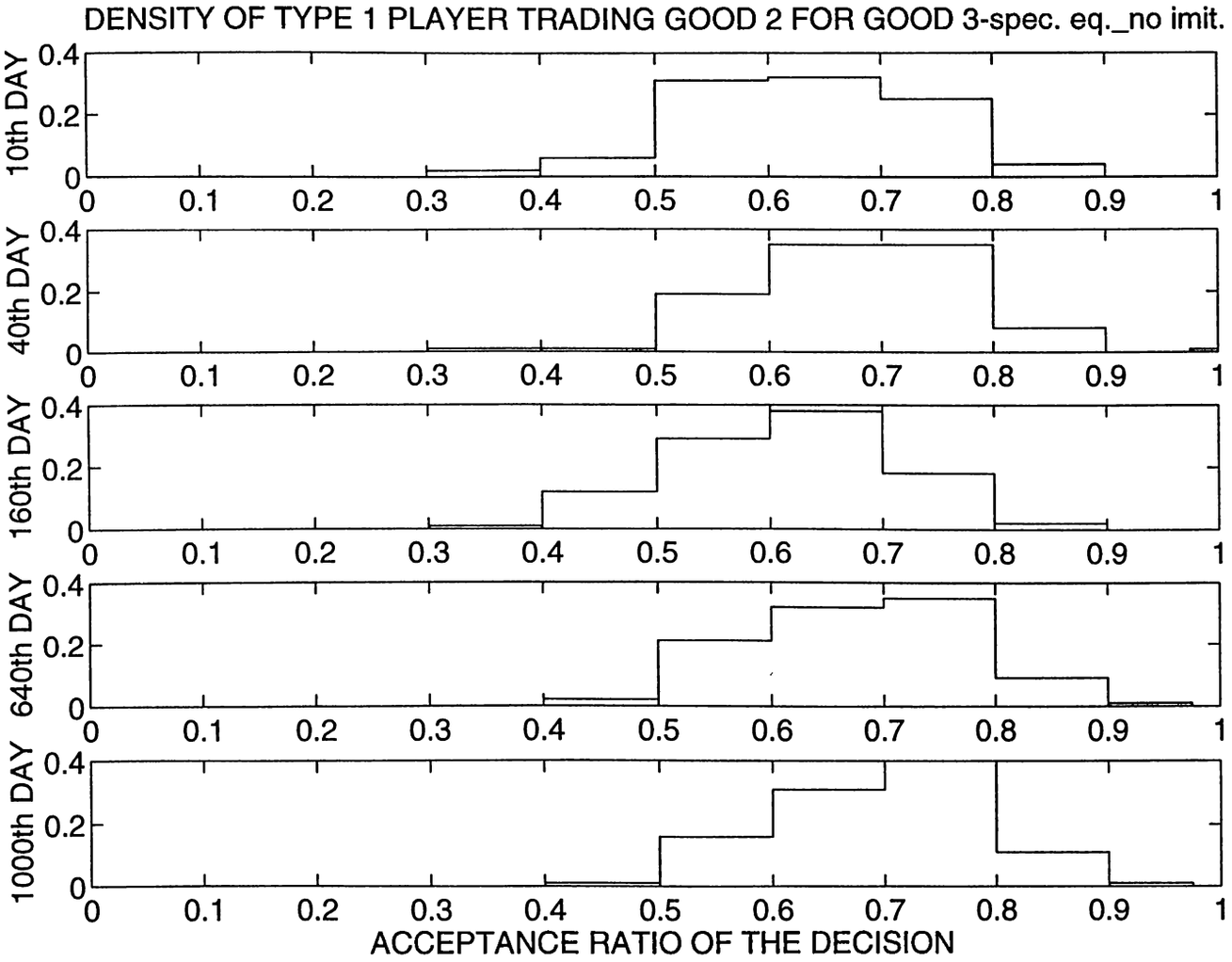

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

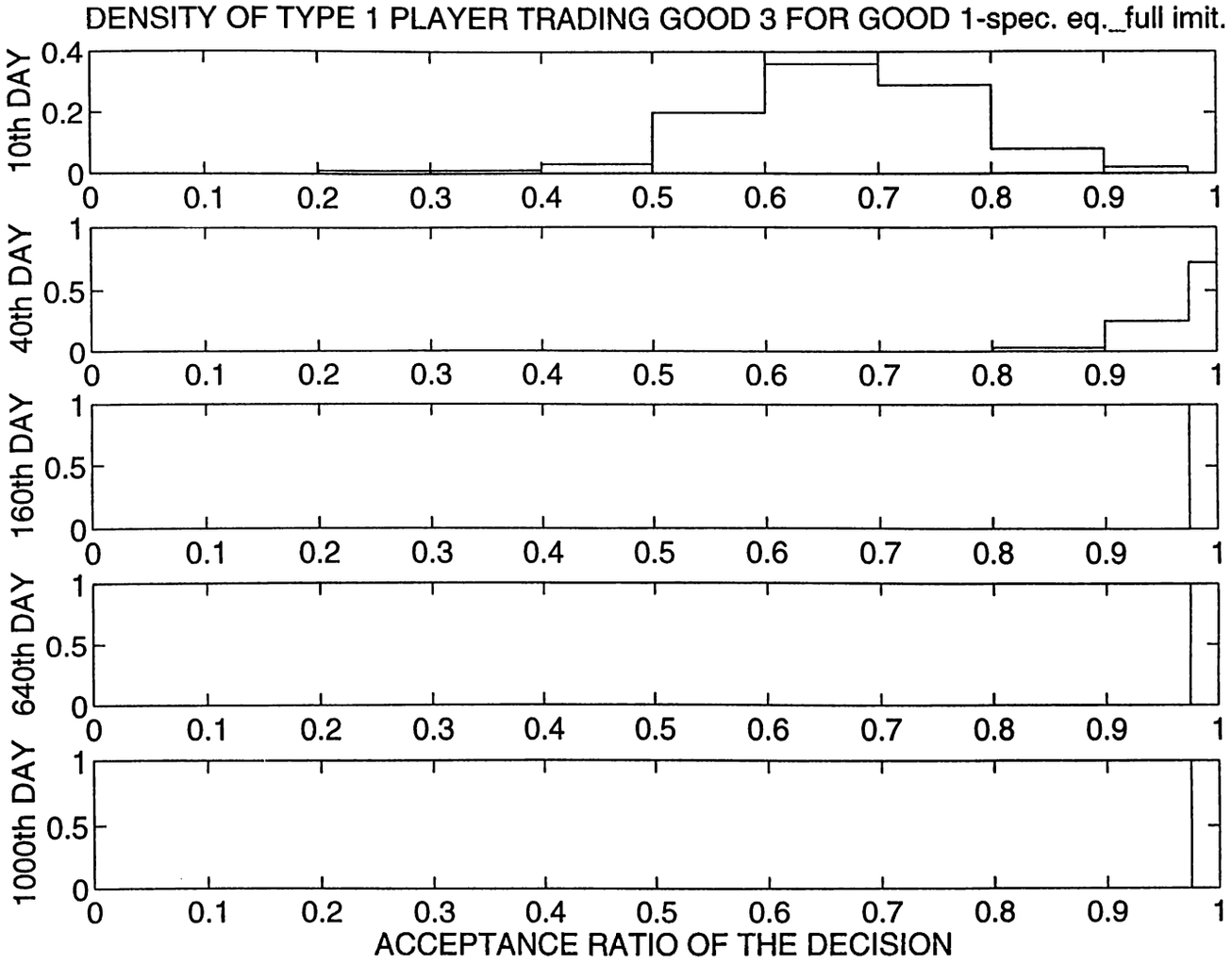

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

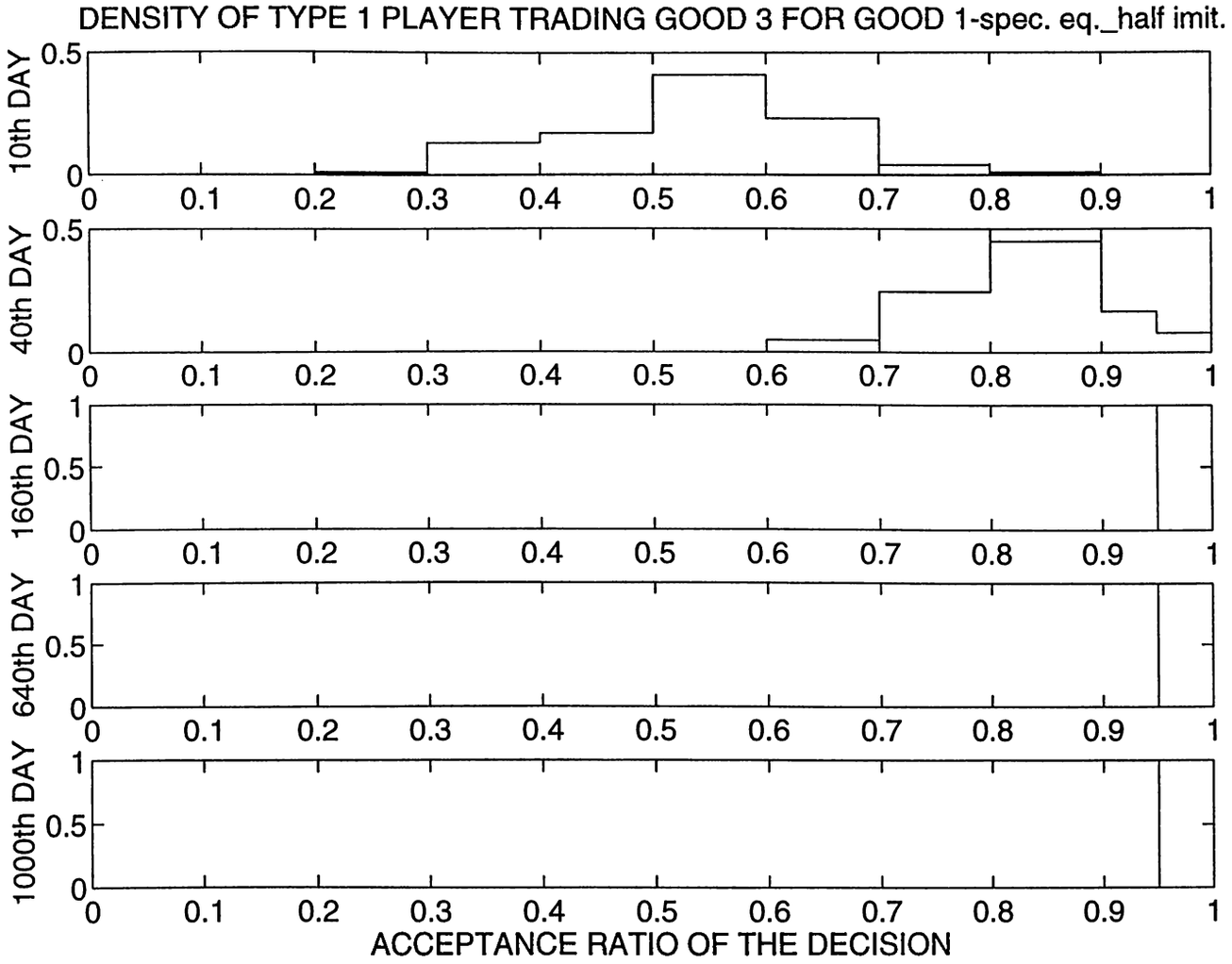

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

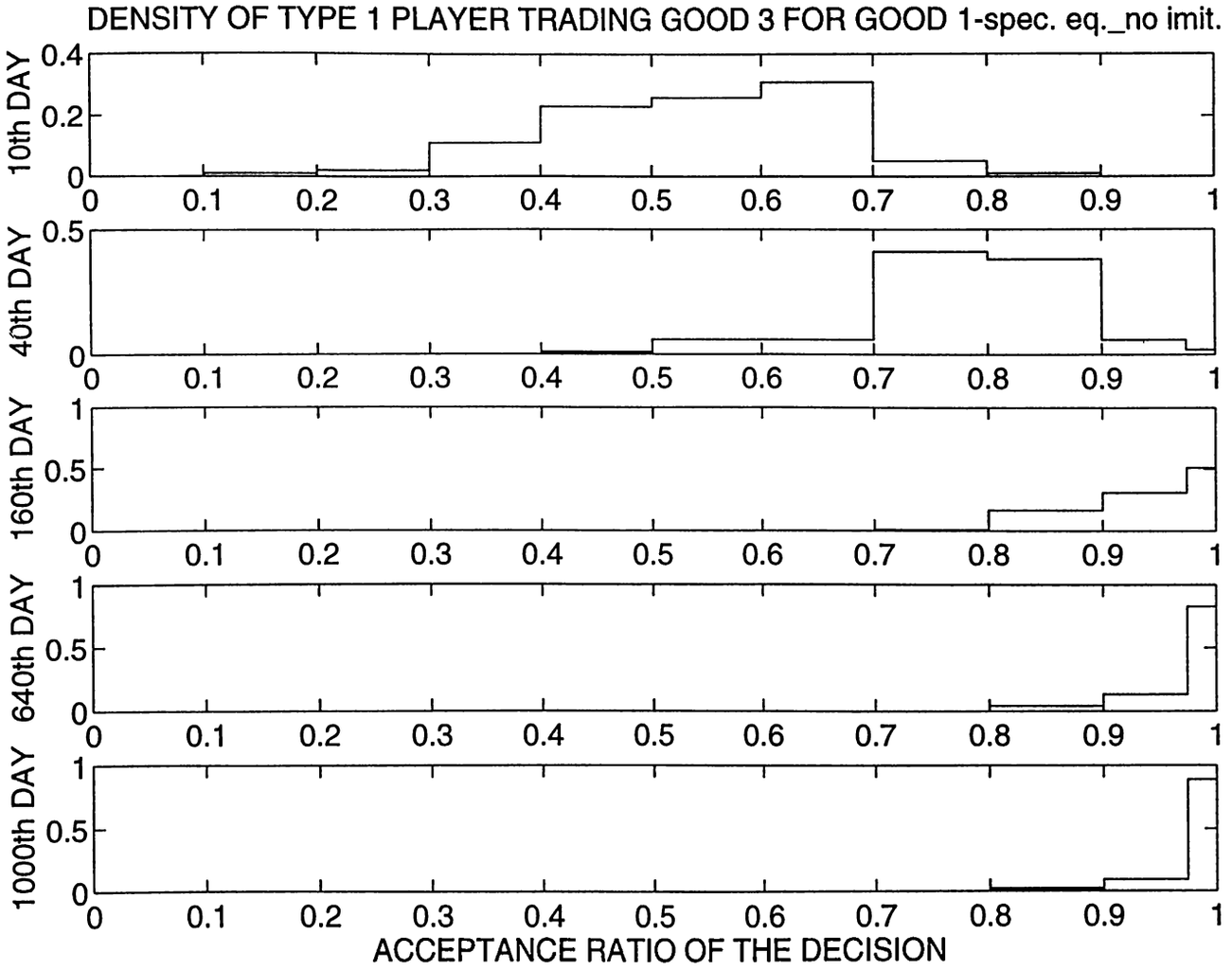

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

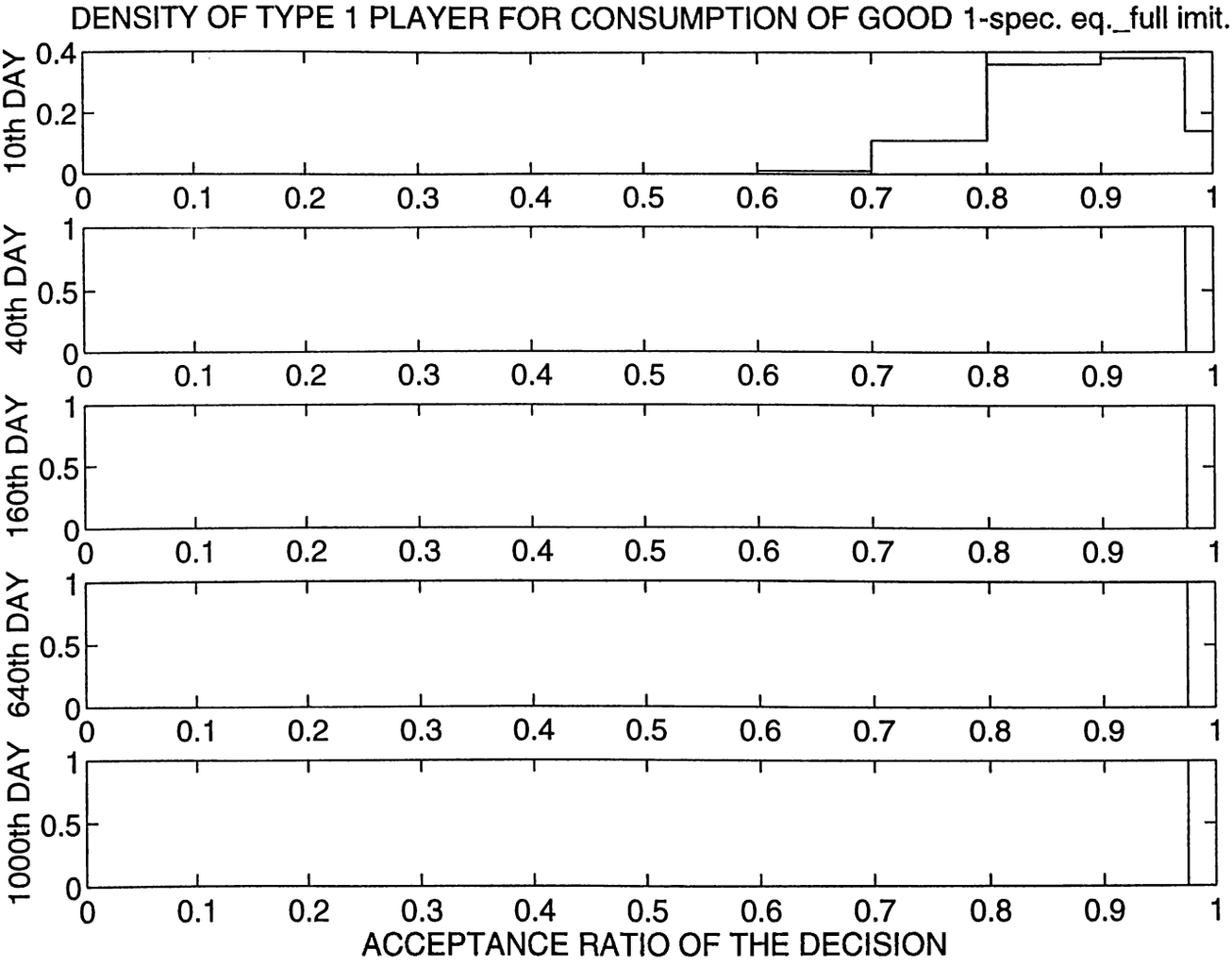

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

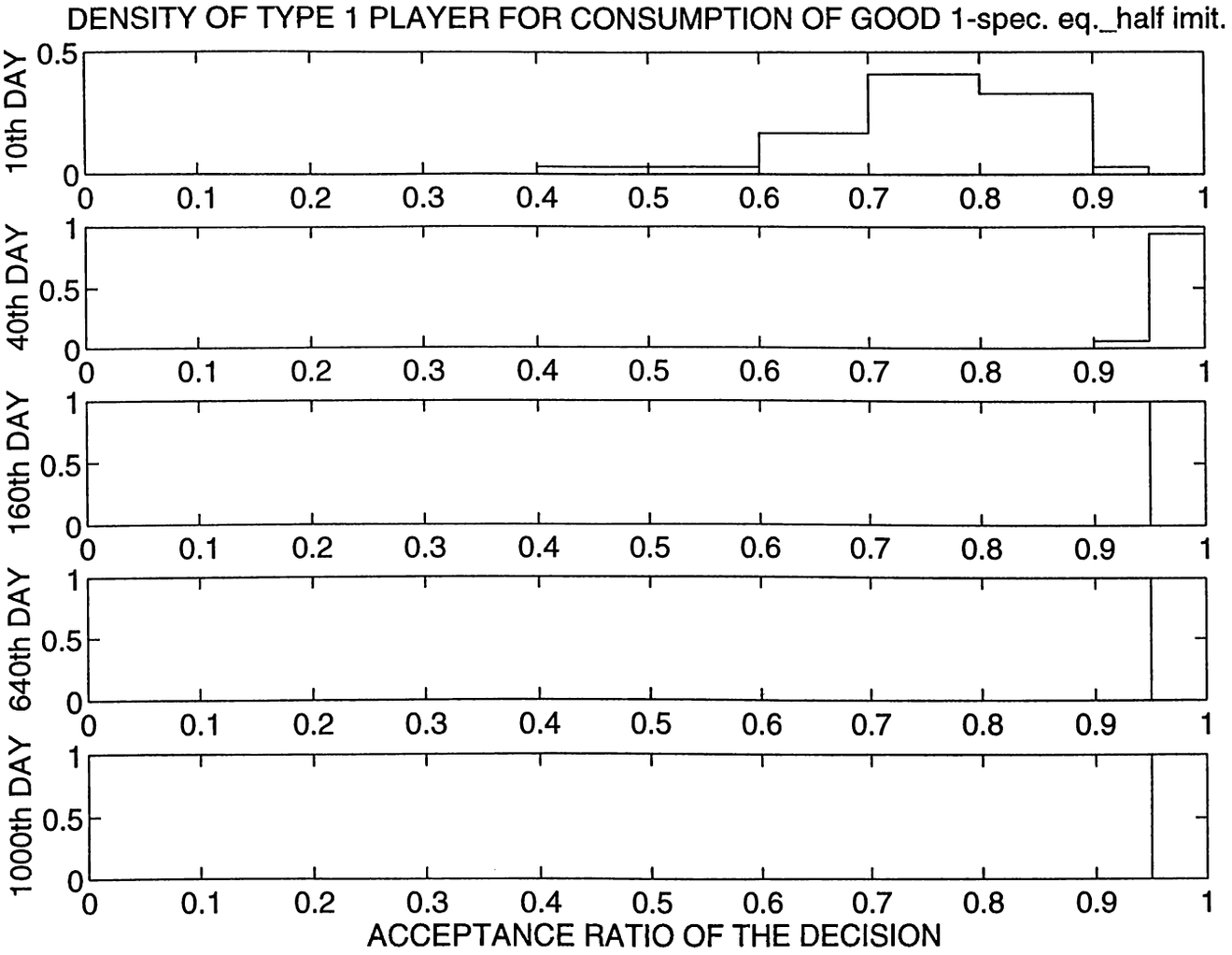

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

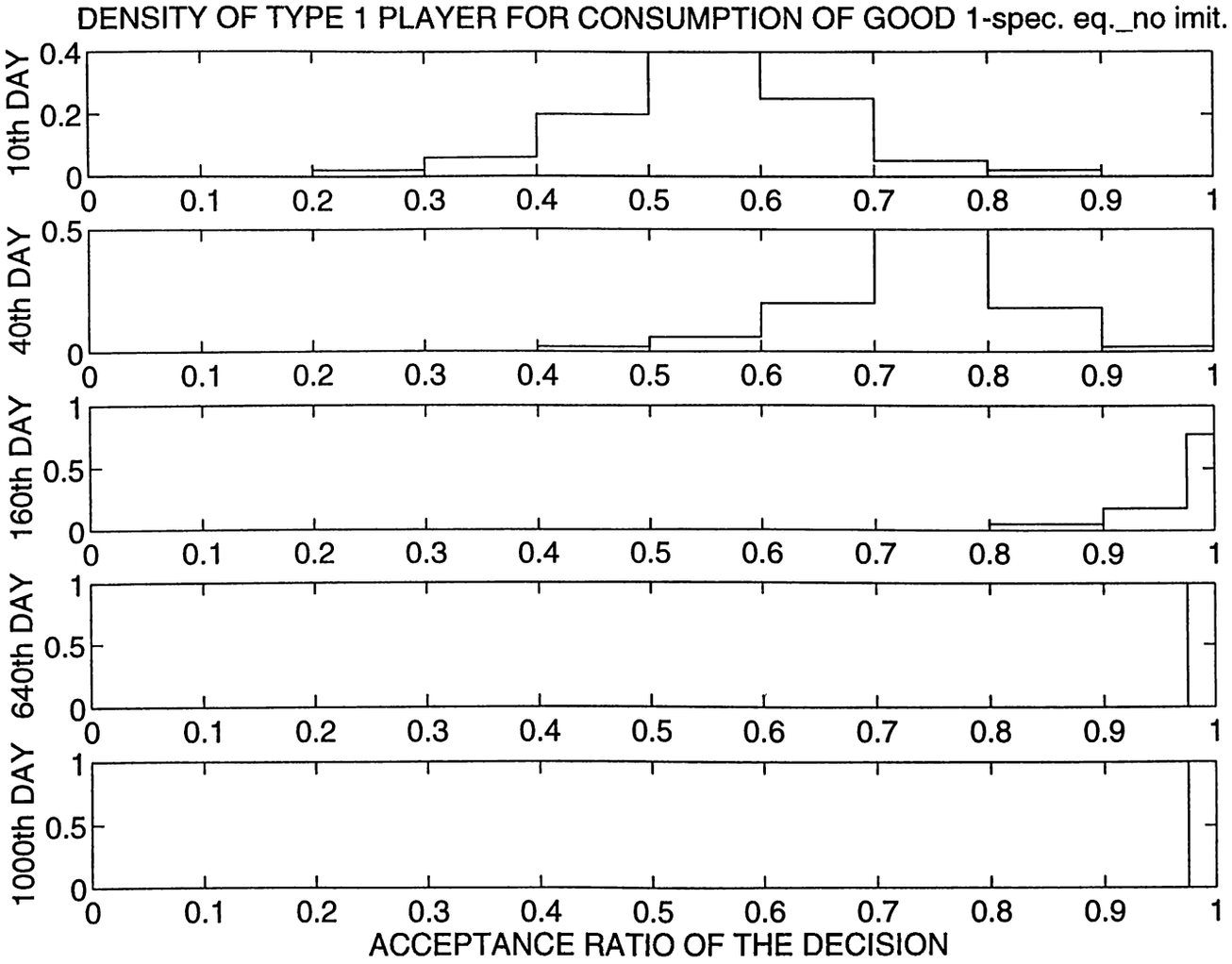

DENSITY OF TYPE 1 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

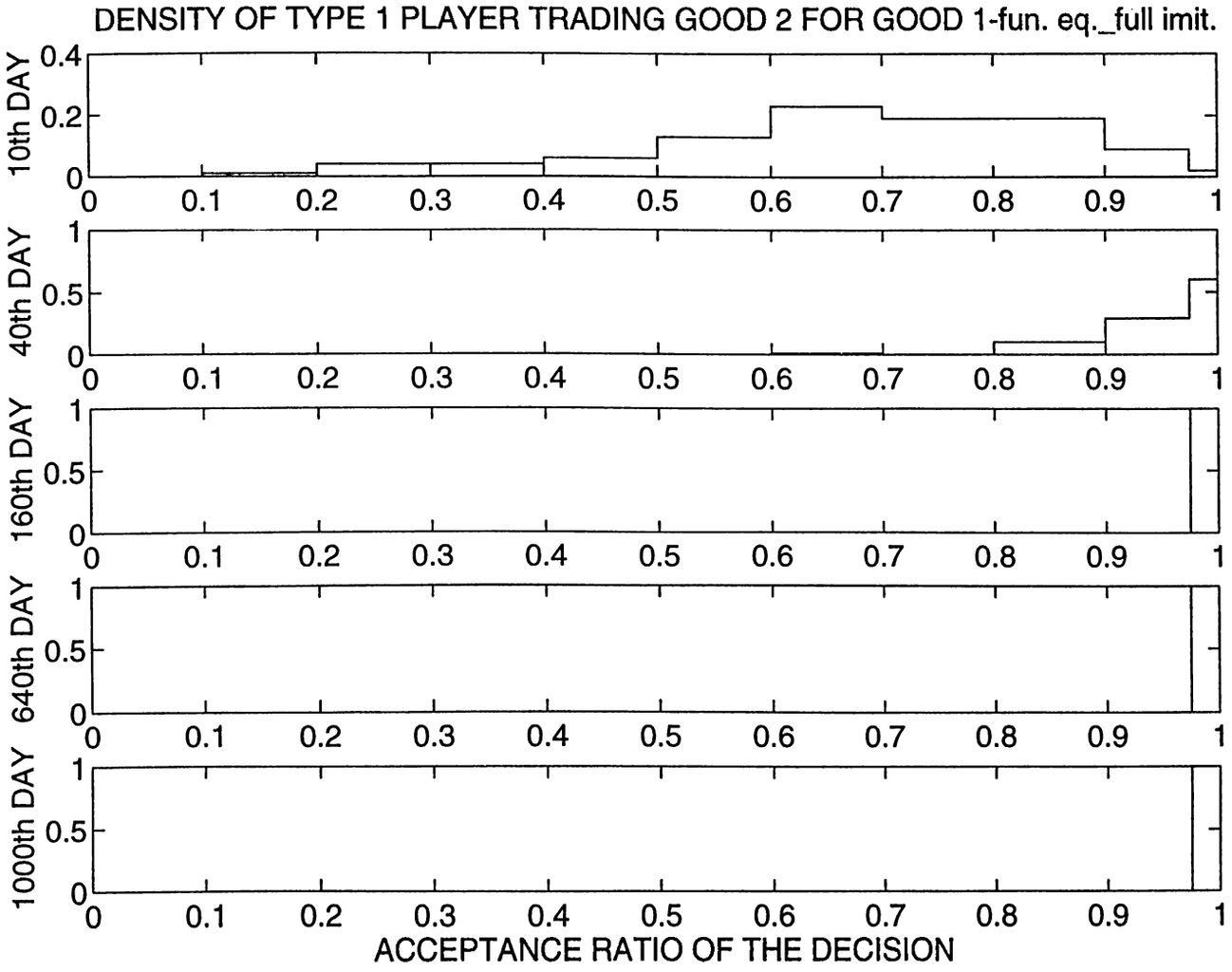

DENSITY OF TYPE 1 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

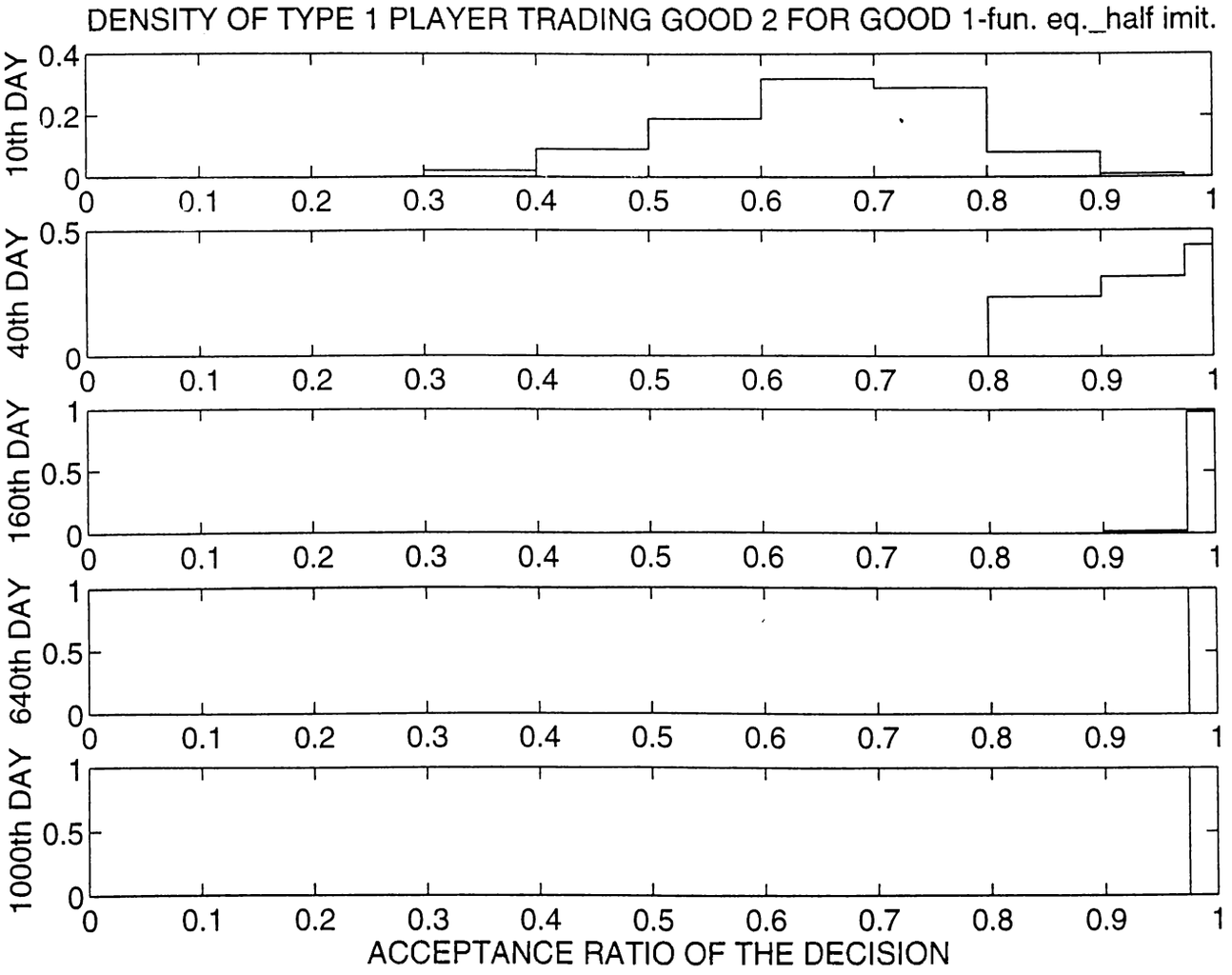

DENSITY OF TYPE 1 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

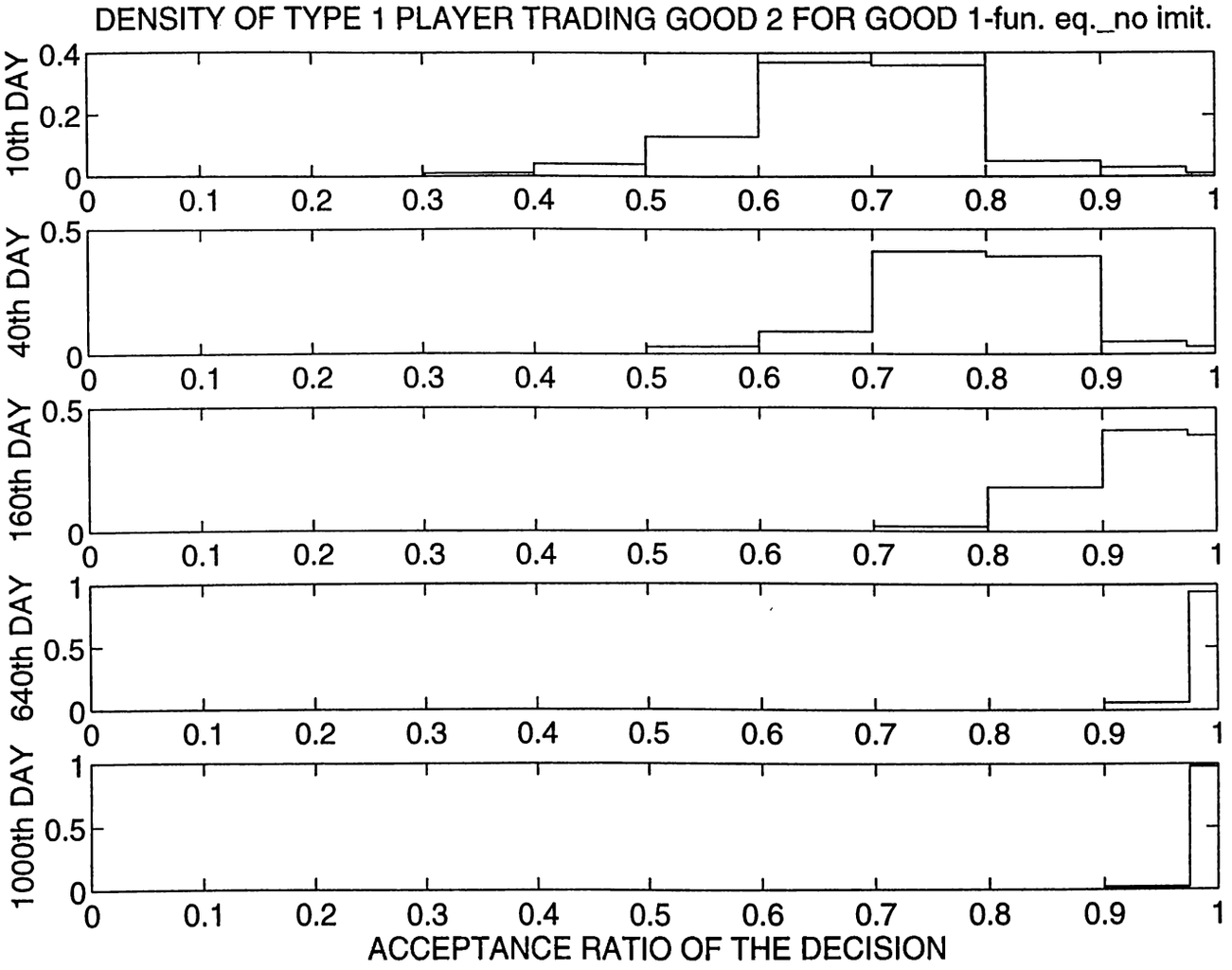

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 1-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

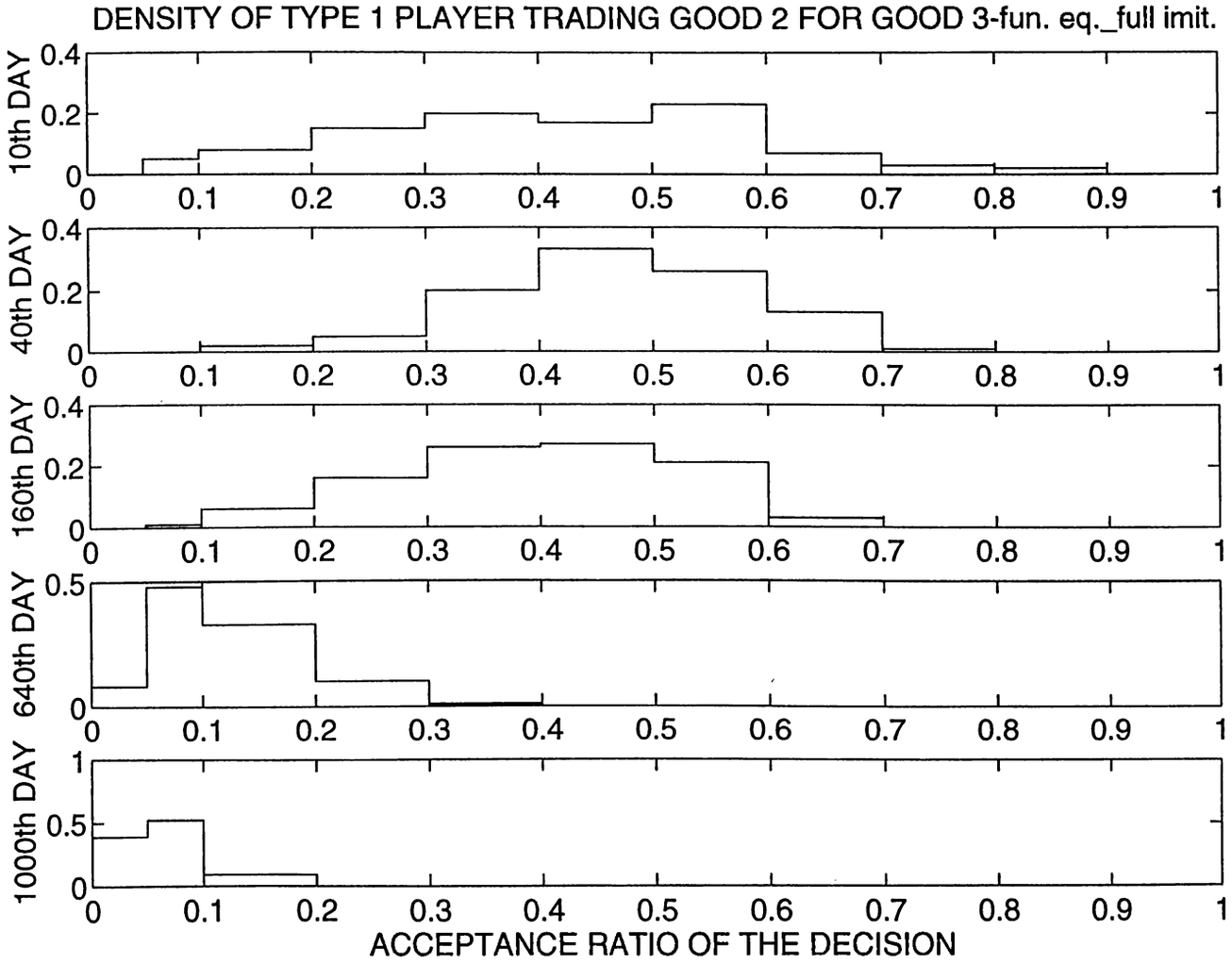

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 1-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

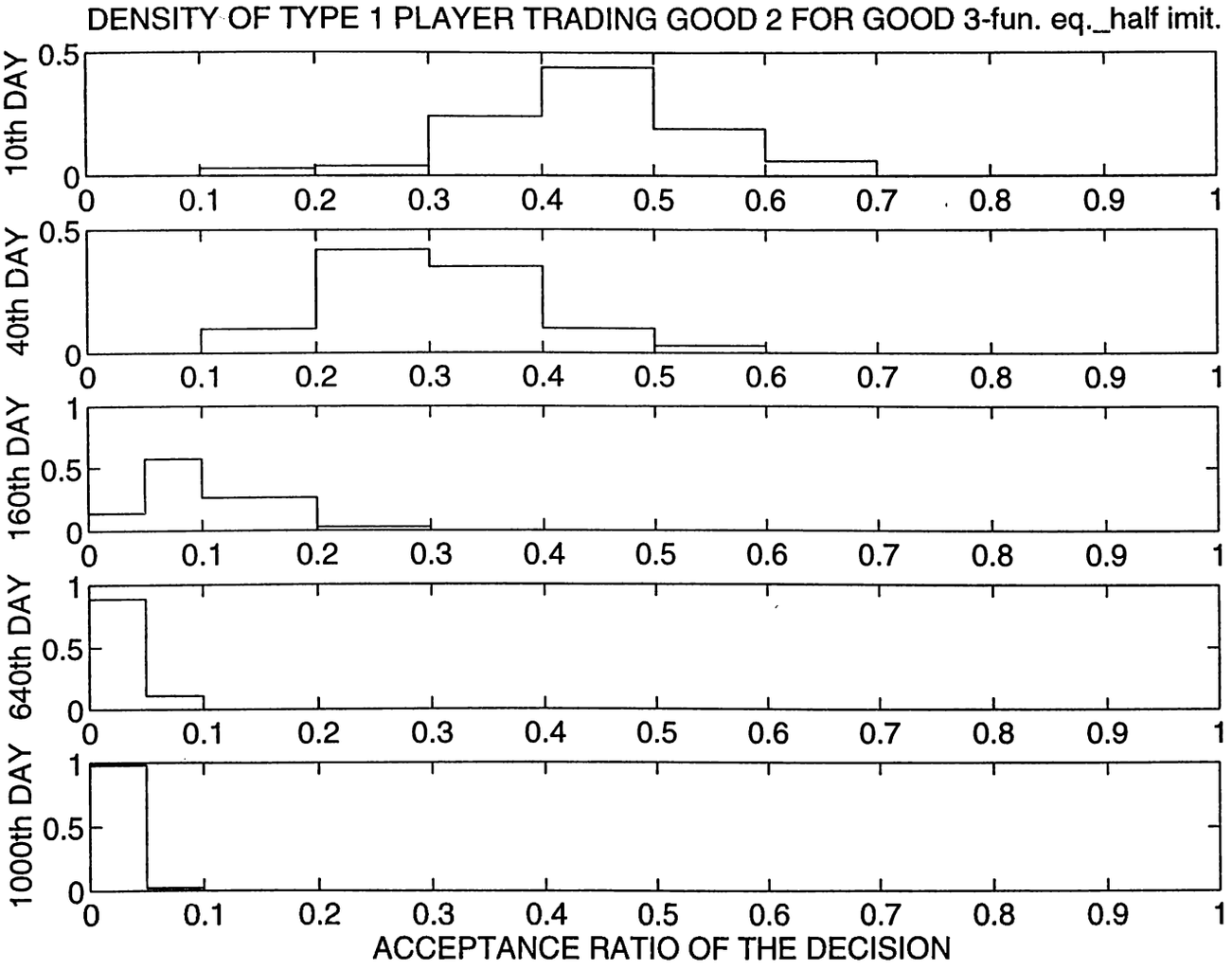

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 1-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

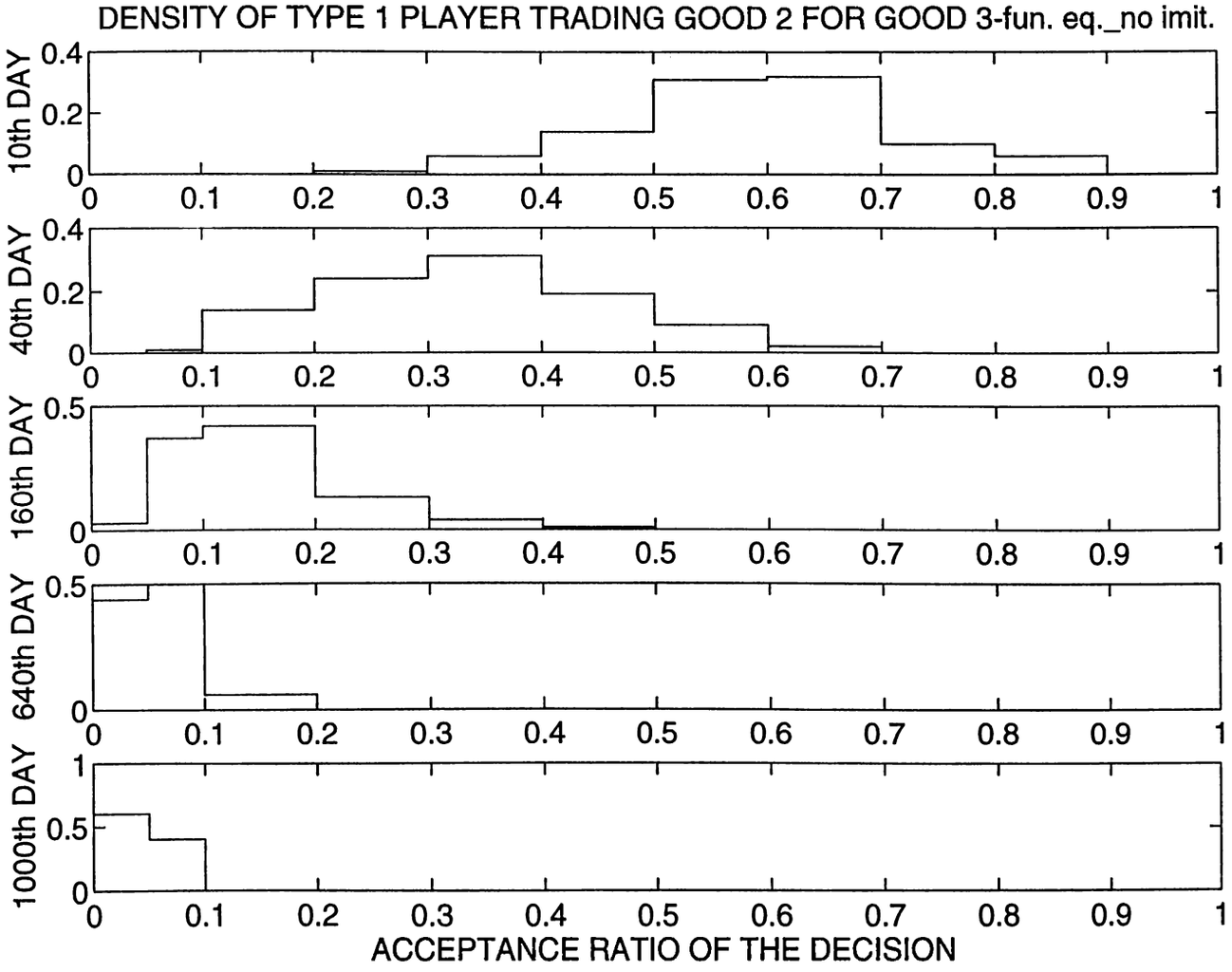

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

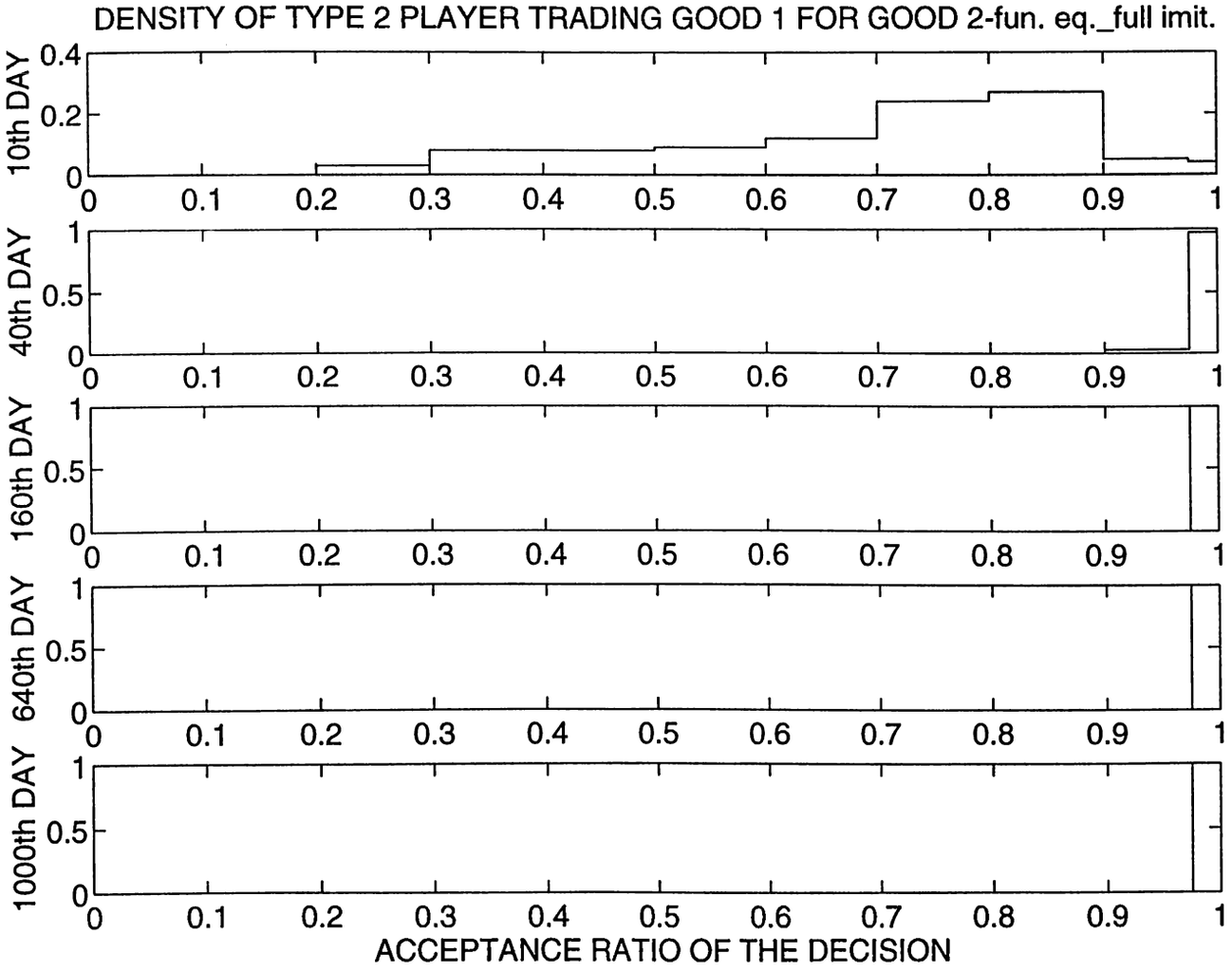

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

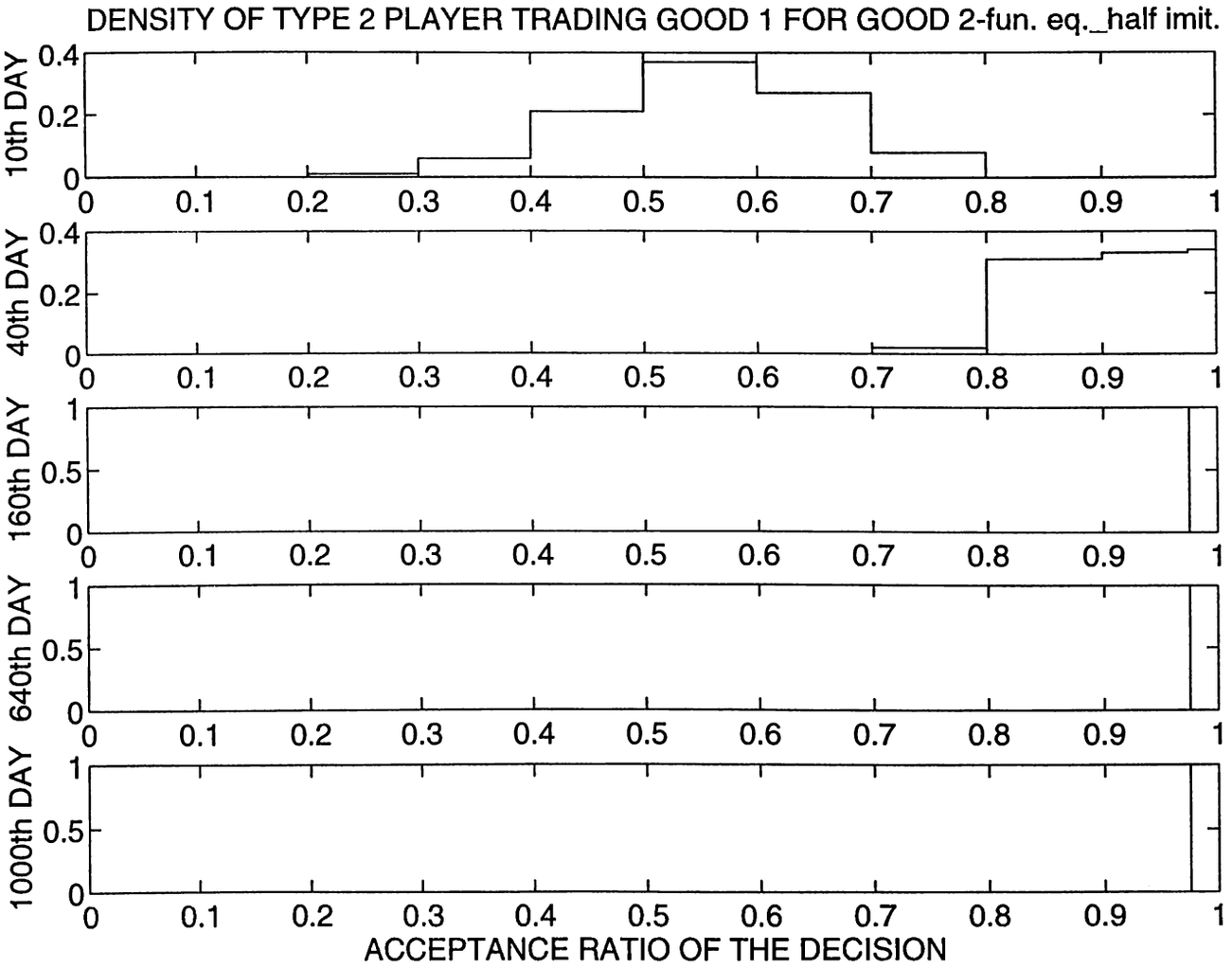

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

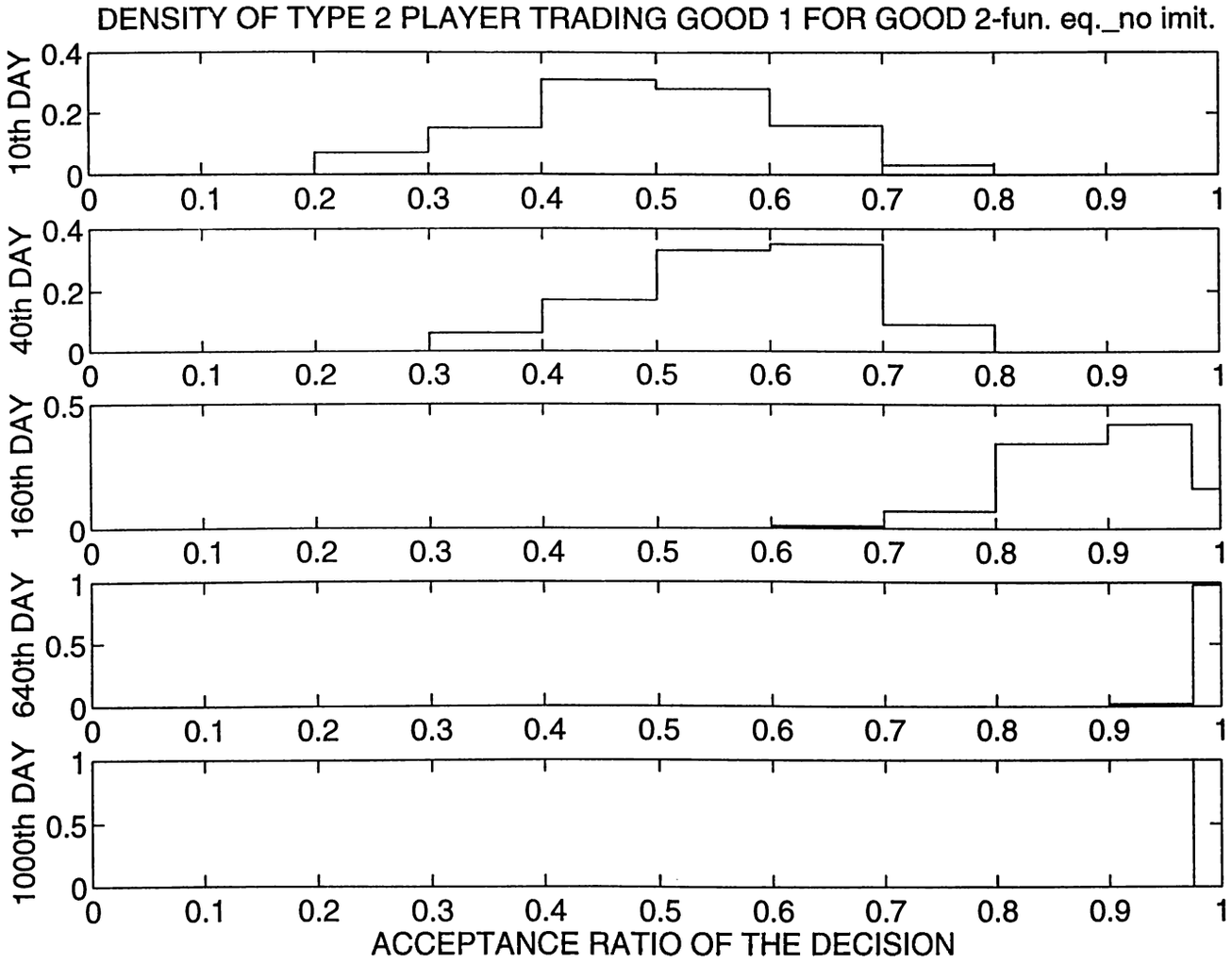

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
0.5
1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

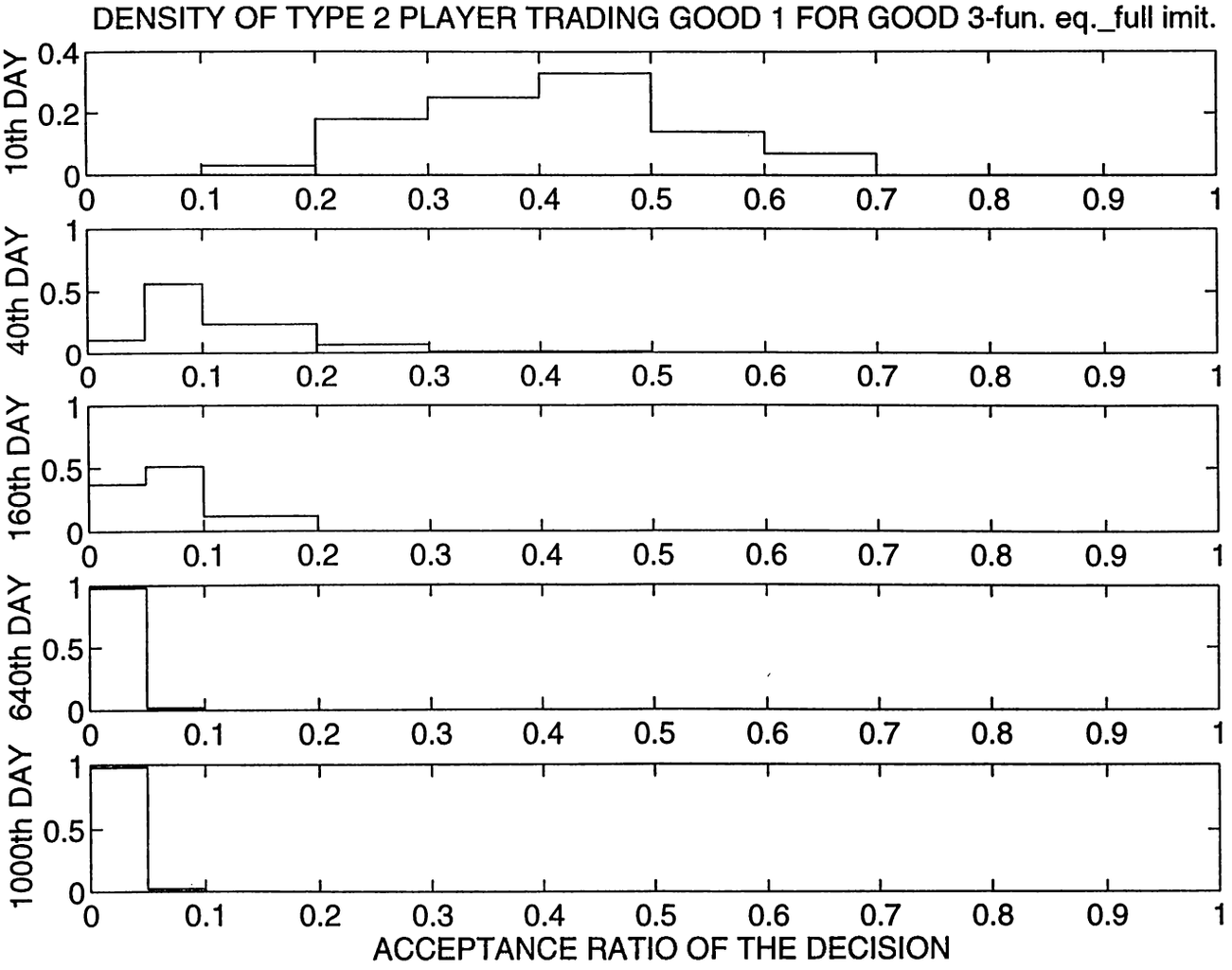

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

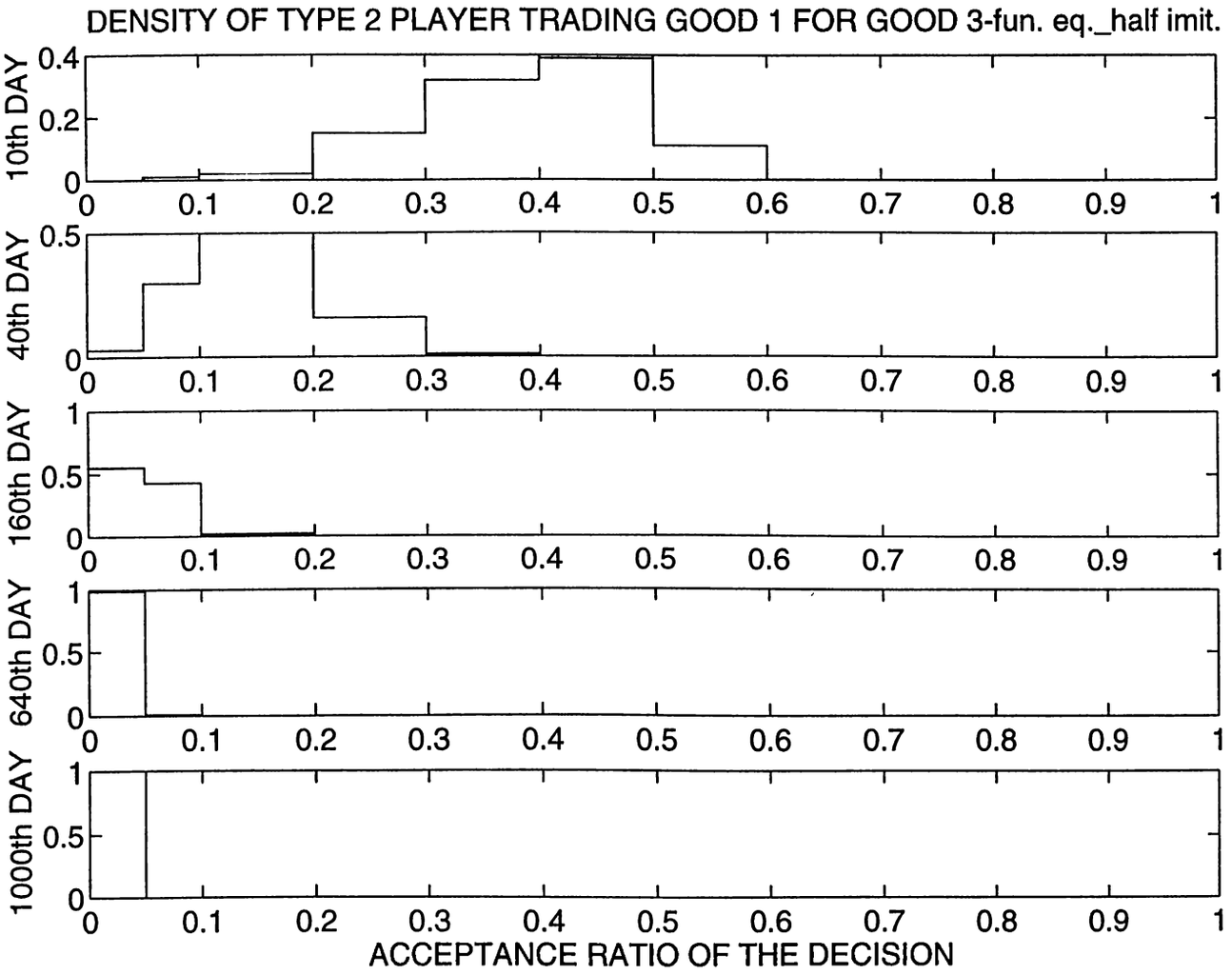

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

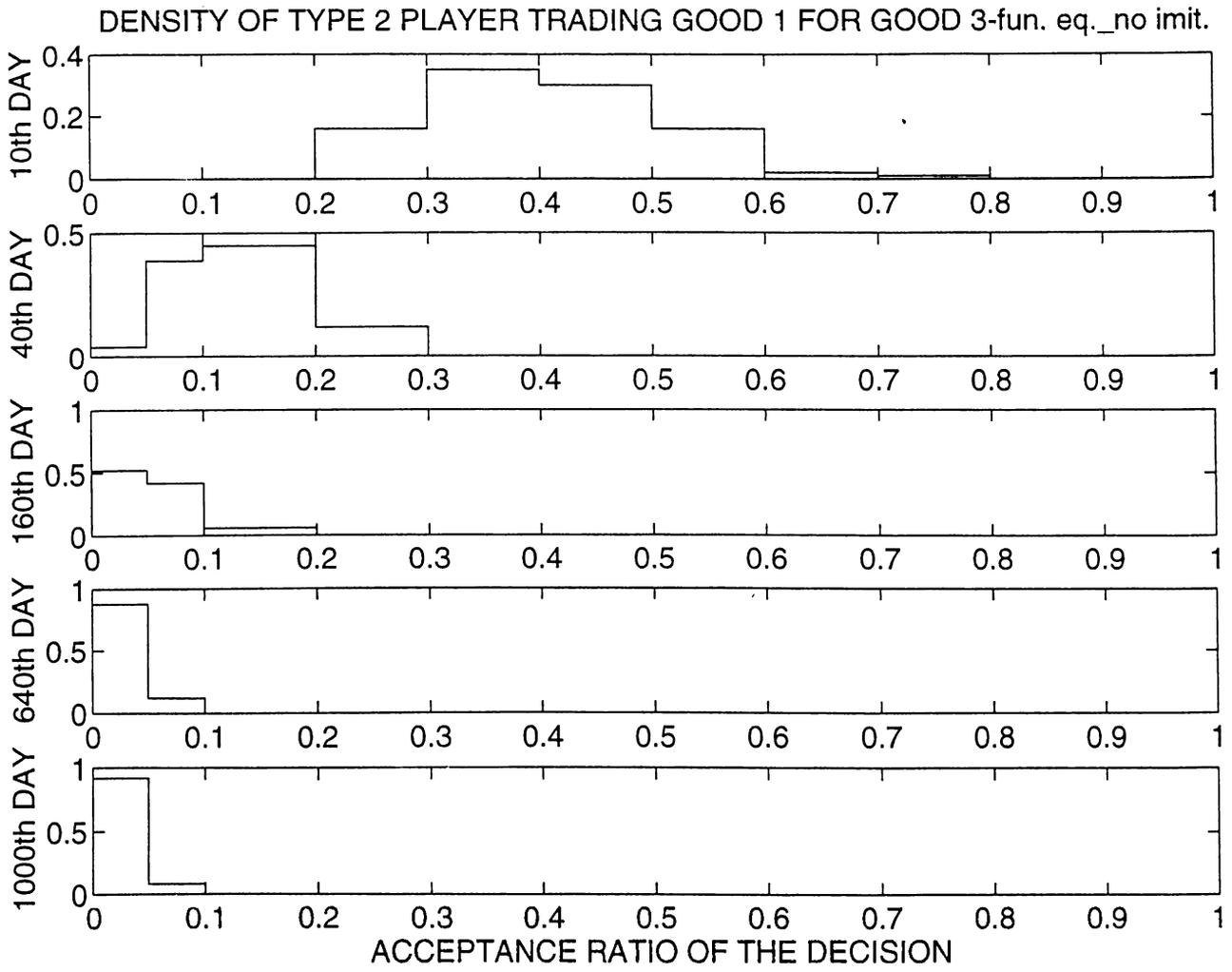

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 2-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

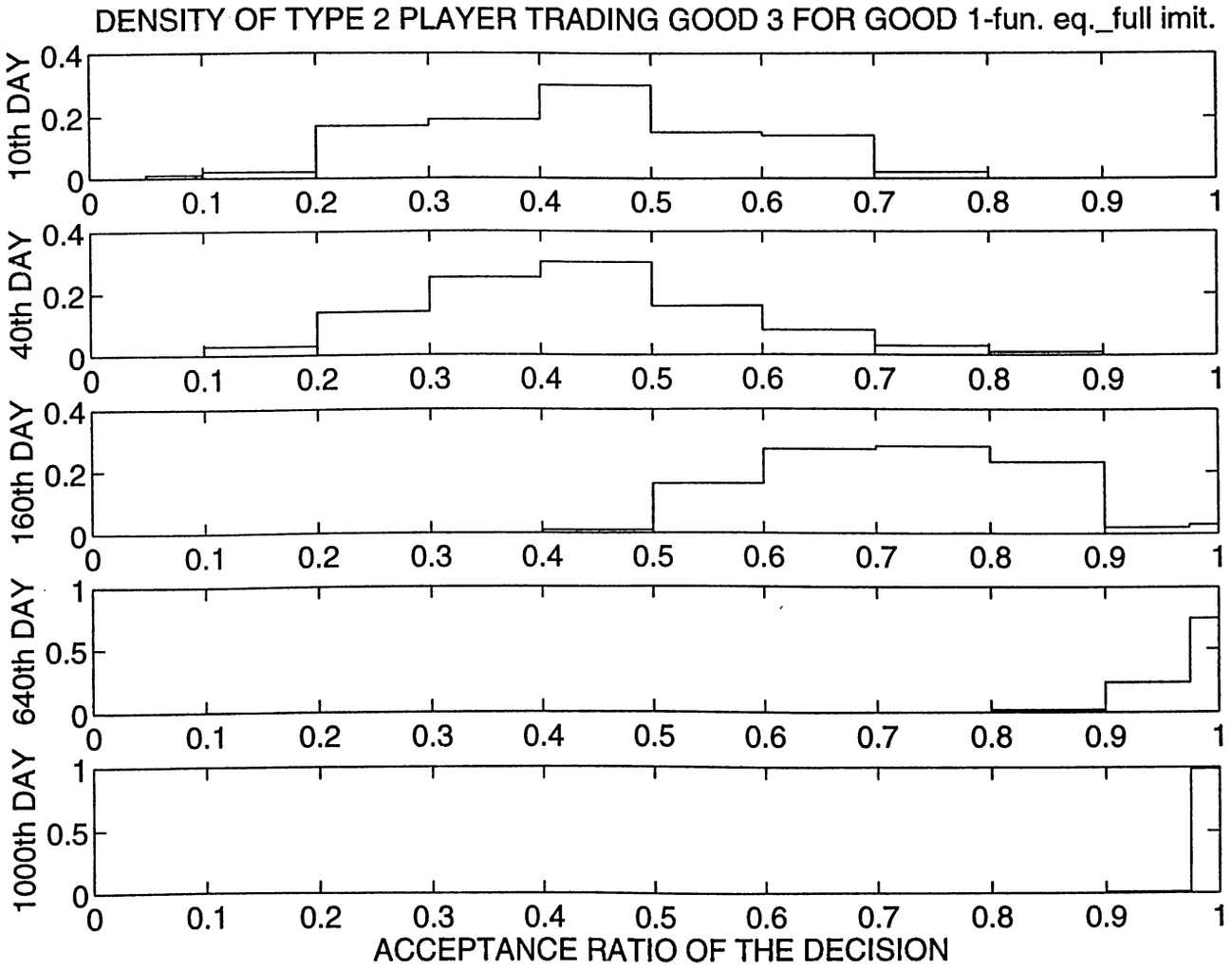

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 2-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

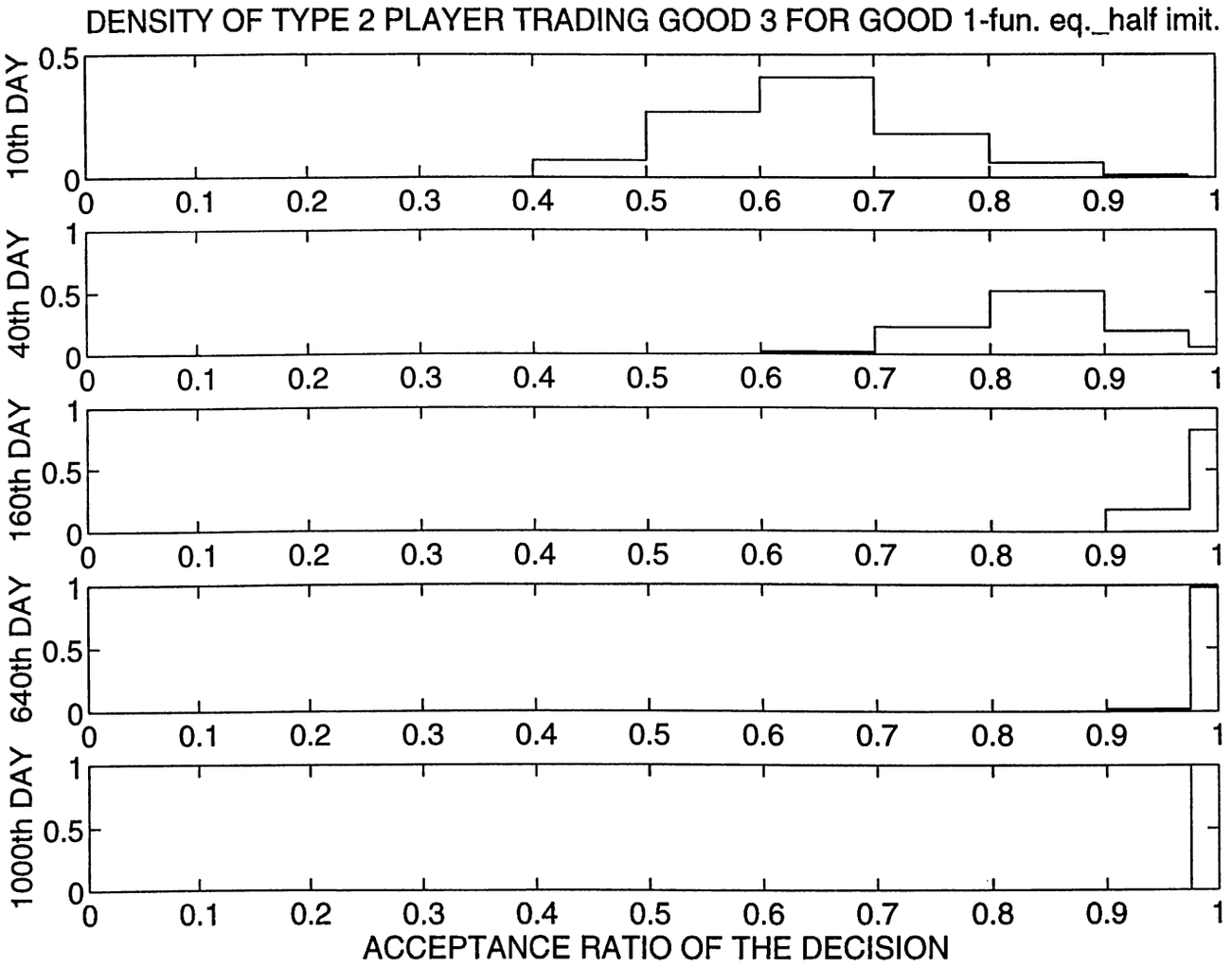

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 2-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

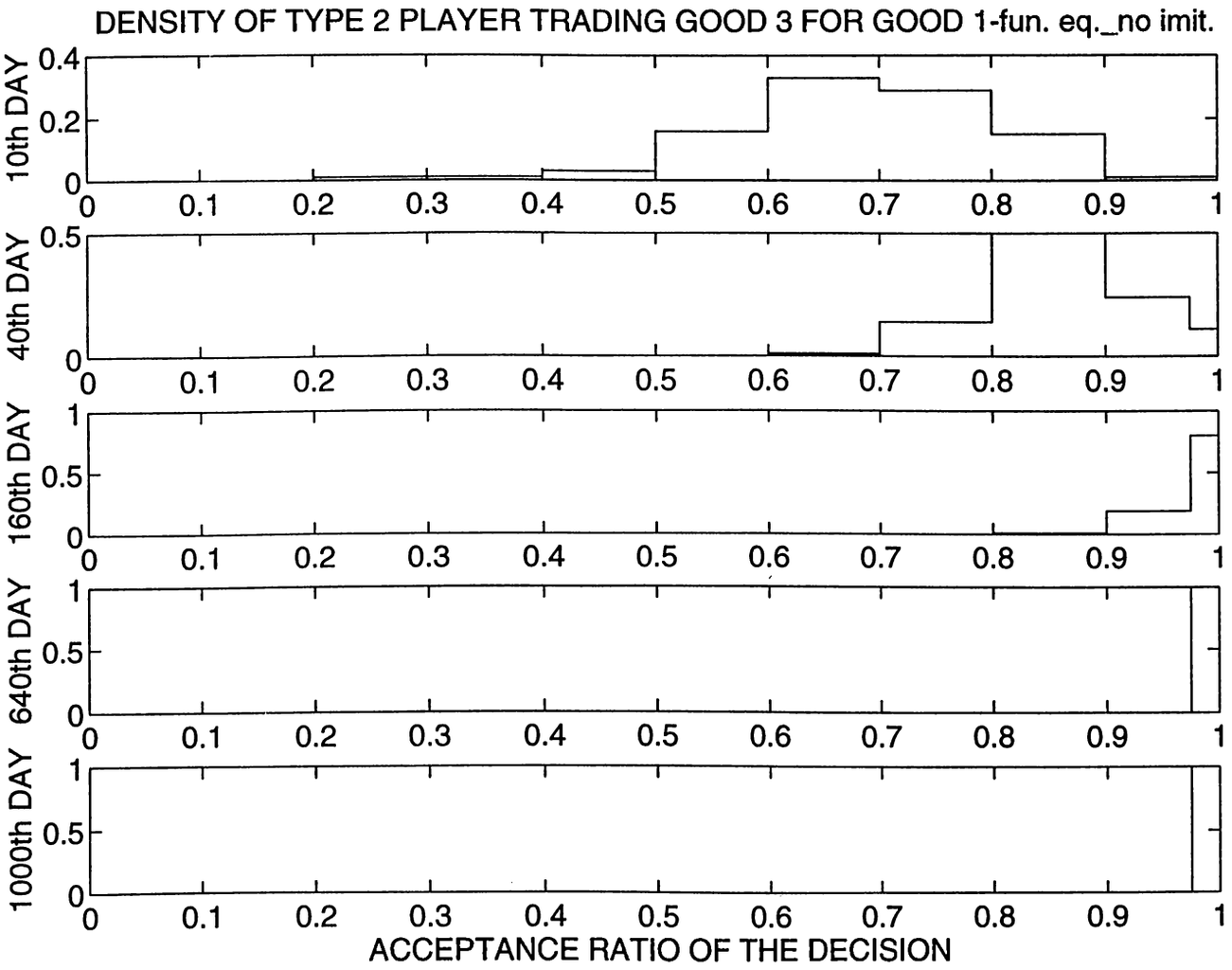

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 3-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

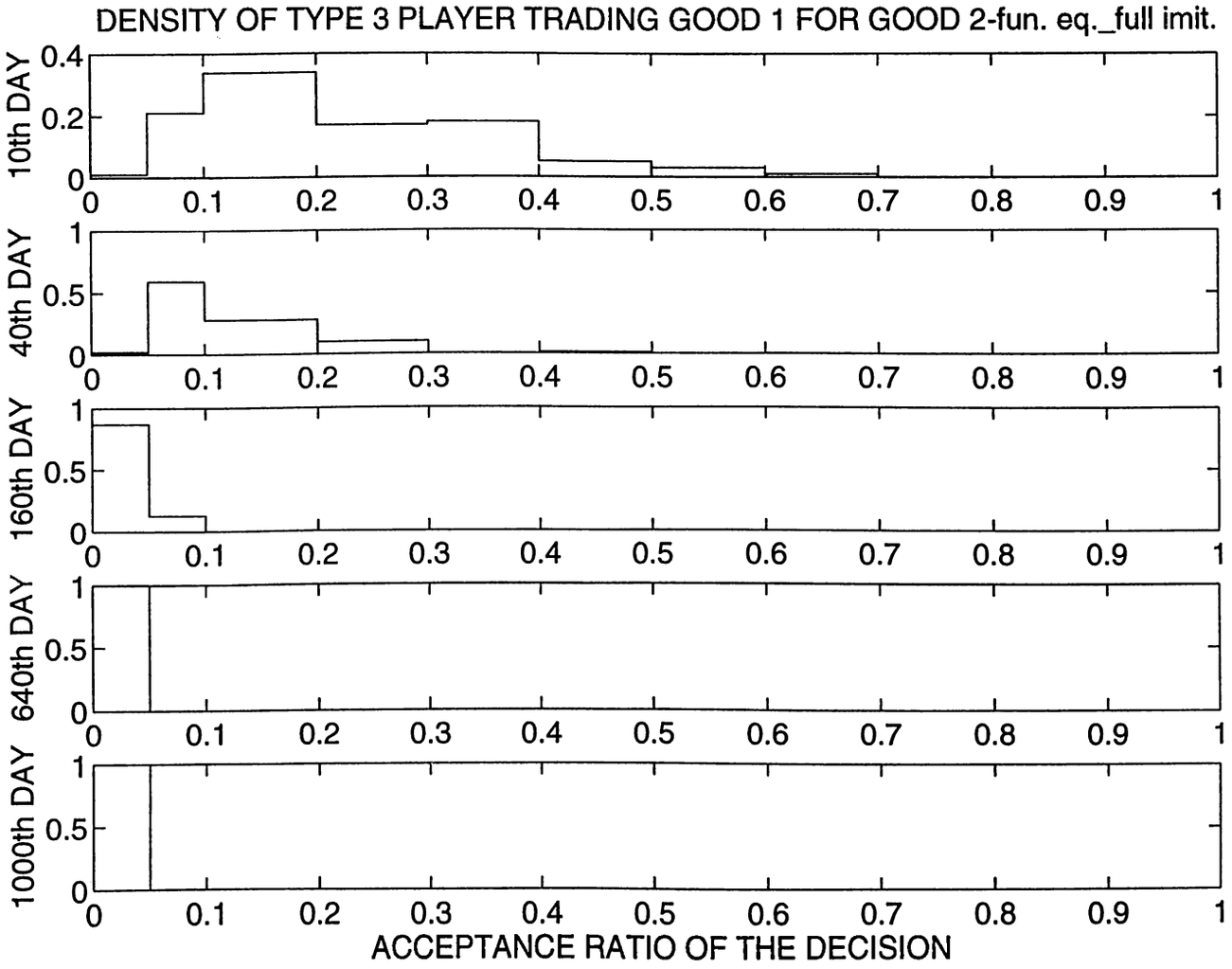

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 3-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

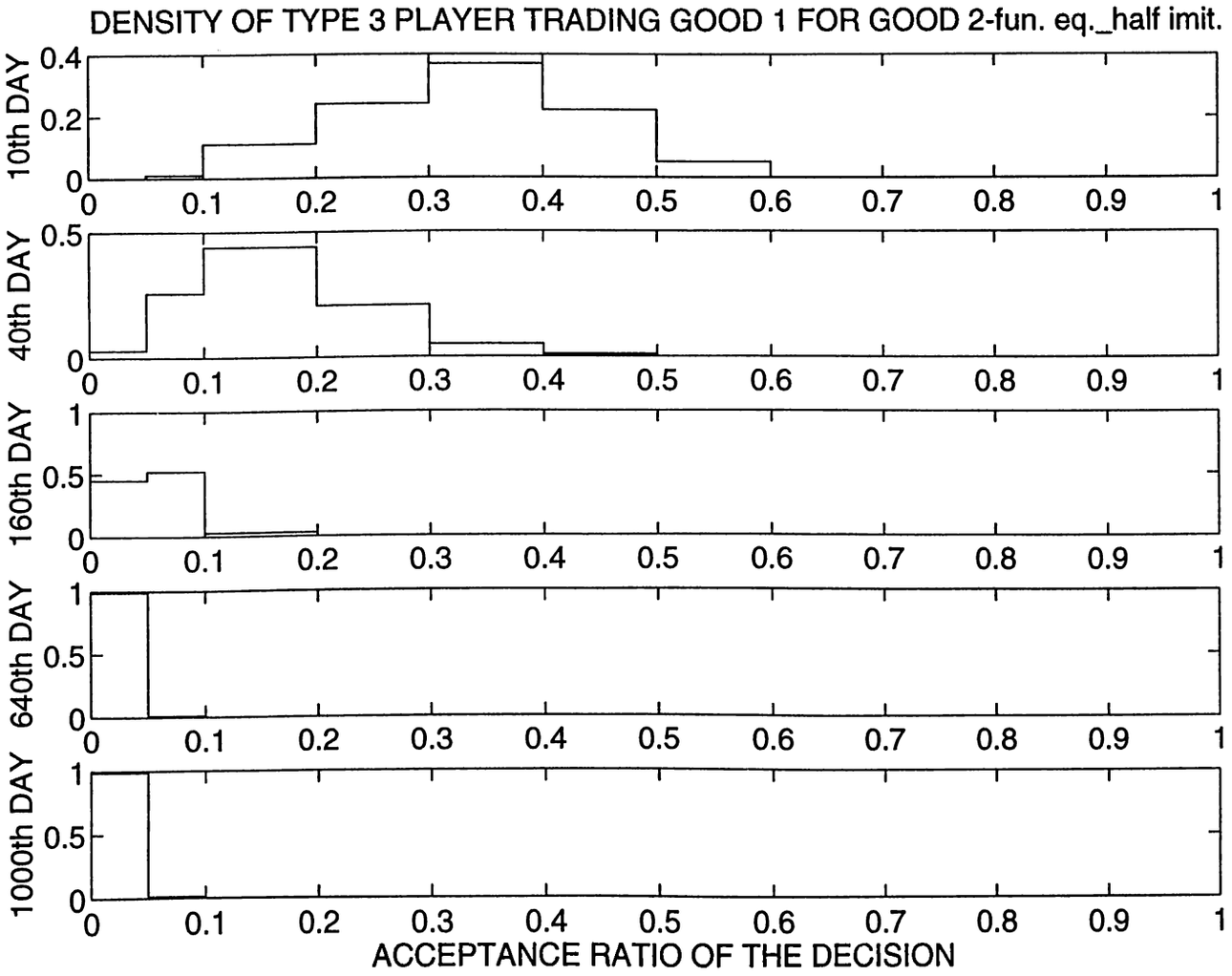

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 3-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

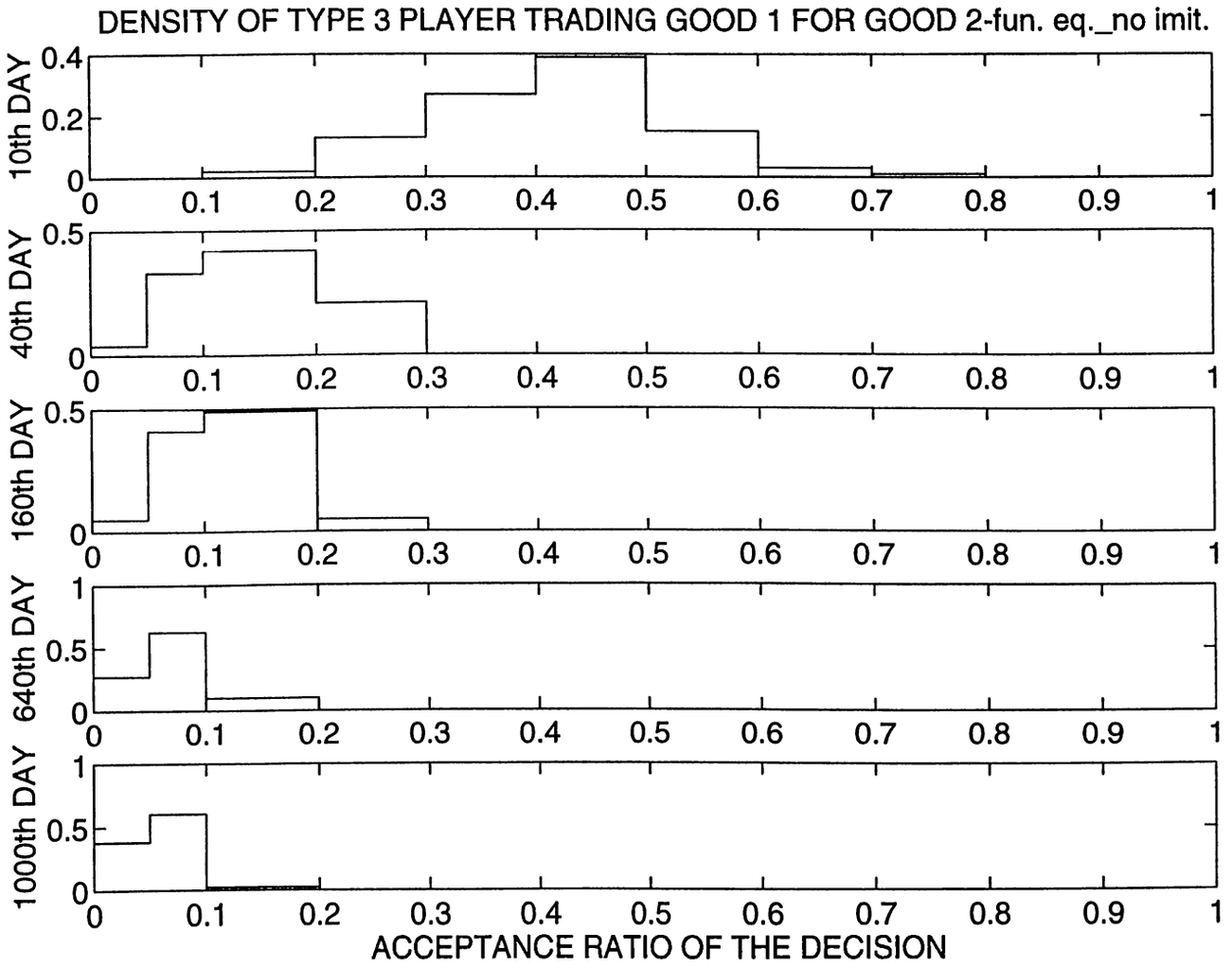

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 1-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

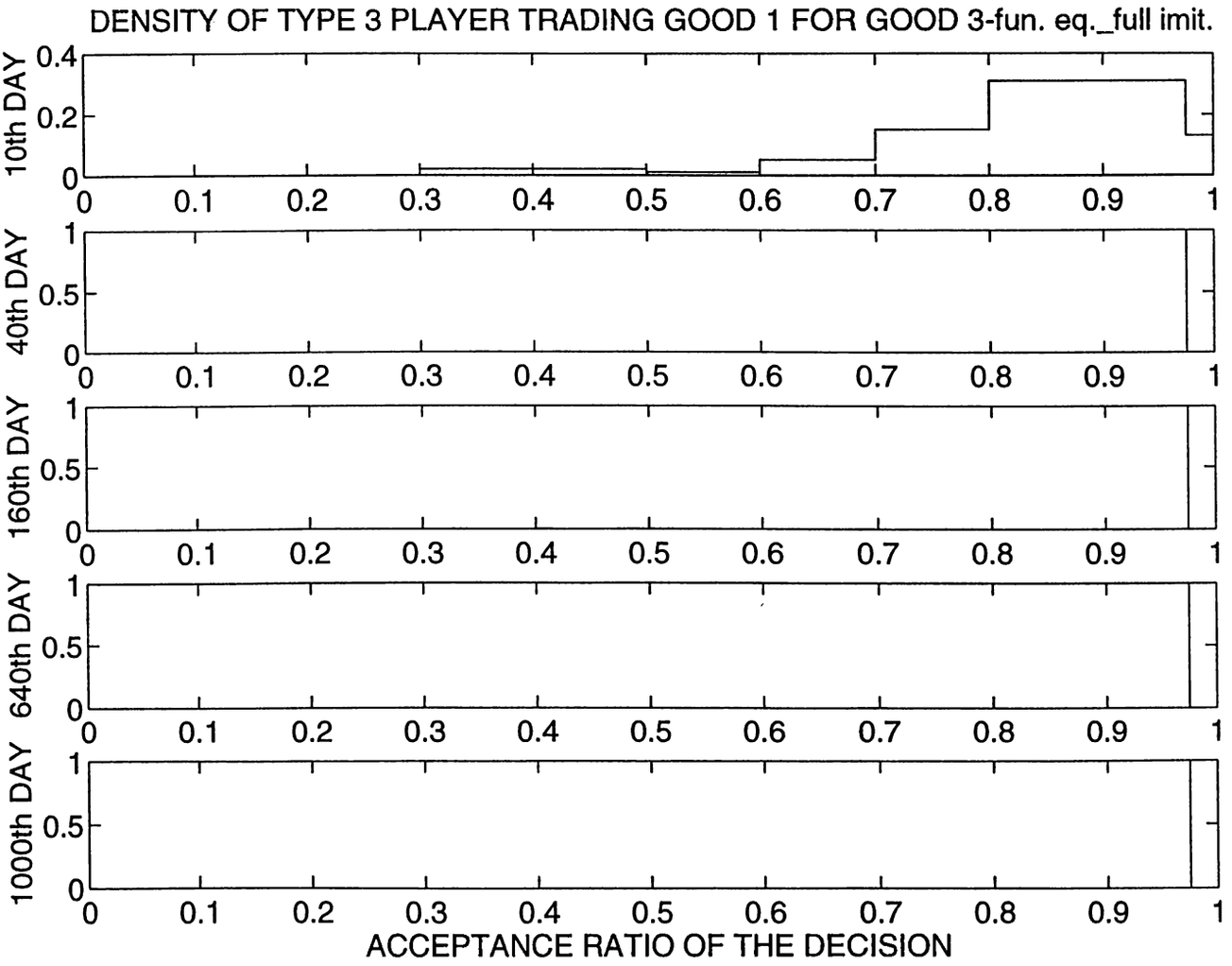

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 1-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

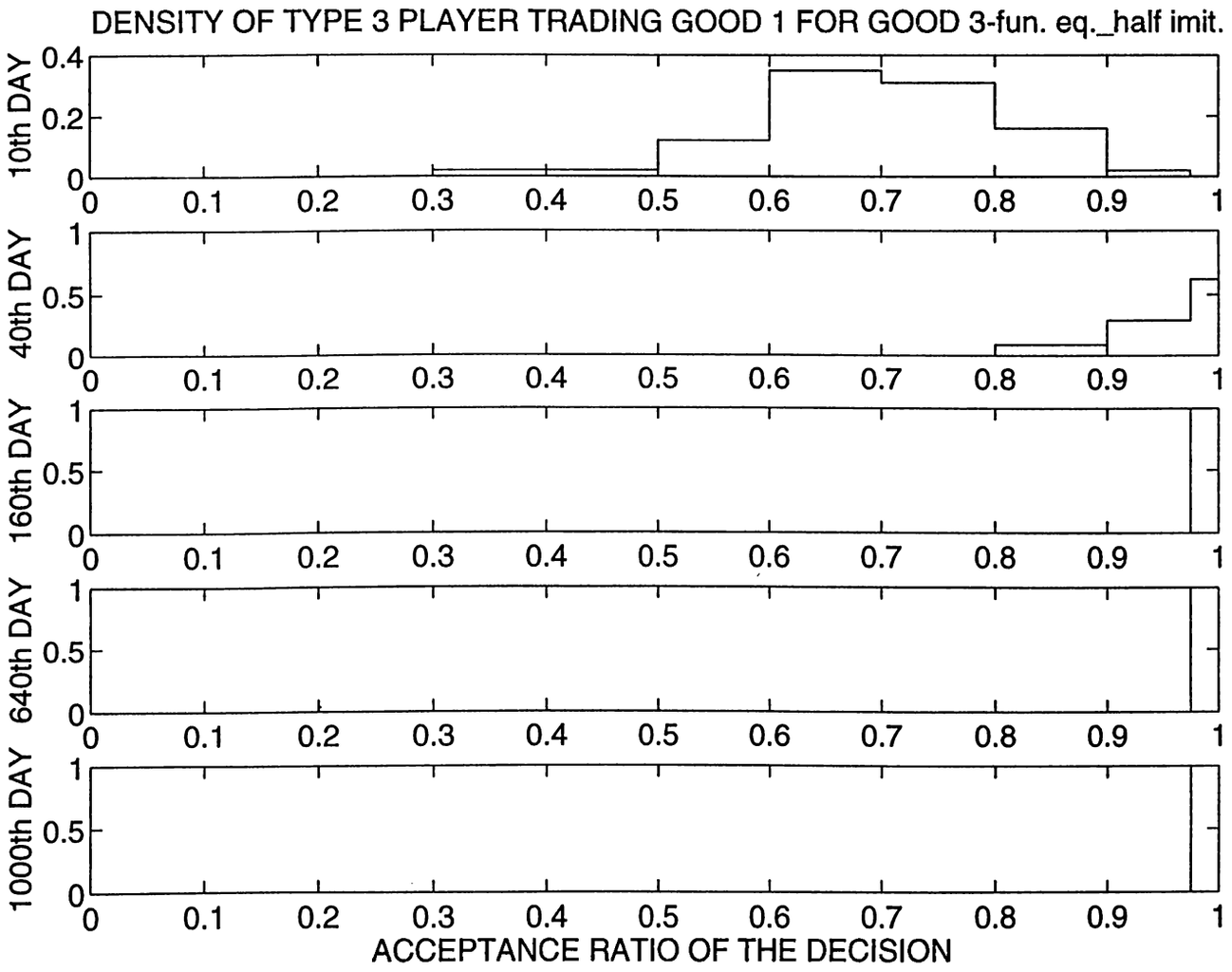

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 1-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

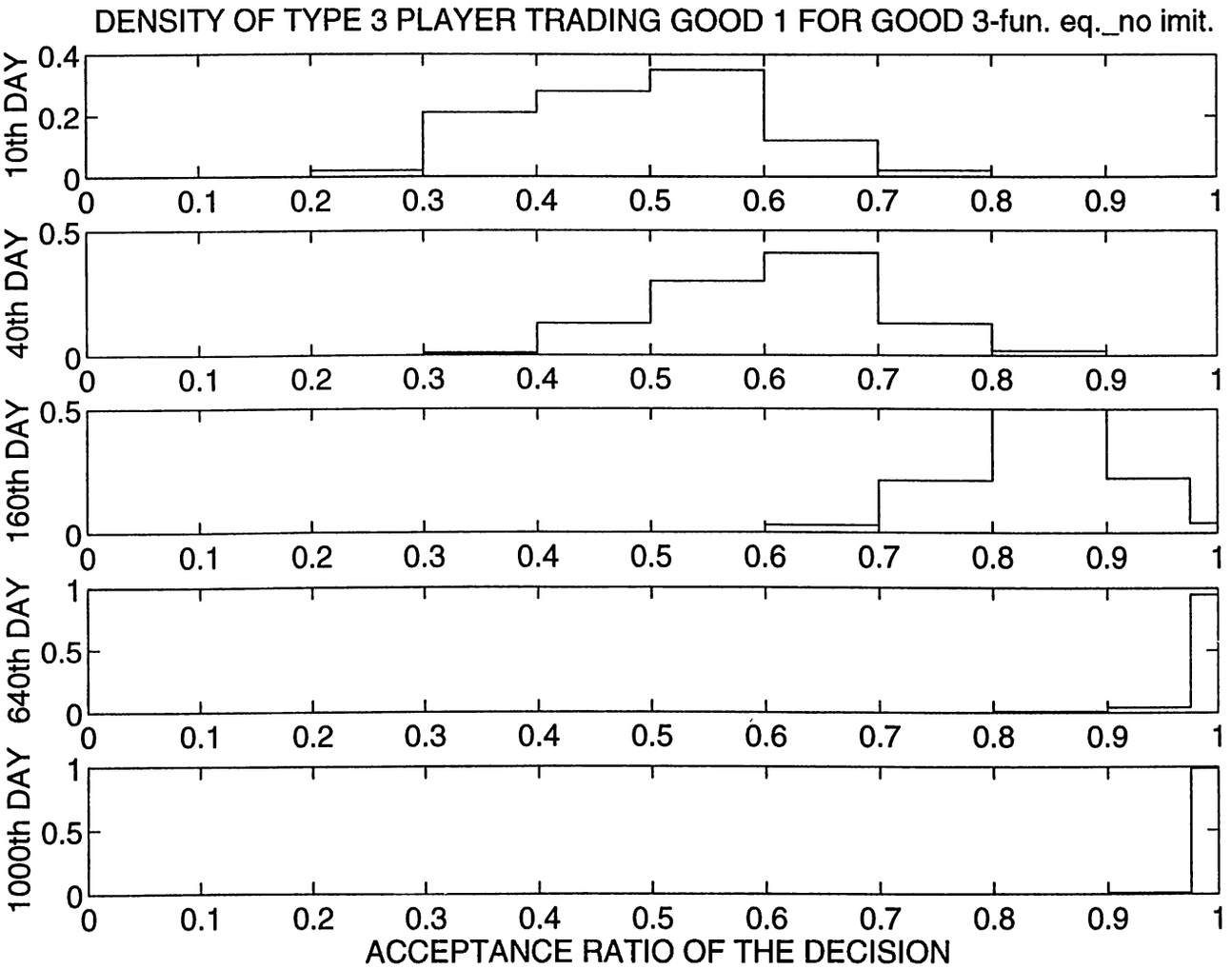

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 2-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

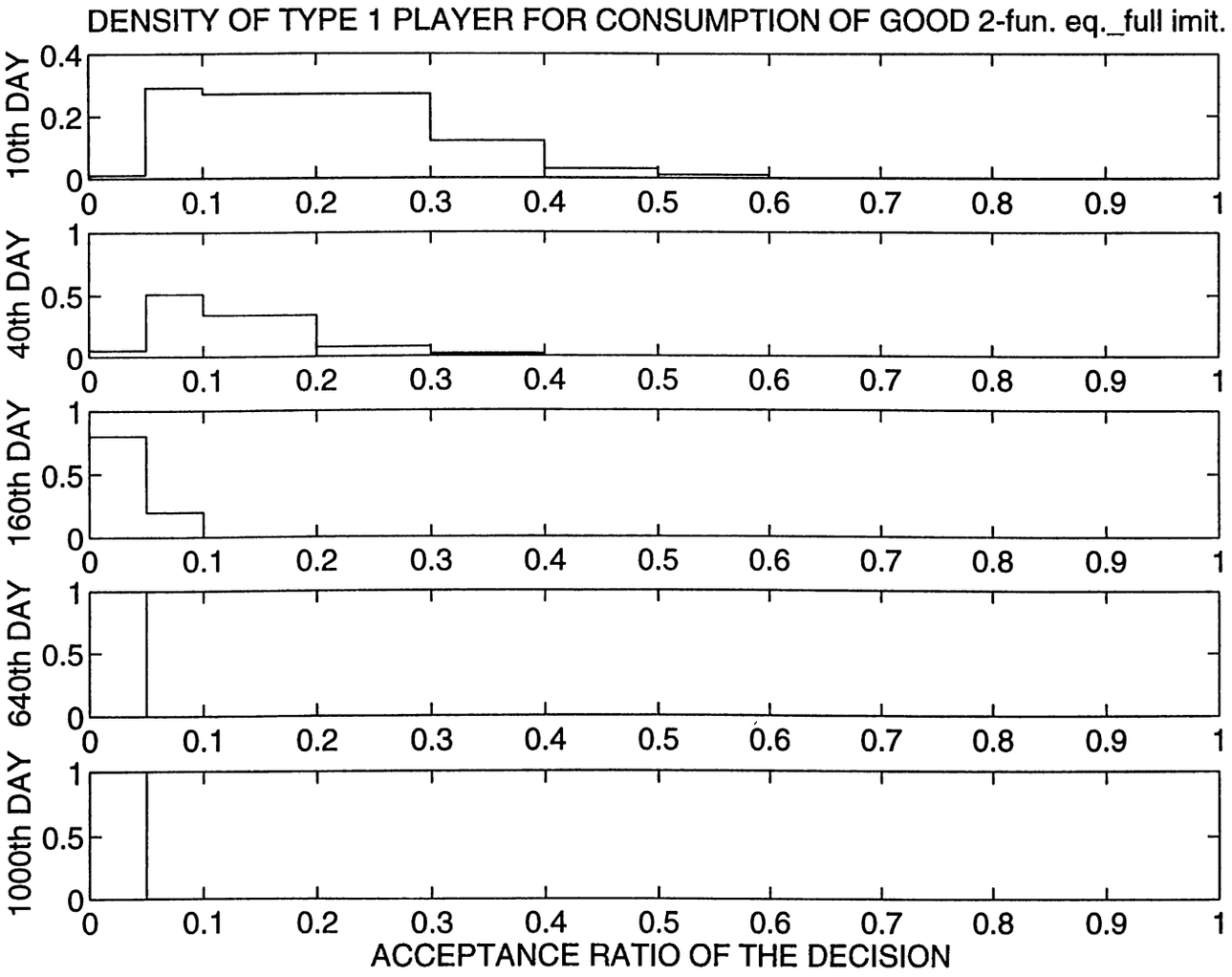

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 2-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

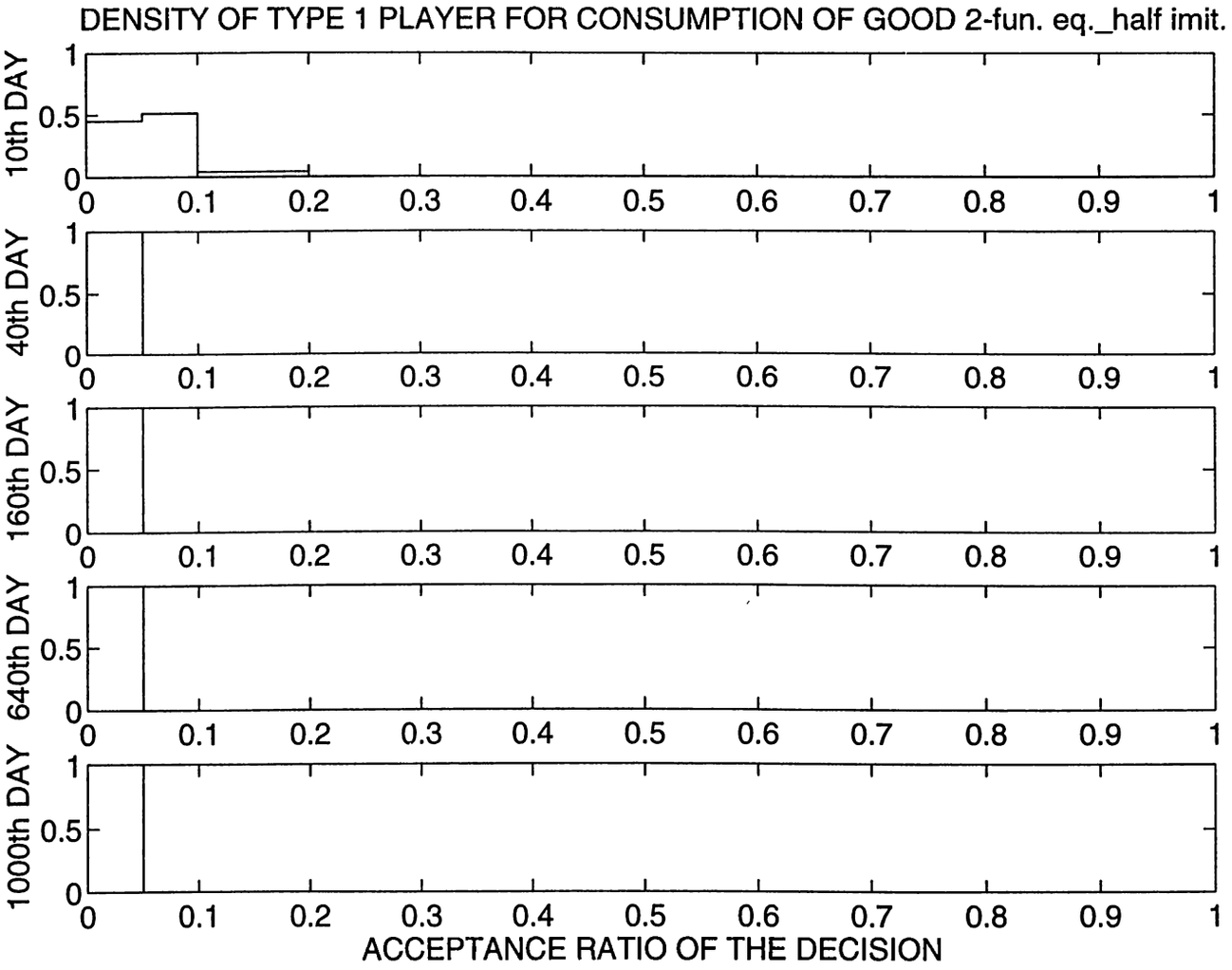

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 2-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

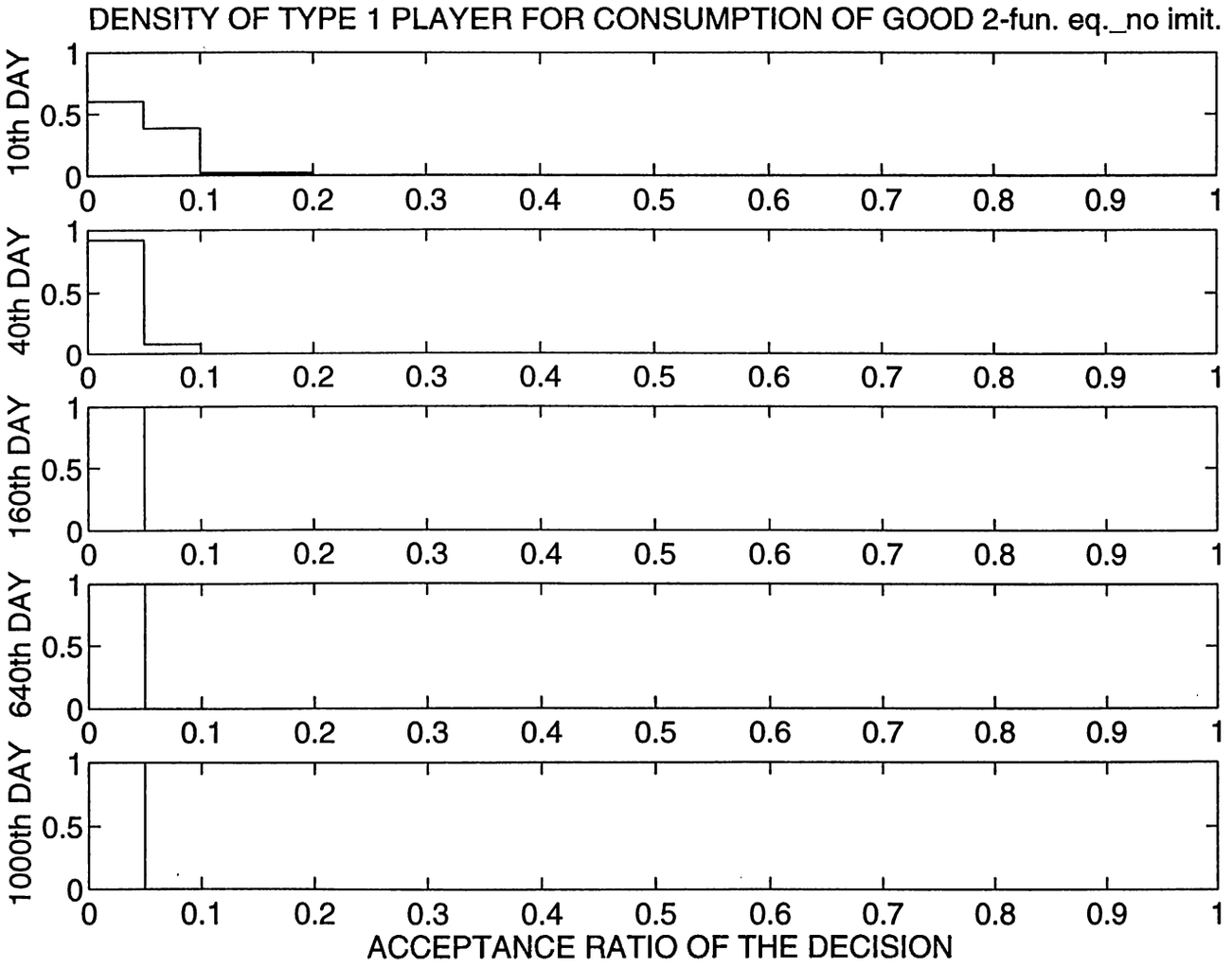

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 3-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

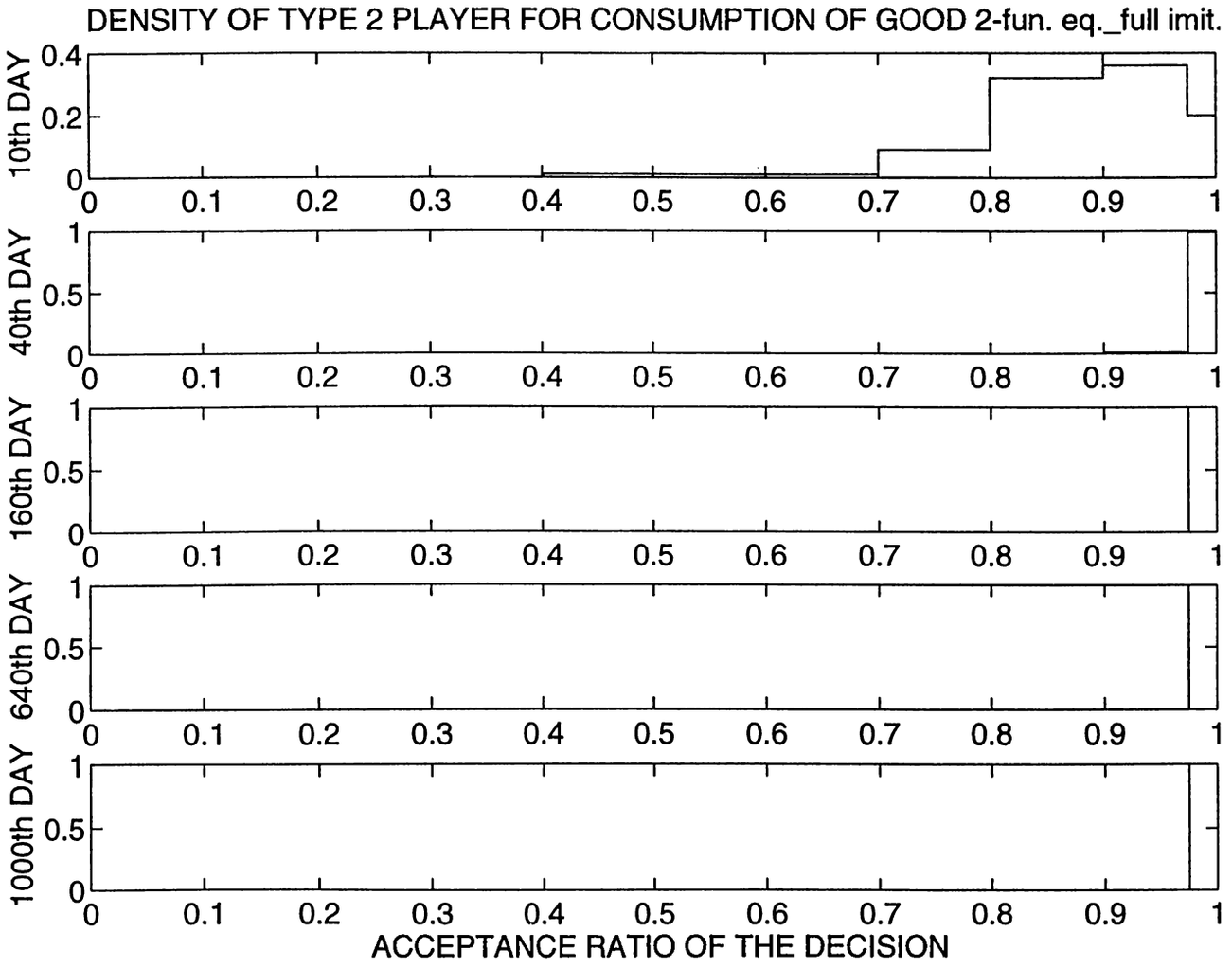

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 3-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

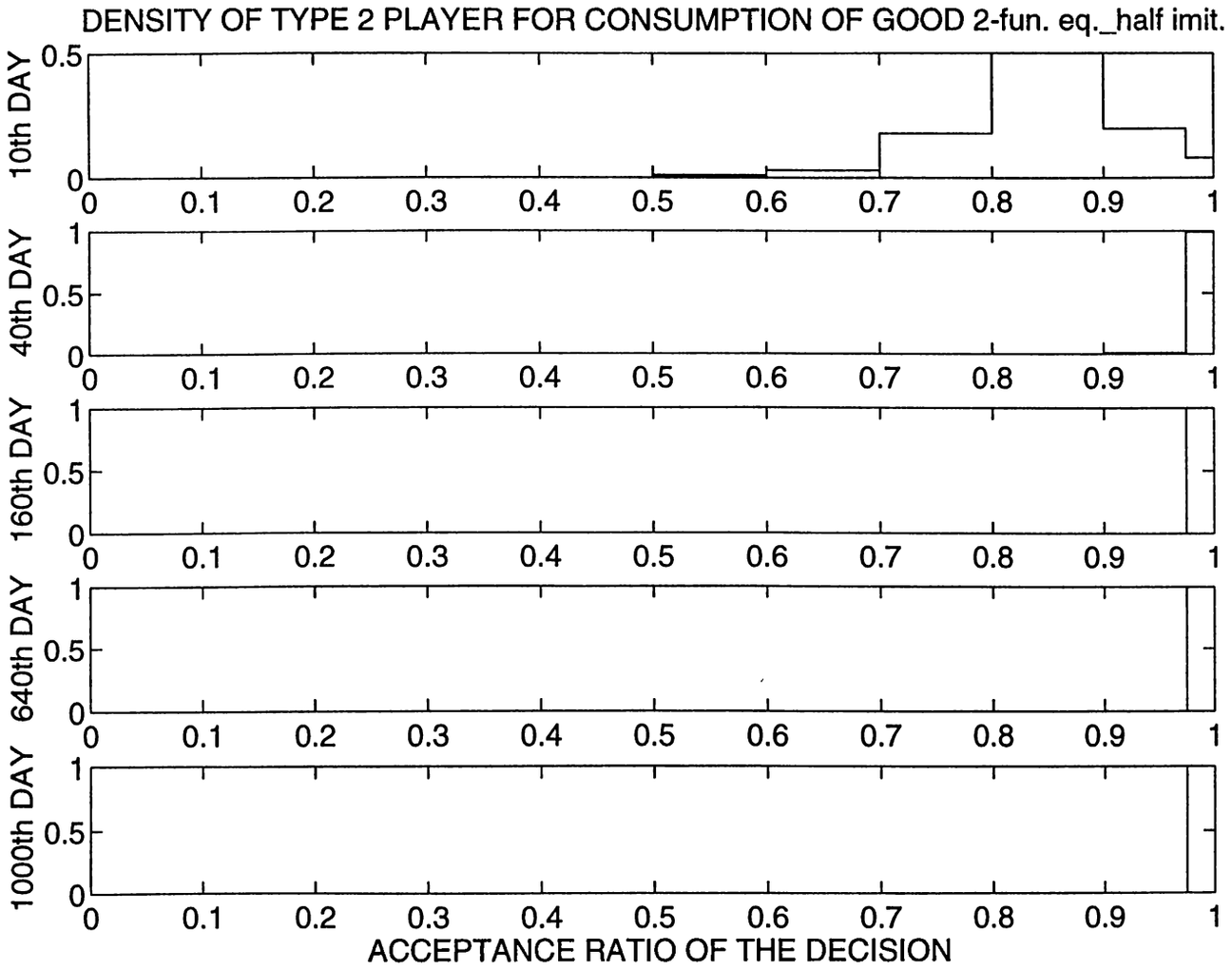

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 3-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

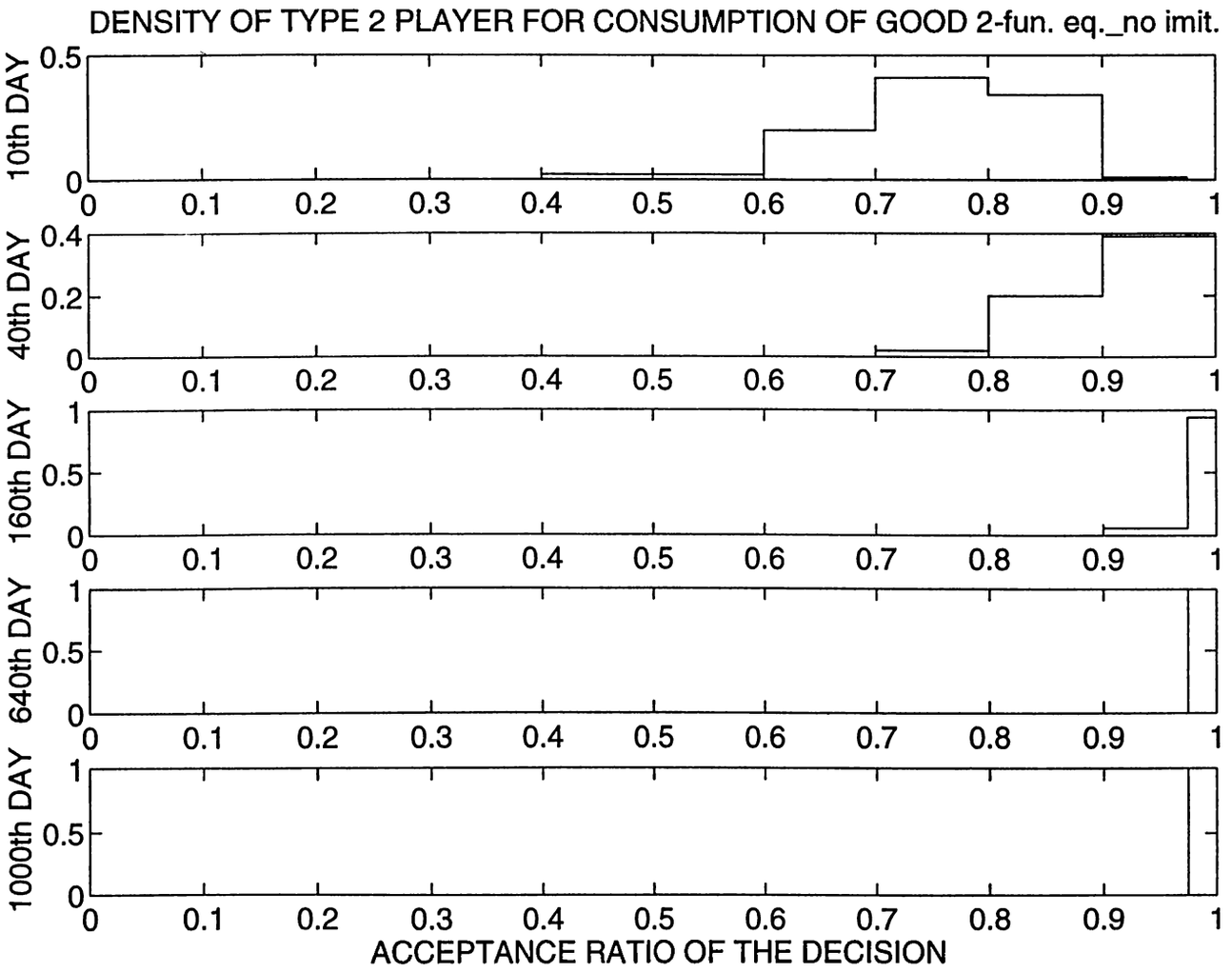

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 2-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

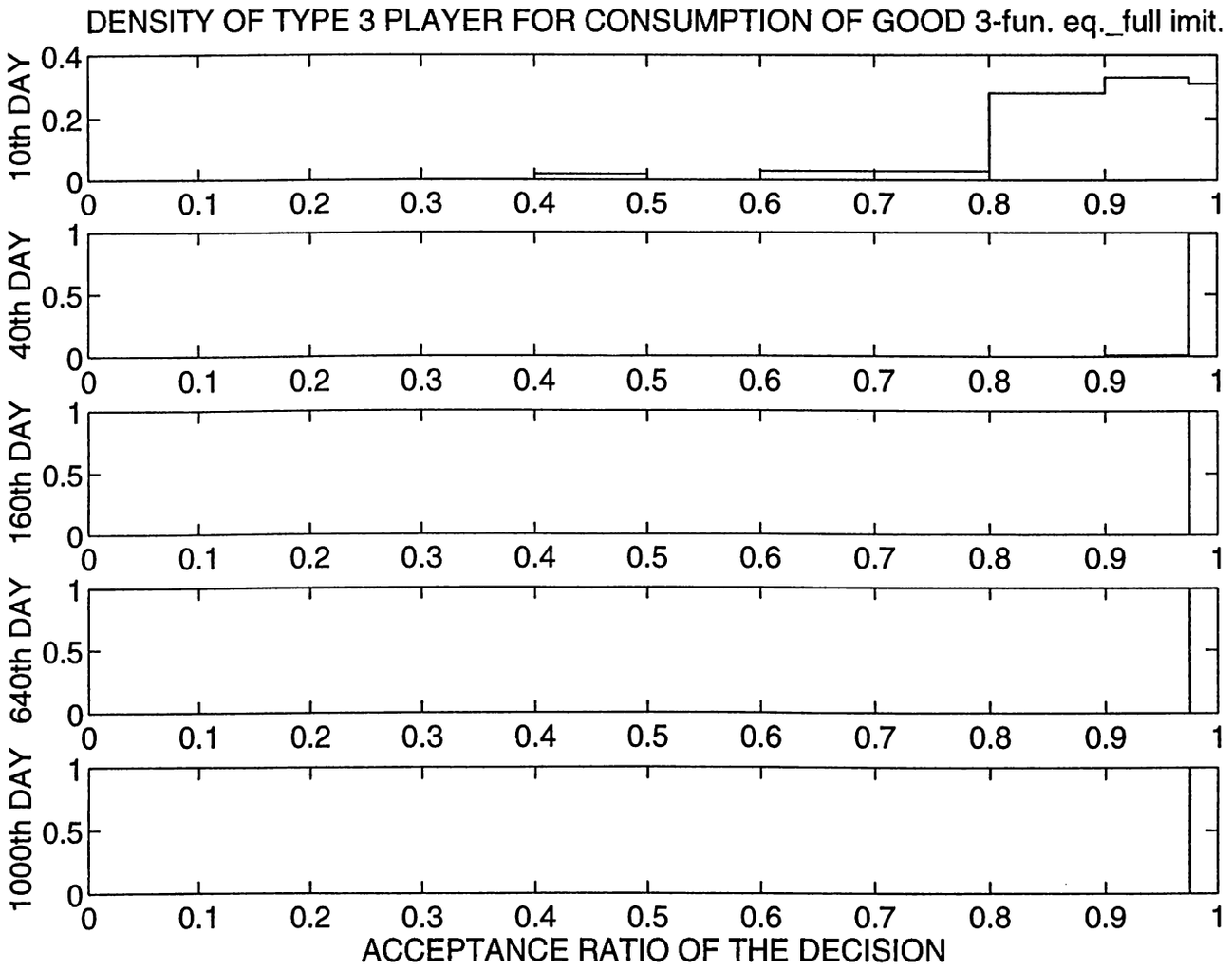

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 2-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
1
0.5
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

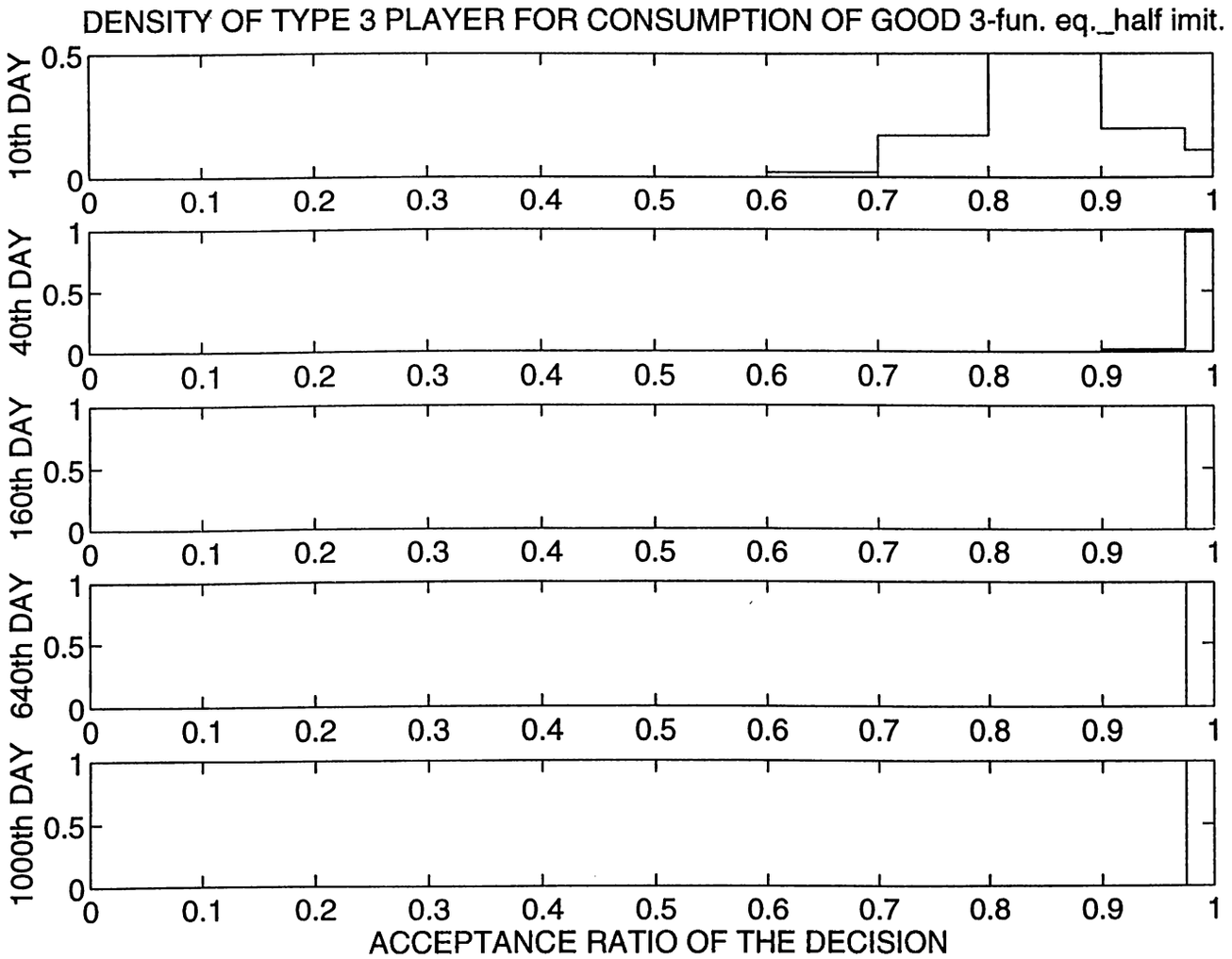

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 2-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
1
0.5
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

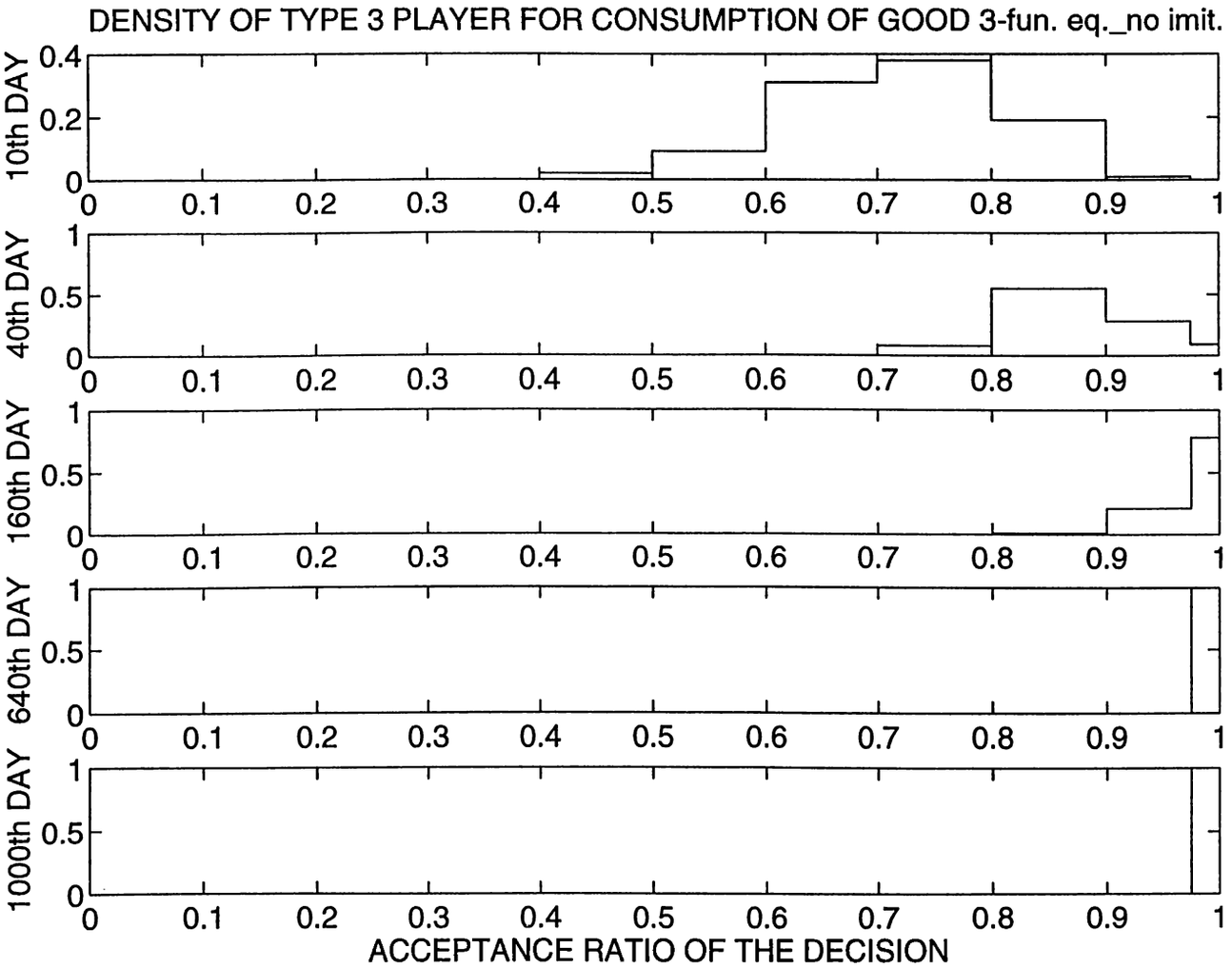

DENSITY OF TYPE 2 PLAYER FOR CONSUMPTION OF GOOD 2-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

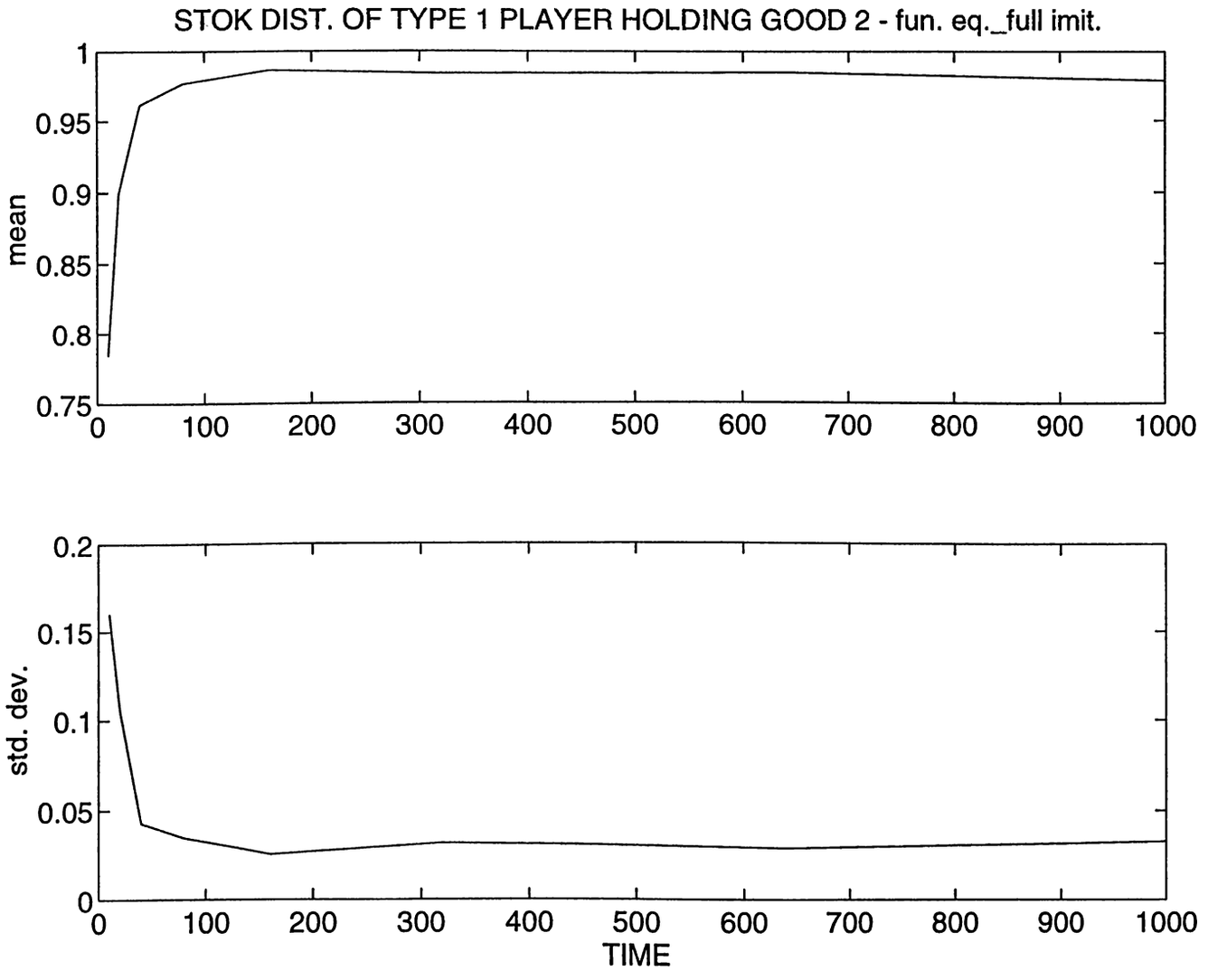

DENSITY OF TYPE 2 PLAYER FOR CONSUMPTION OF GOOD 2-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

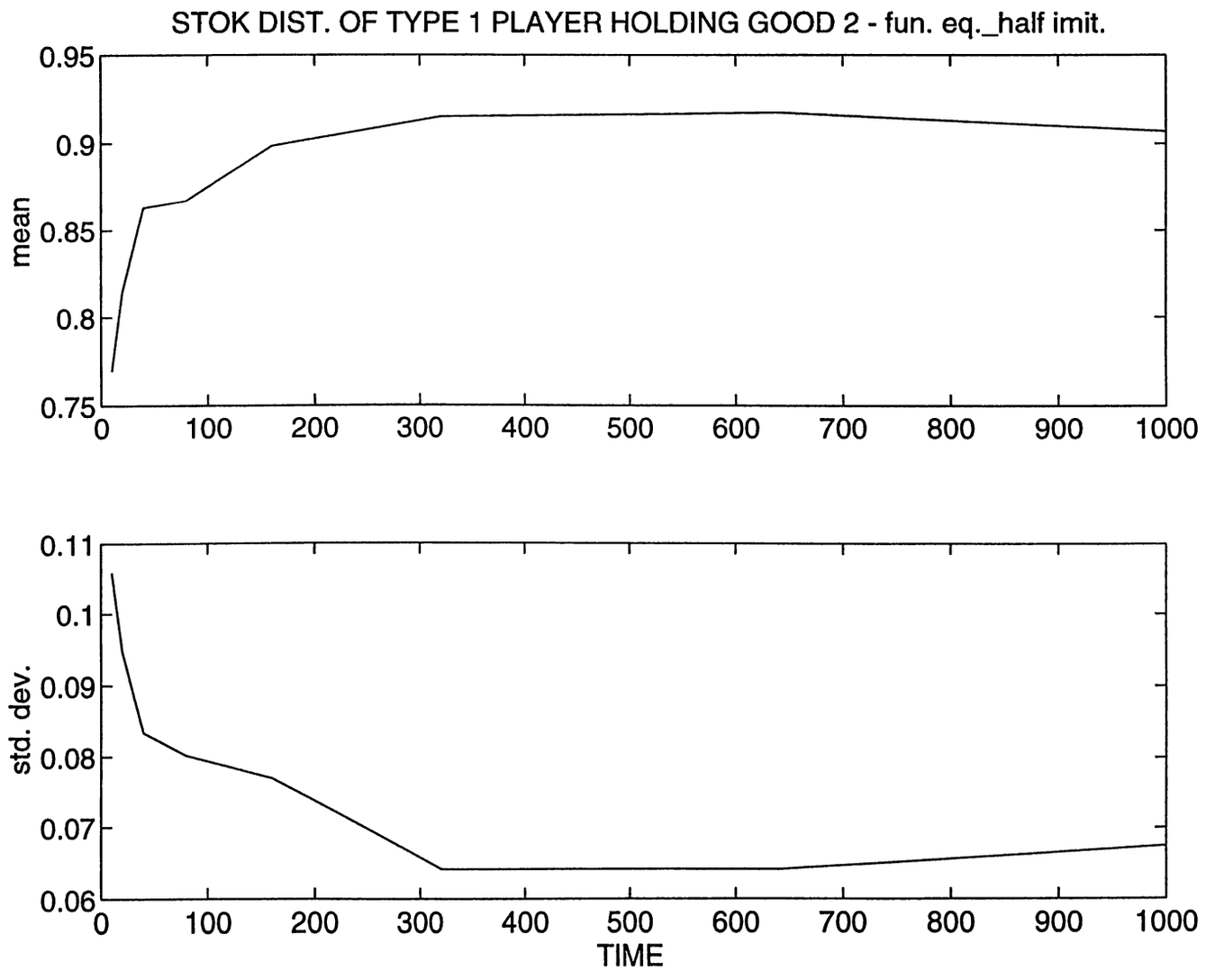

DENSITY OF TYPE 2 PLAYER FOR CONSUMPTION OF GOOD 2-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
0.4
0.2
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

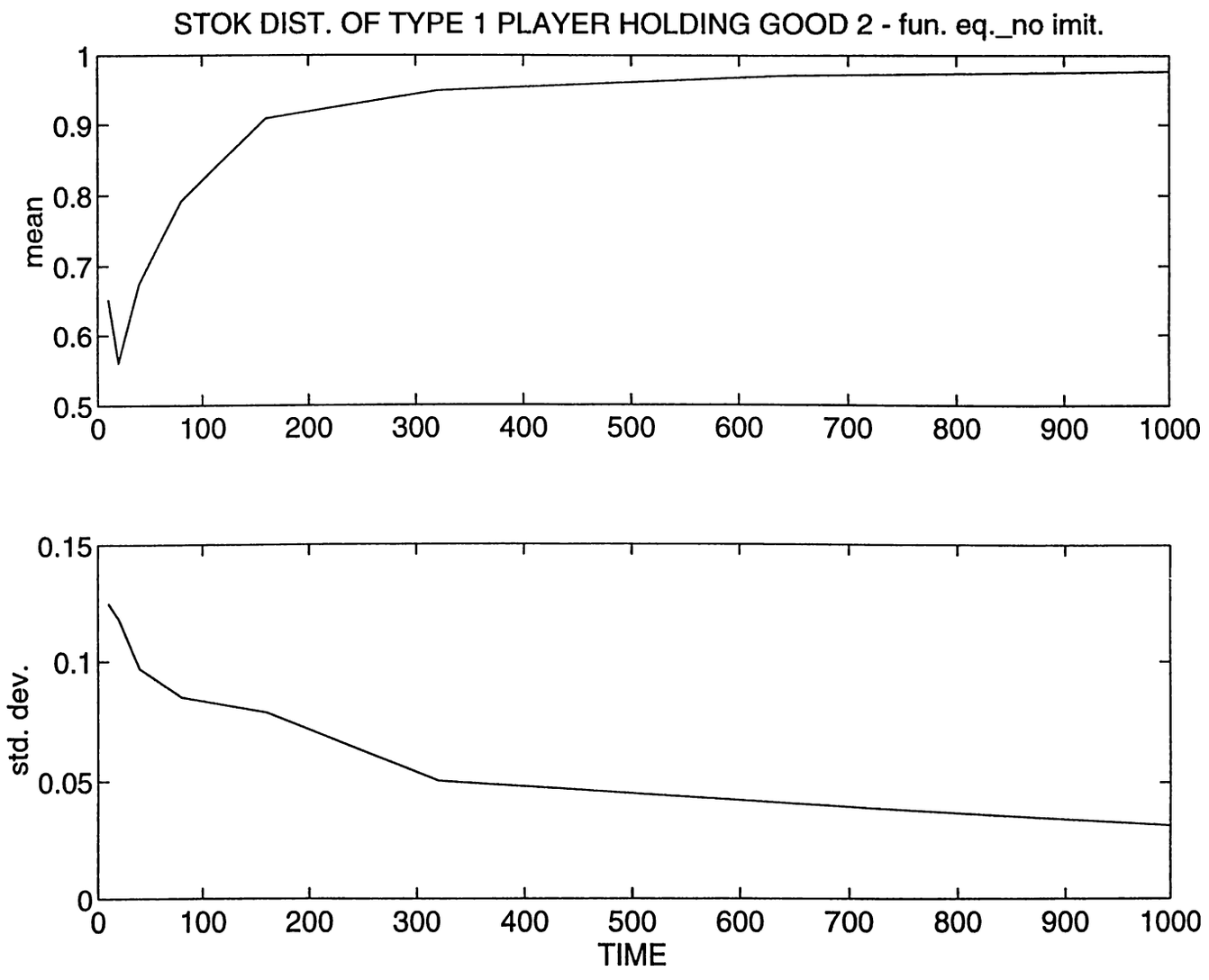

DENSITY OF TYPE 3 PLAYER FOR CONSUMPTION OF GOOD 3-fun. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

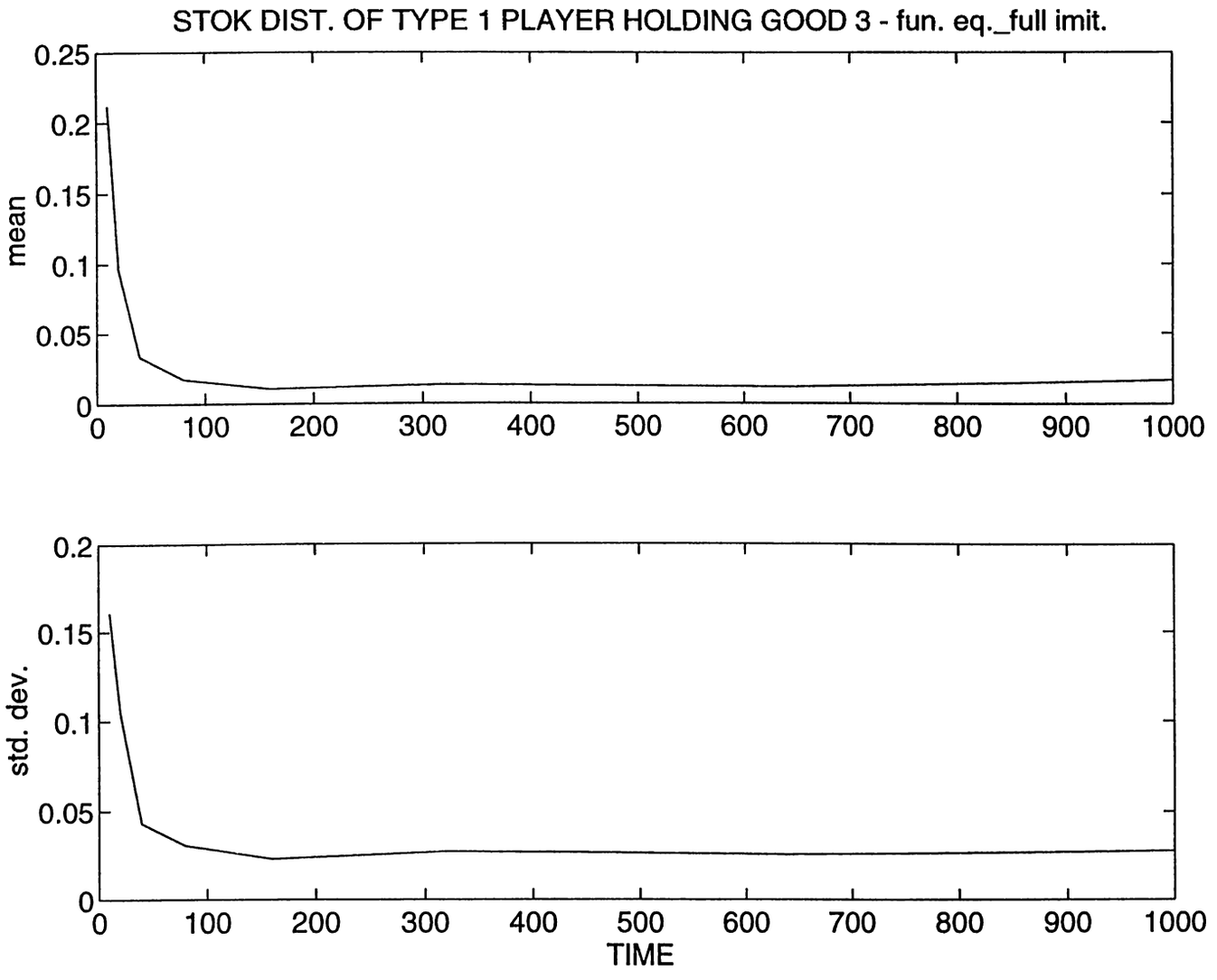

DENSITY OF TYPE 3 PLAYER FOR CONSUMPTION OF GOOD 3-fun. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

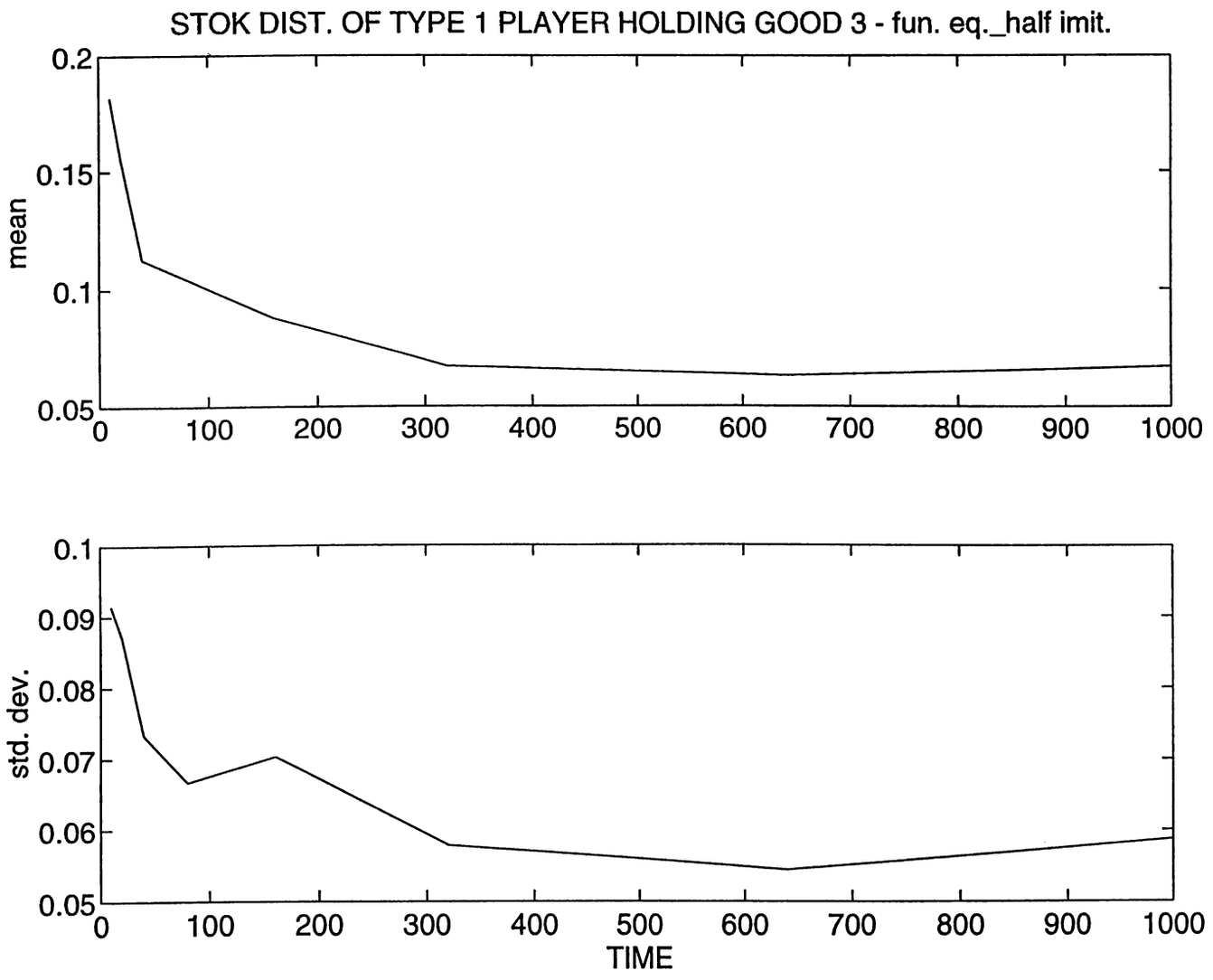

DENSITY OF TYPE 3 PLAYER FOR CONSUMPTION OF GOOD 3-fun. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

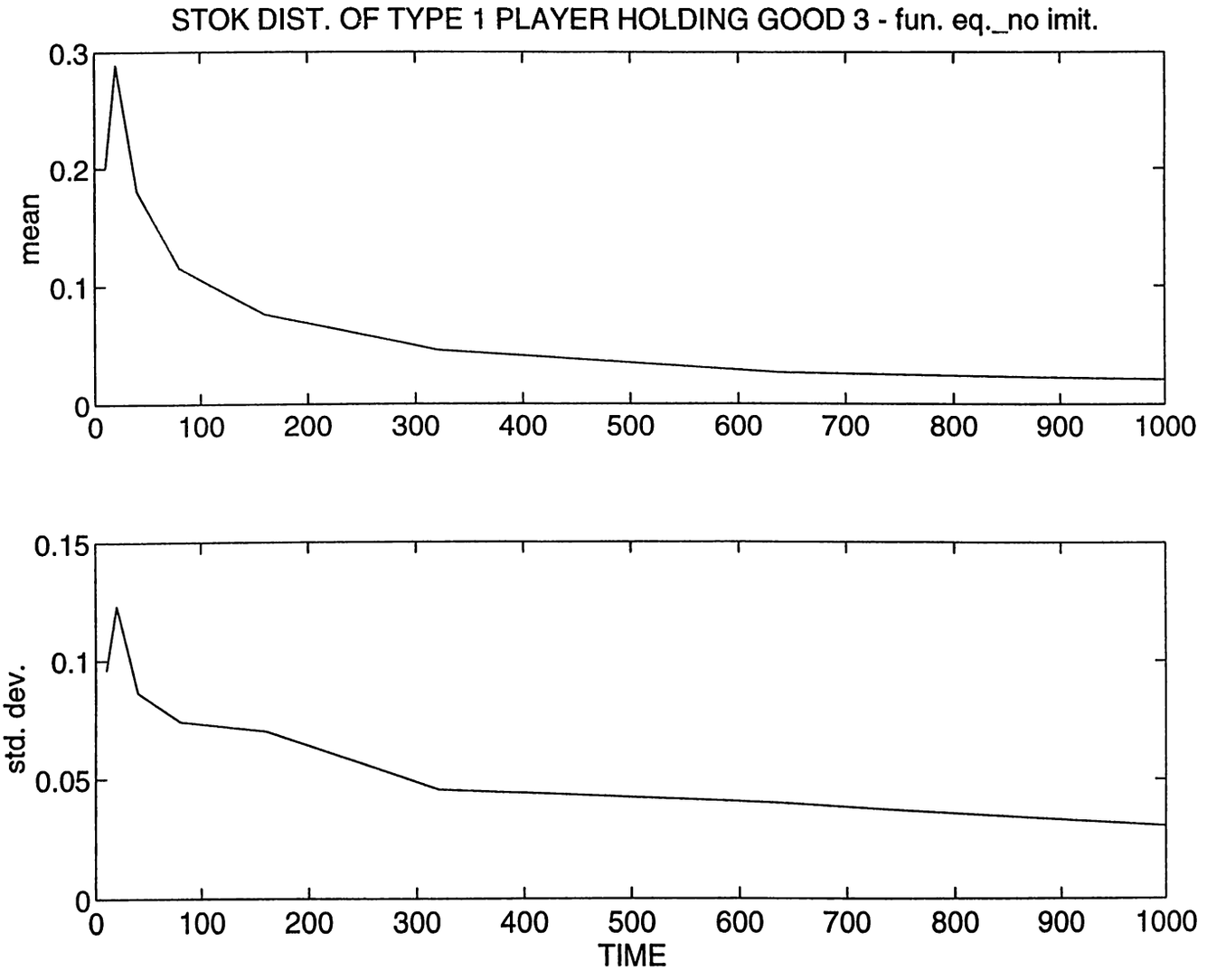

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 2 - fun. eq._full imit.
mean
1
0.95
0.9
0.85
0.8
0.75
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.2
0.15
0.1
0.05
0
TIME

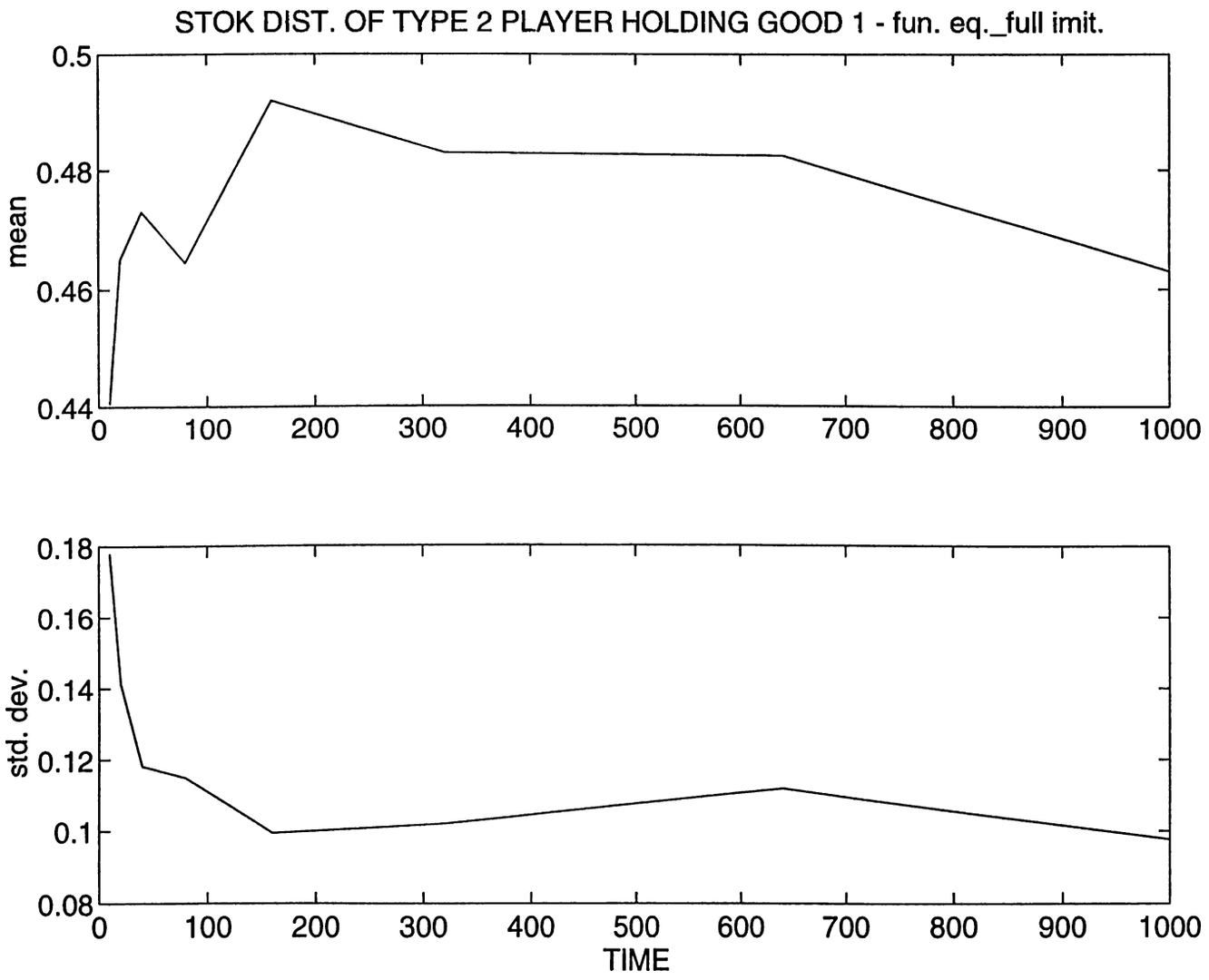

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 2 - fun. eq._half imit.
mean
0.95
0.9
0.85
0.8
0.75
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.11
0.1
0.09
0.08
0.07
0.06
TIME

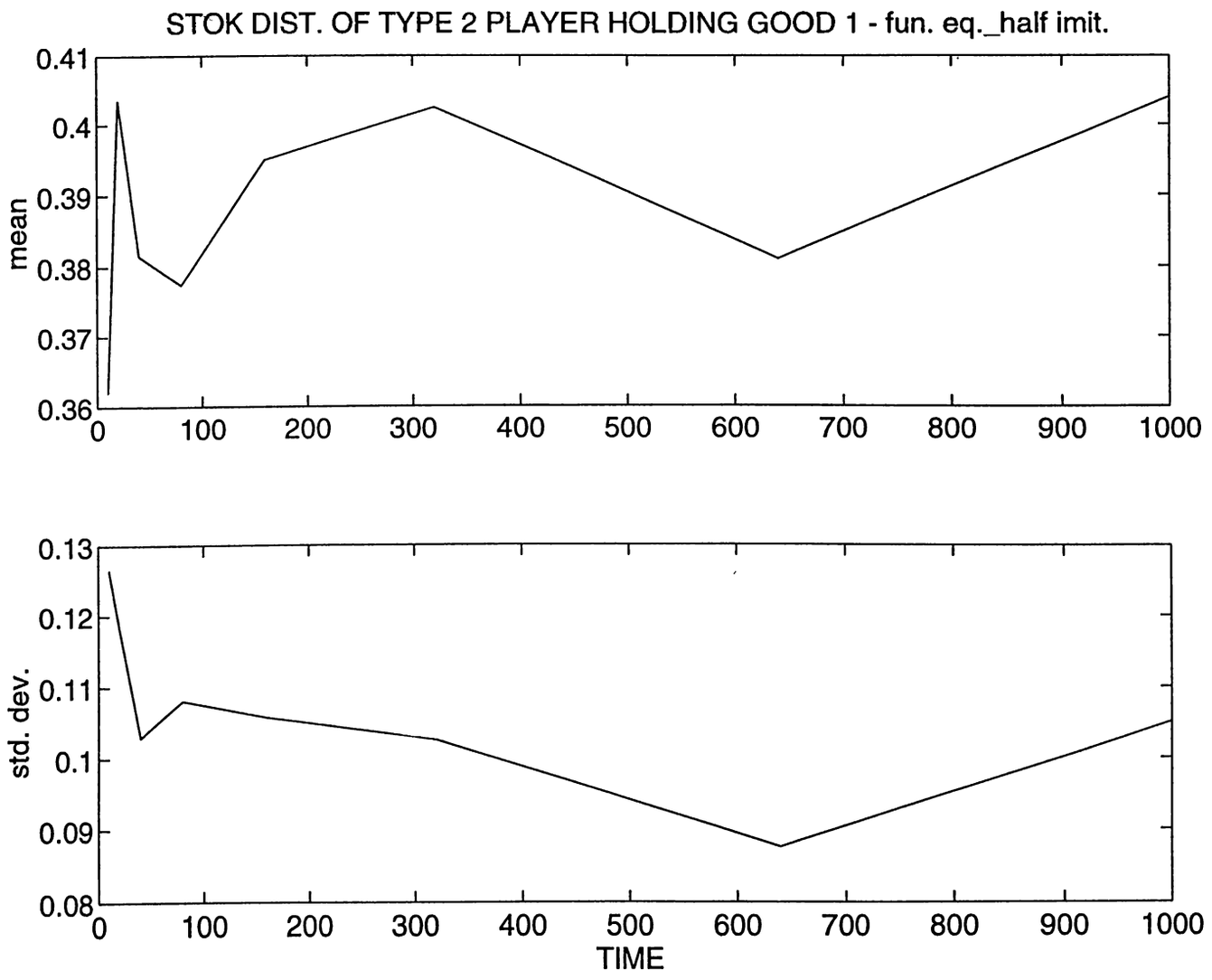

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 2 - fun. eq._no imit.
mean
1
0.9
0.8
0.7
0.6
0.5
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
TIME

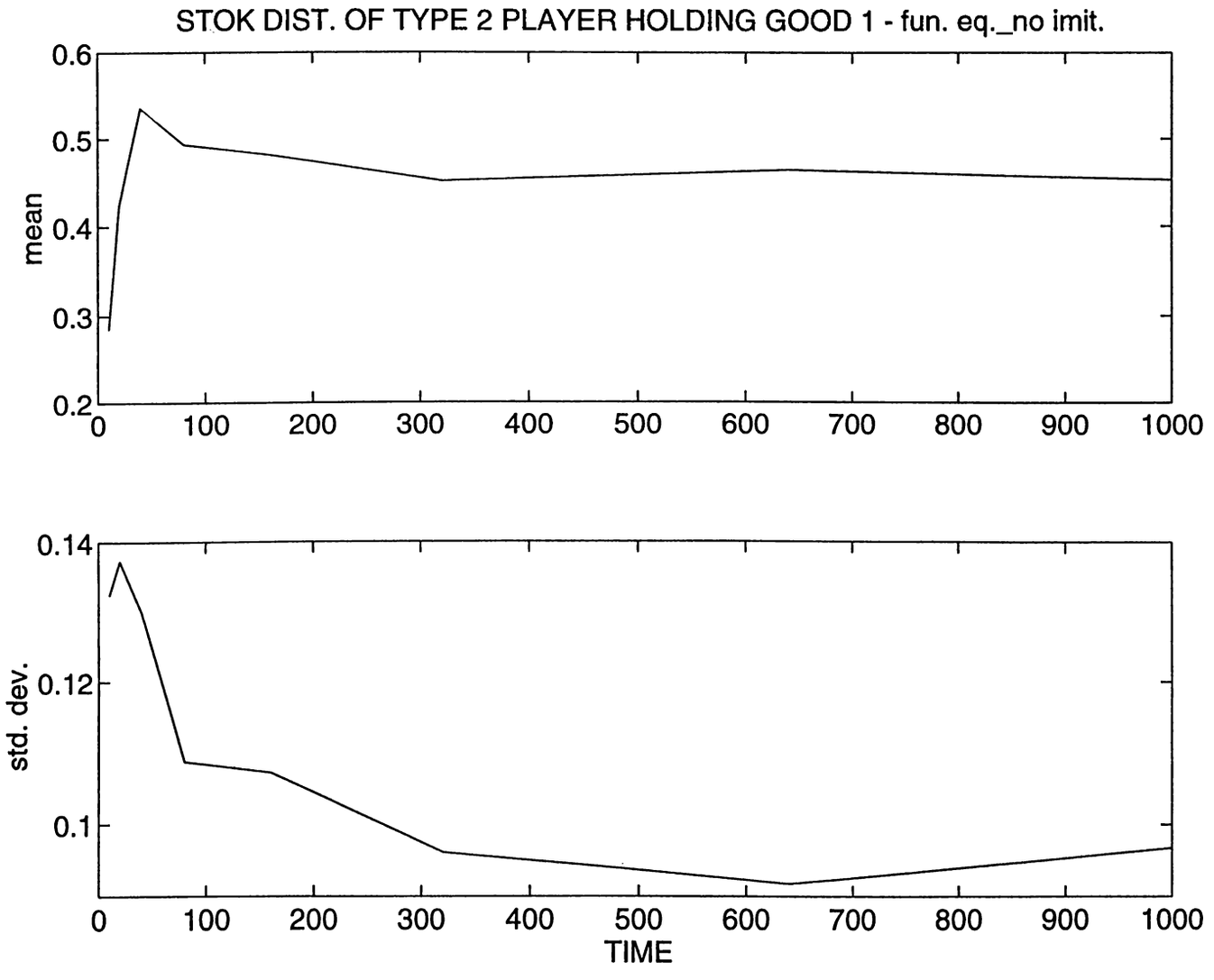

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 3 - fun. eq._full imit.
mean
0.25
0.2
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.2
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
TIME

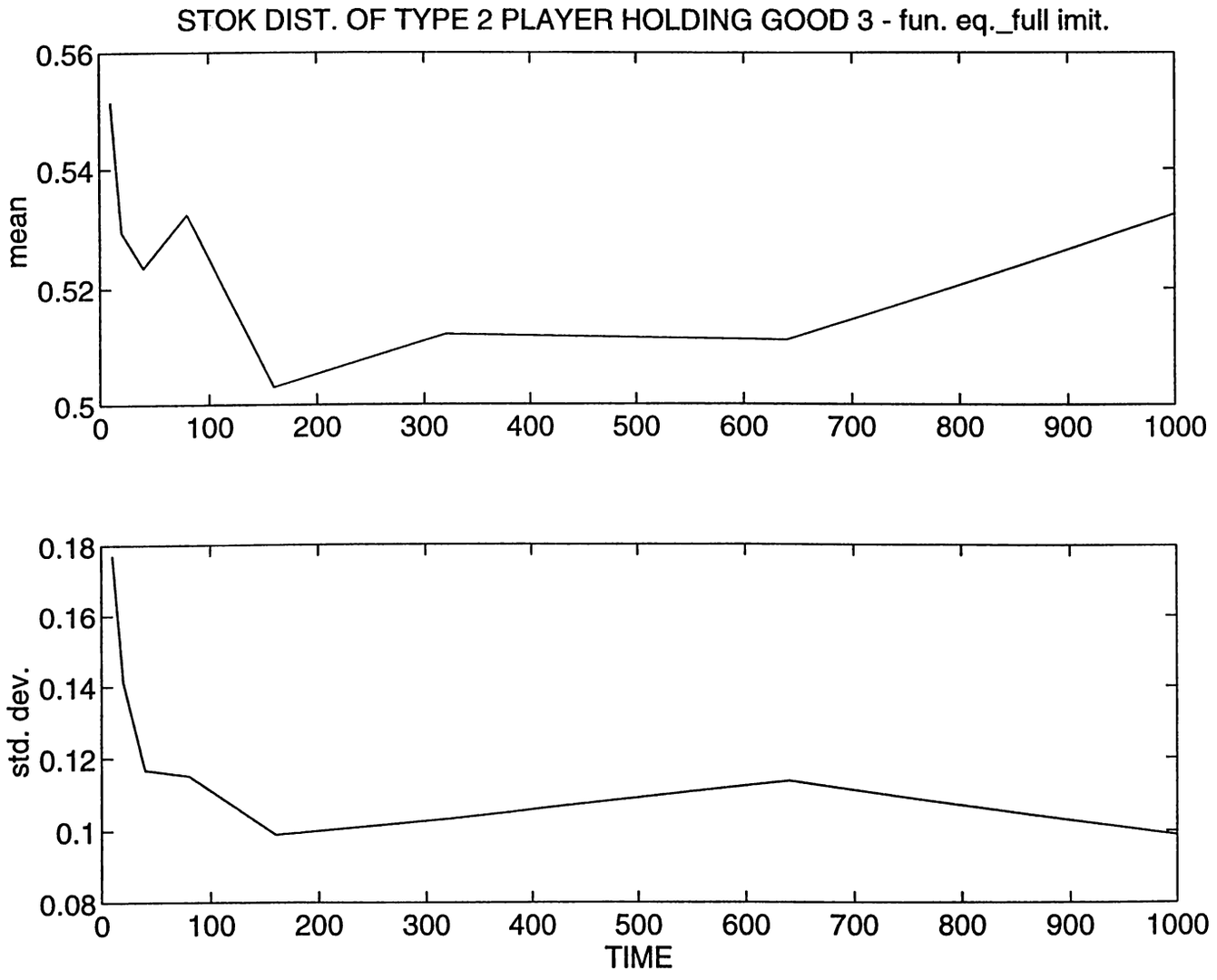

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 3 - fun. eq._half imit.
mean
0.2
0.15
0.1
0.05
std. dev.
0.1
0.09
0.08
0.07
0.06
0.05
0
100
200
300
400
500
600
700
800
900
1000
TIME

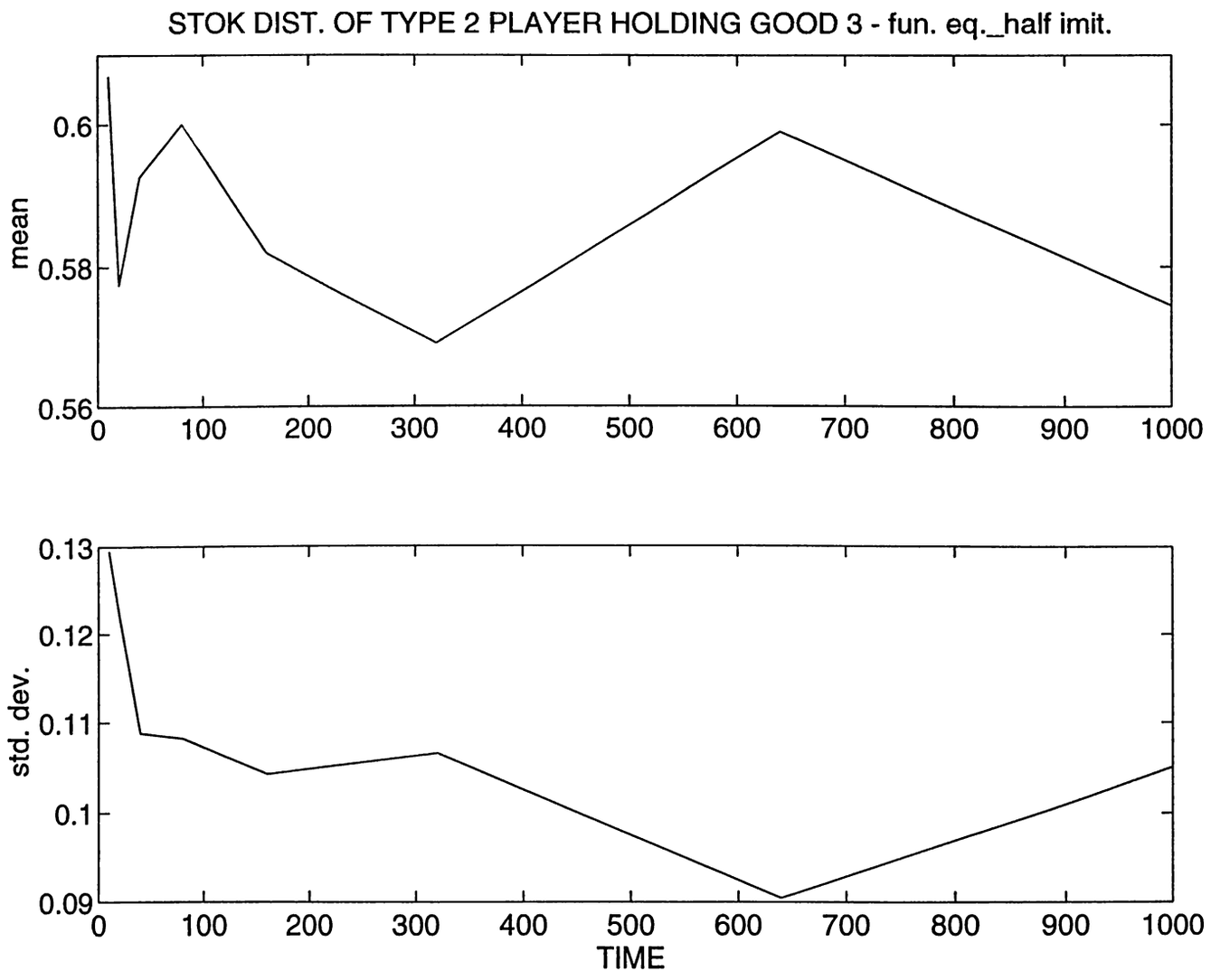

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 3 - fun. eq._no imit.
mean
0.3
0.2
0.1
0
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
TIME

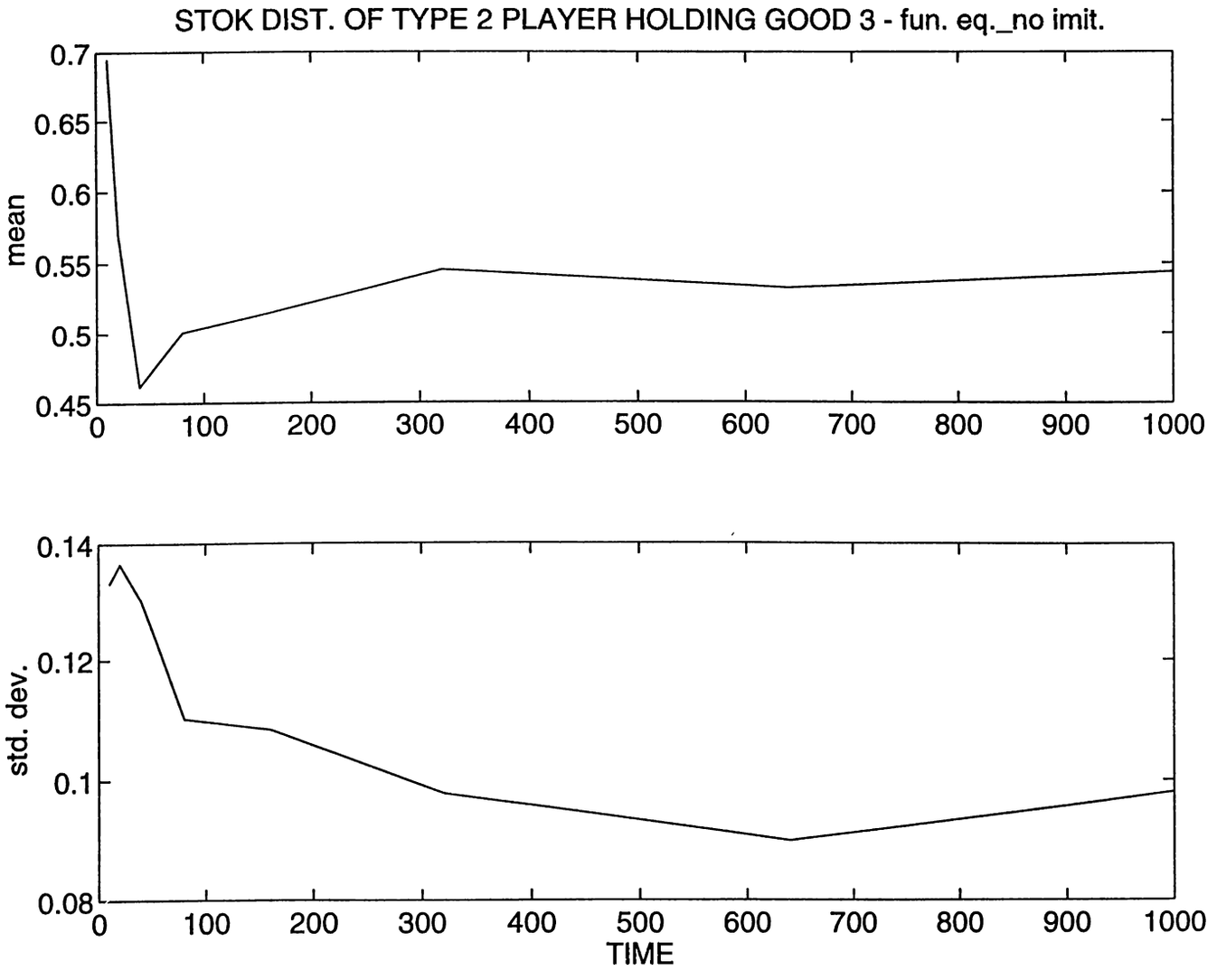

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 1 - fun. eq._full imit.
mean
0.5
0.48
0.46
0.44
std. dev.
0.18
0.16
0.14
0.12
0.1
0.08
0
100
200
300
400
500
600
700
800
900
1000
TIME

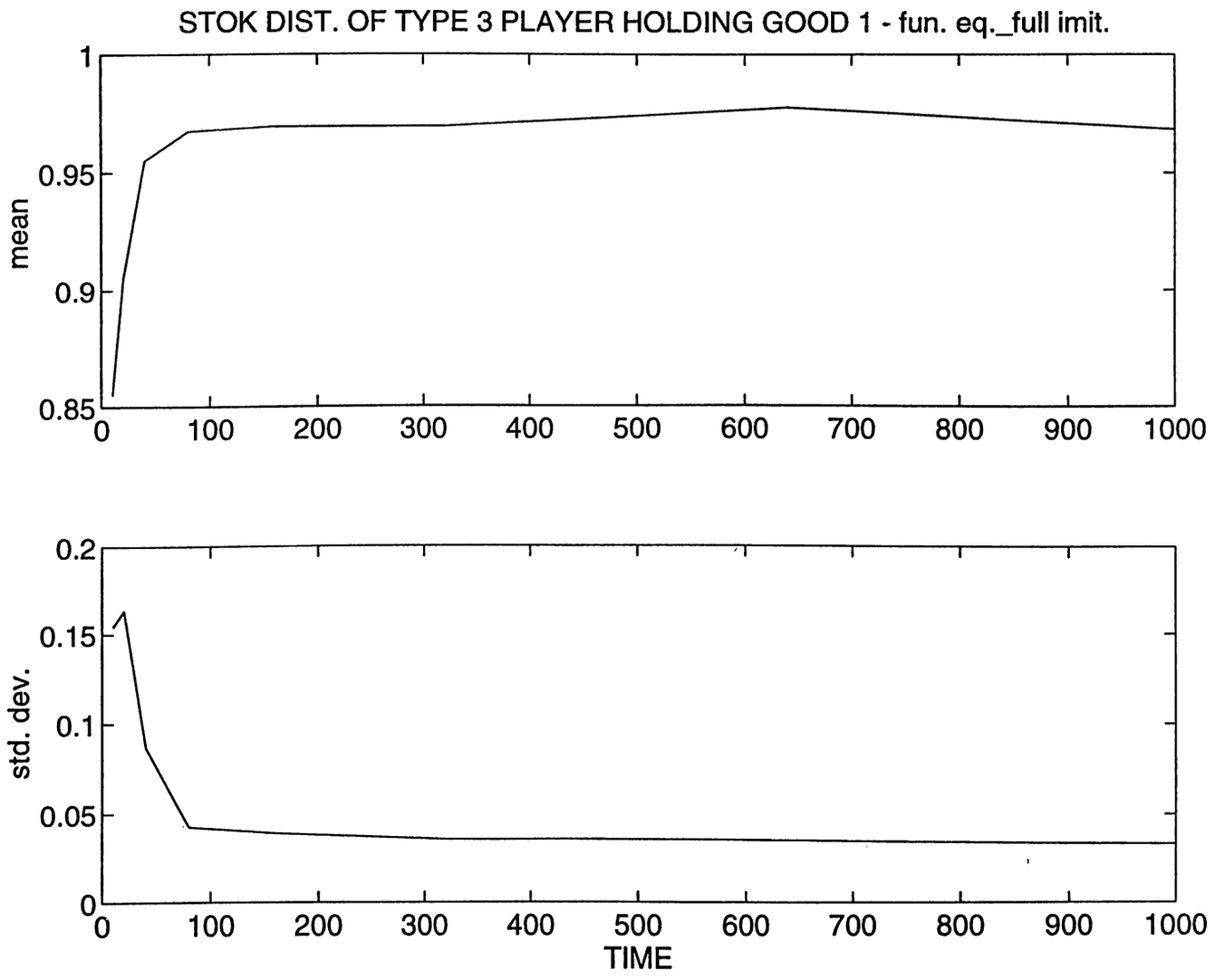

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 1 - fun. eq._half imit.
mean
0.41
0.4
0.39
0.38
0.37
0.36
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.13
0.12
0.11
0.1
0.09
0.08
0
100
200
300
400
500
600
700
800
900
1000
TIME

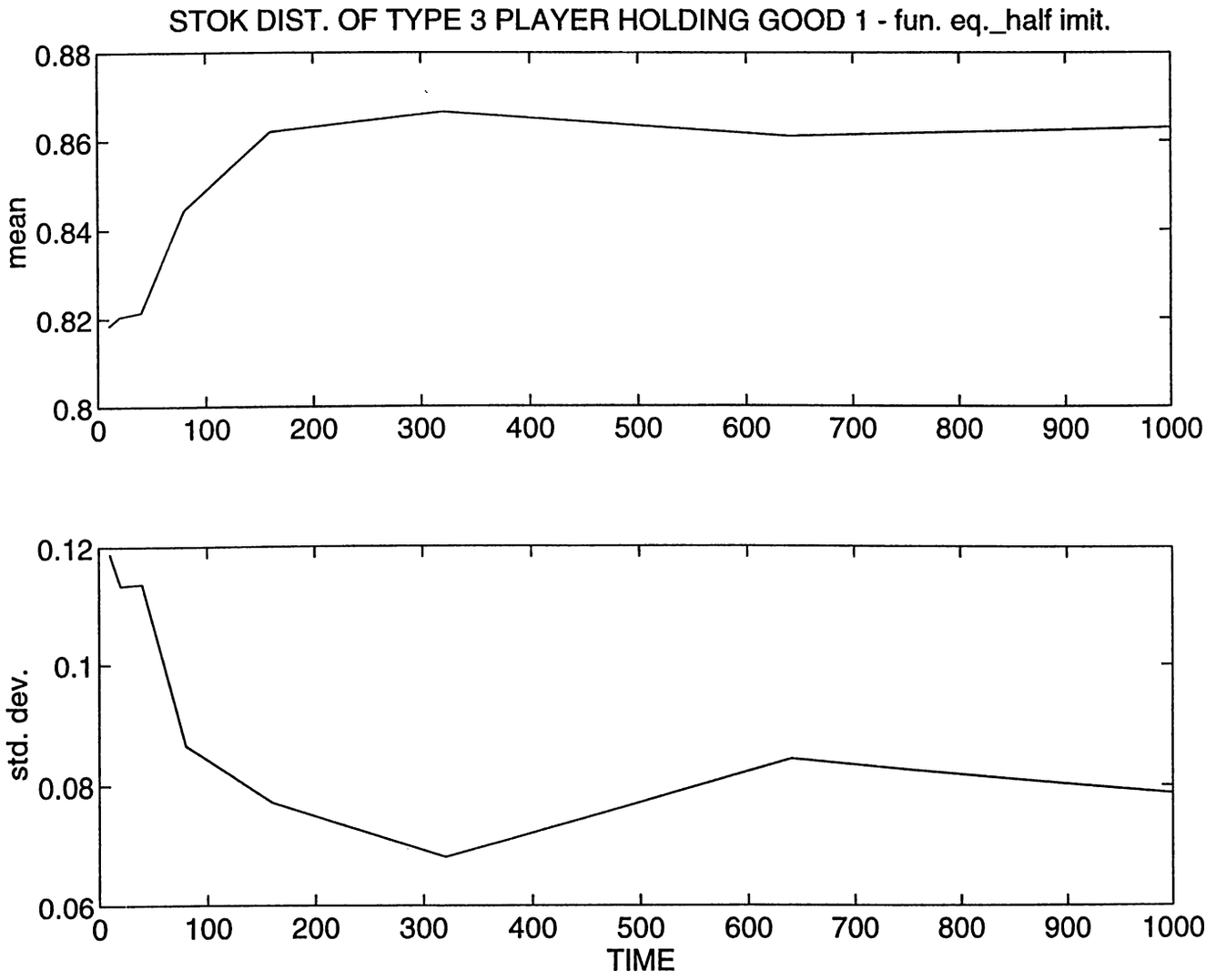

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 1 - fun. eq._no imit.
mean
0.6
0.5
0.4
0.3
0.2
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.14
0.12
0.1
0
100
200
300
400
500
600
700
800
900
1000
TIME

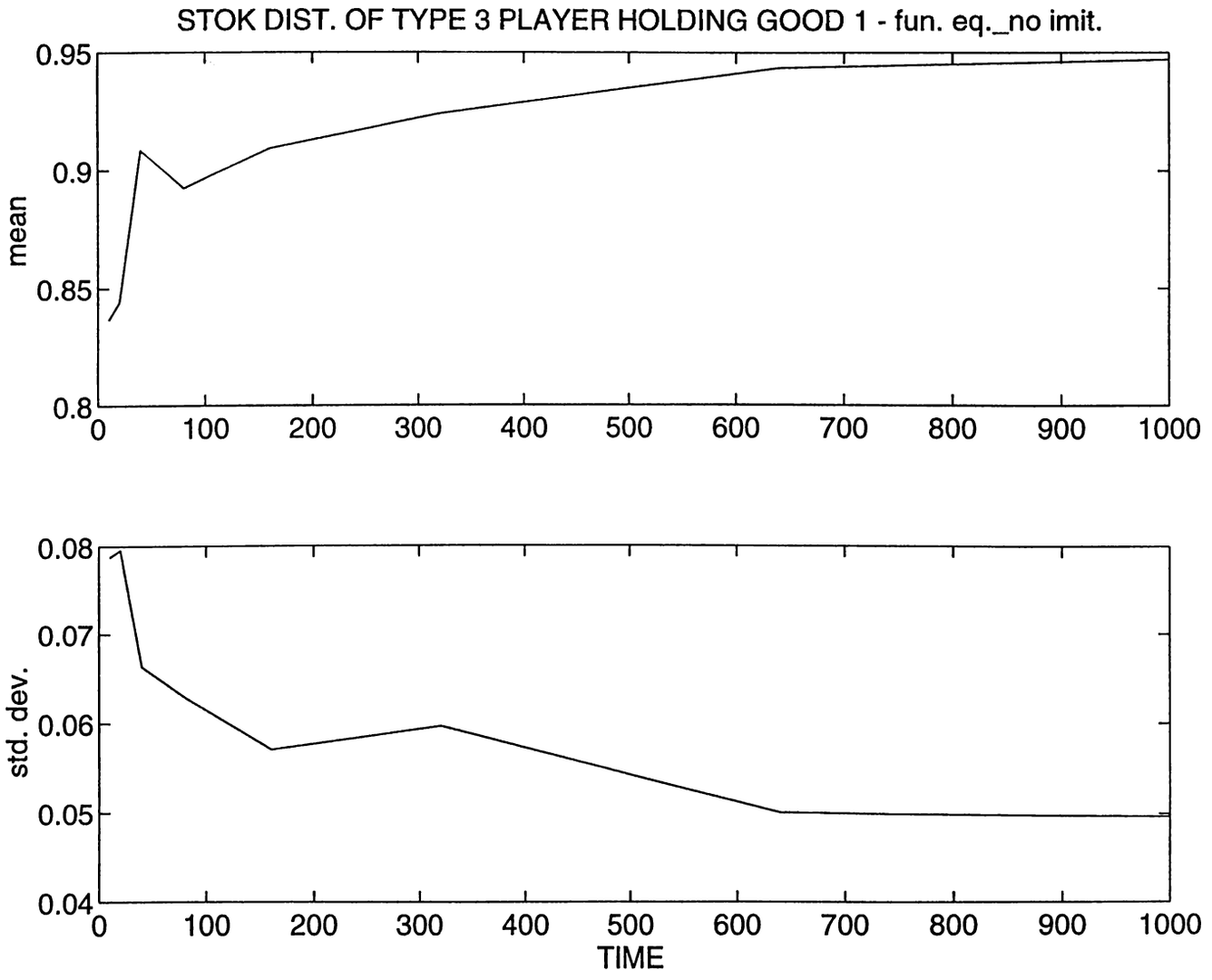

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 3 - fun. eq._full imit.
mean
0.56
0.54
0.52
0.5
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.18
0.16
0.14
0.12
0.1
0.08
0
100
200
300
400
500
600
700
800
900
1000
TIME

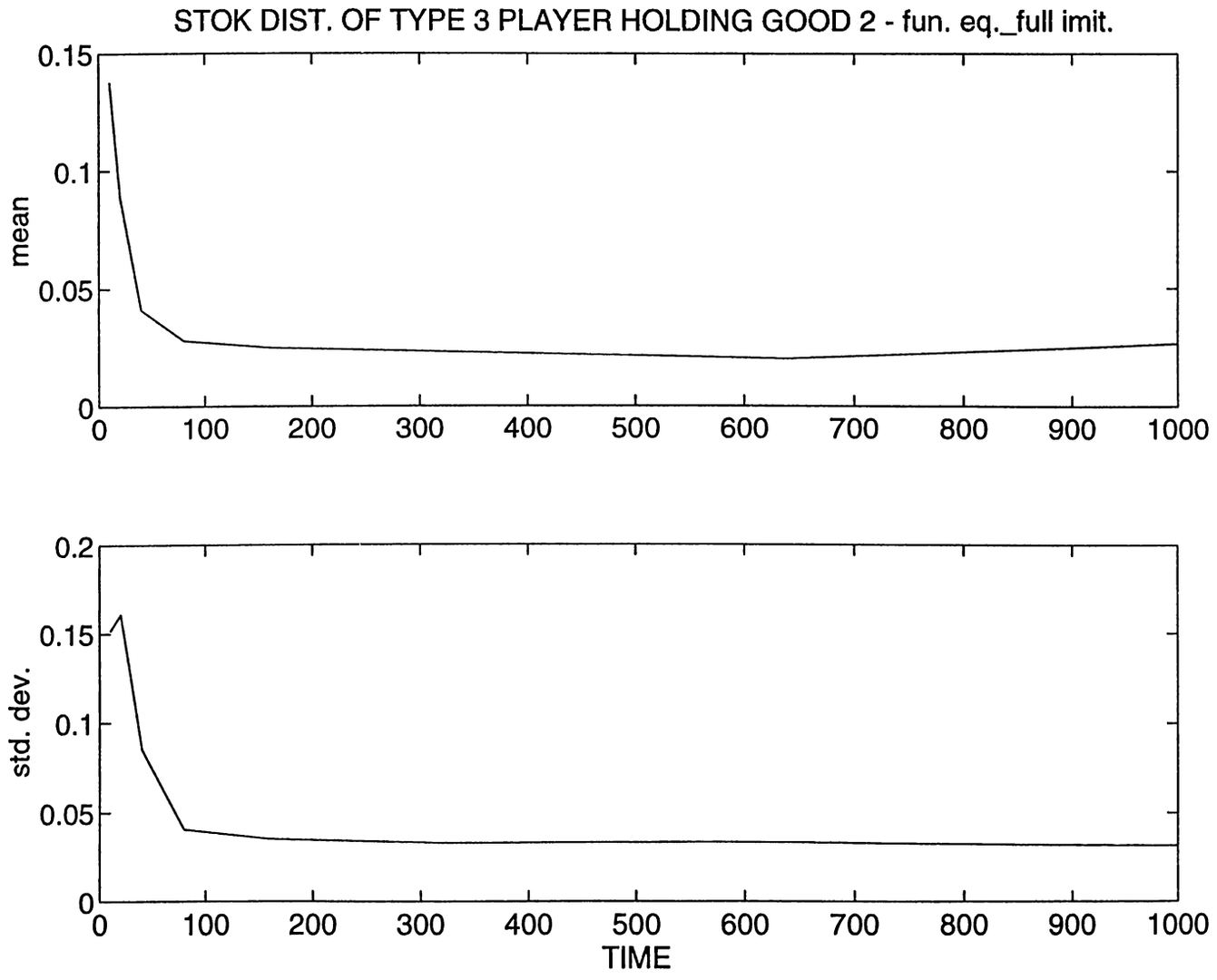

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 3 - fun. eq._half imit.
mean
0.6
0.58
0.56
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.13
0.12
0.11
0.1
0.09
TIME

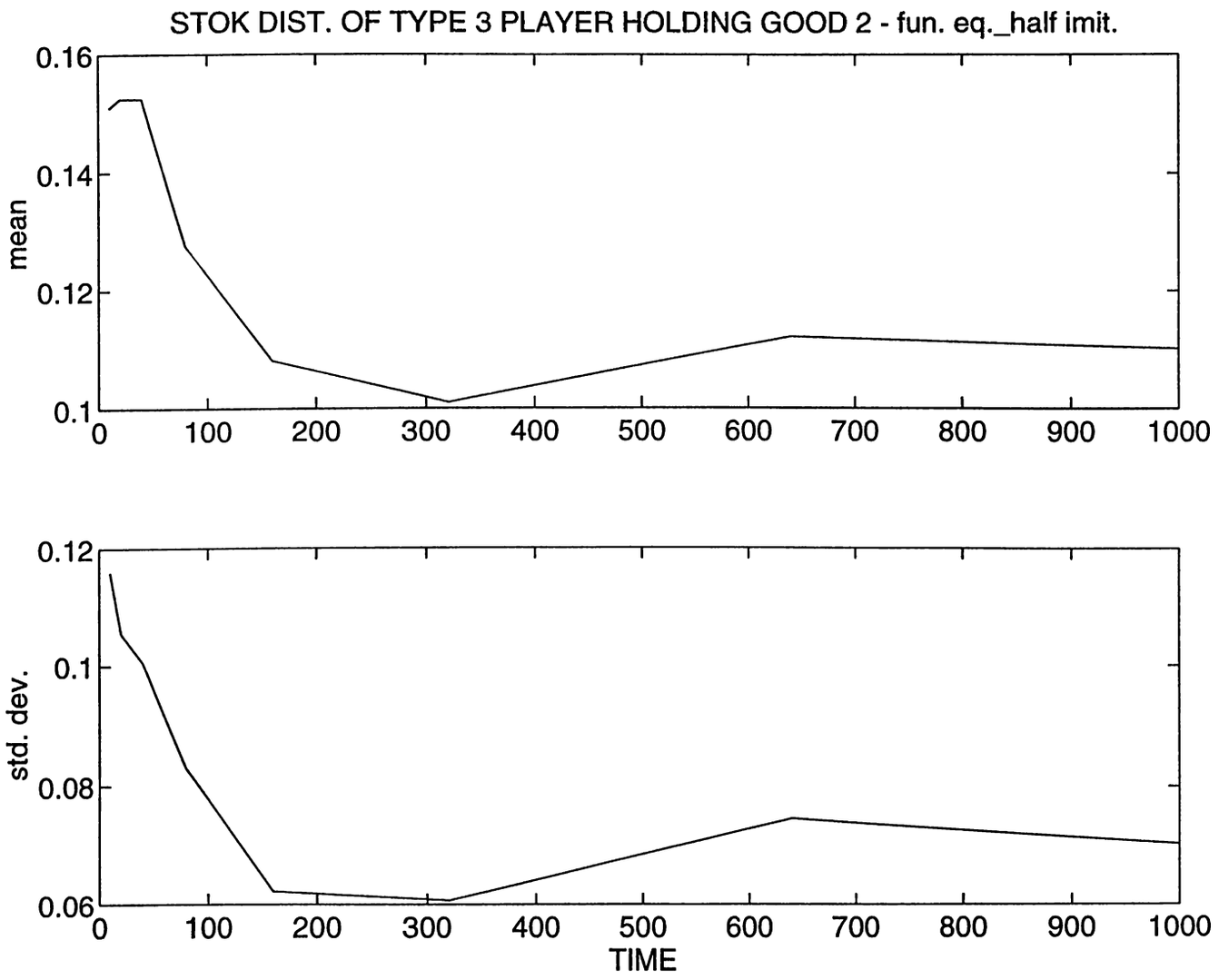

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 3 - fun. eq._no imit.
mean
0.7
0.65
0.6
0.55
0.5
0.45
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.14
0.12
0.1
0.08
0
100
200
300
400
500
600
700
800
900
1000
TIME

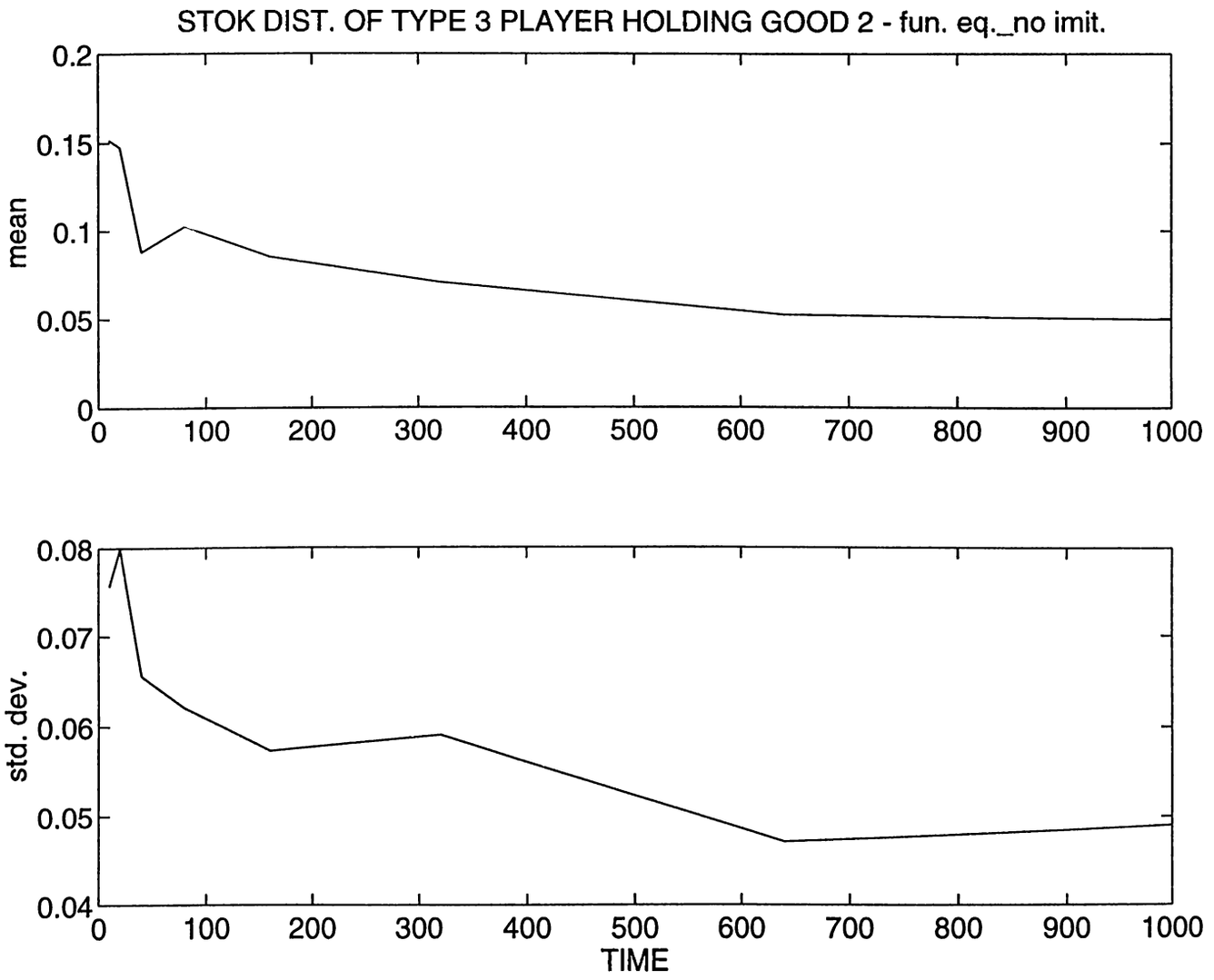

STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 1 - fun. eq._full imit.
mean
std. dev.
TIME
1
0.95
0.9
0.85
0.2
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000

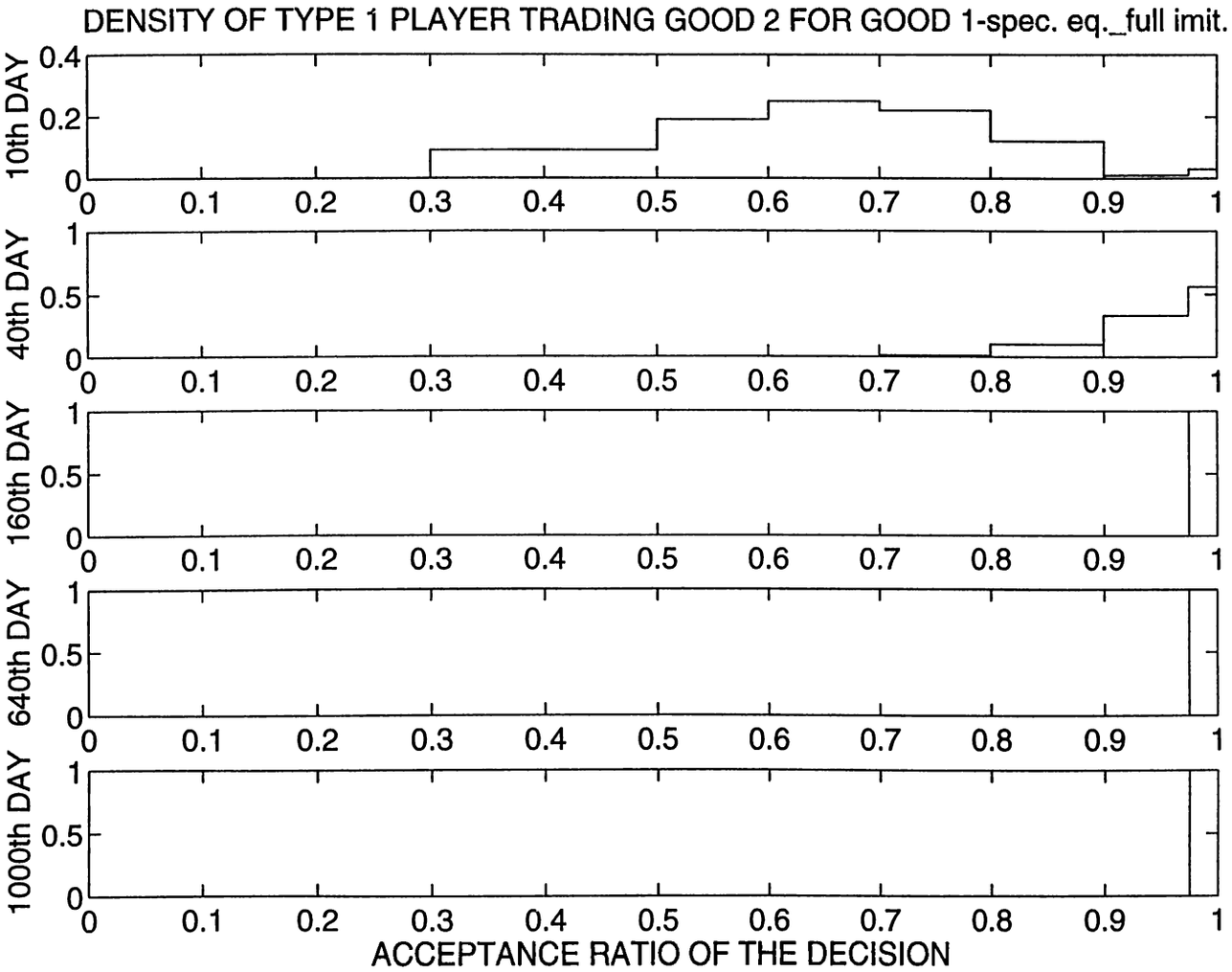

STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 1 - fun. eq._half imit.
mean
0.88
0.86
0.84
0.82
0.8
std. dev.
0.12
0.1
0.08
0.06
0
100
200
300
400
500
600
700
800
900
1000
TIME

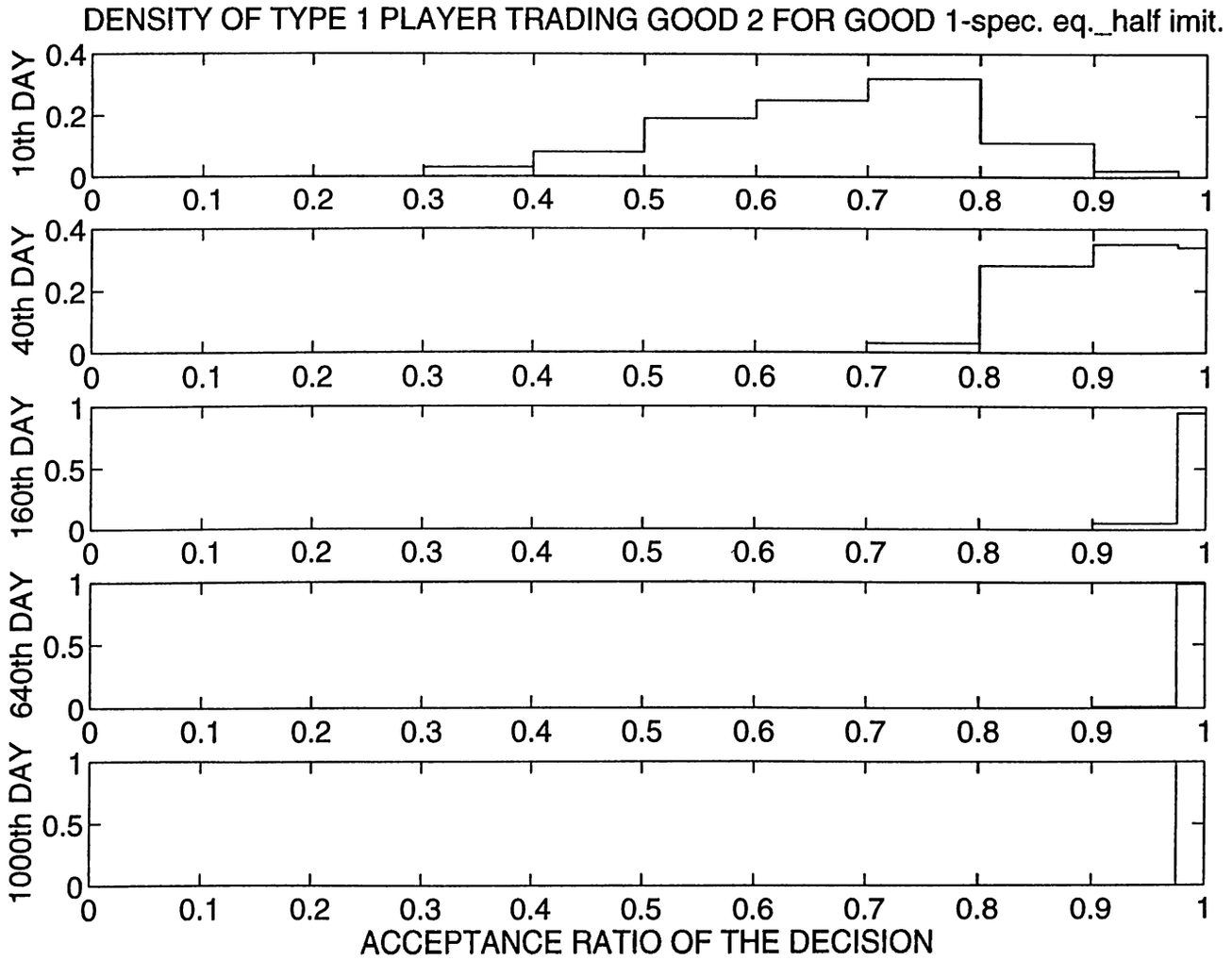

STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 1 - fun. eq._no imit.
mean
0.95
0.9
0.85
0.8
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.08
0.07
0.06
0.05
0.04
TIME

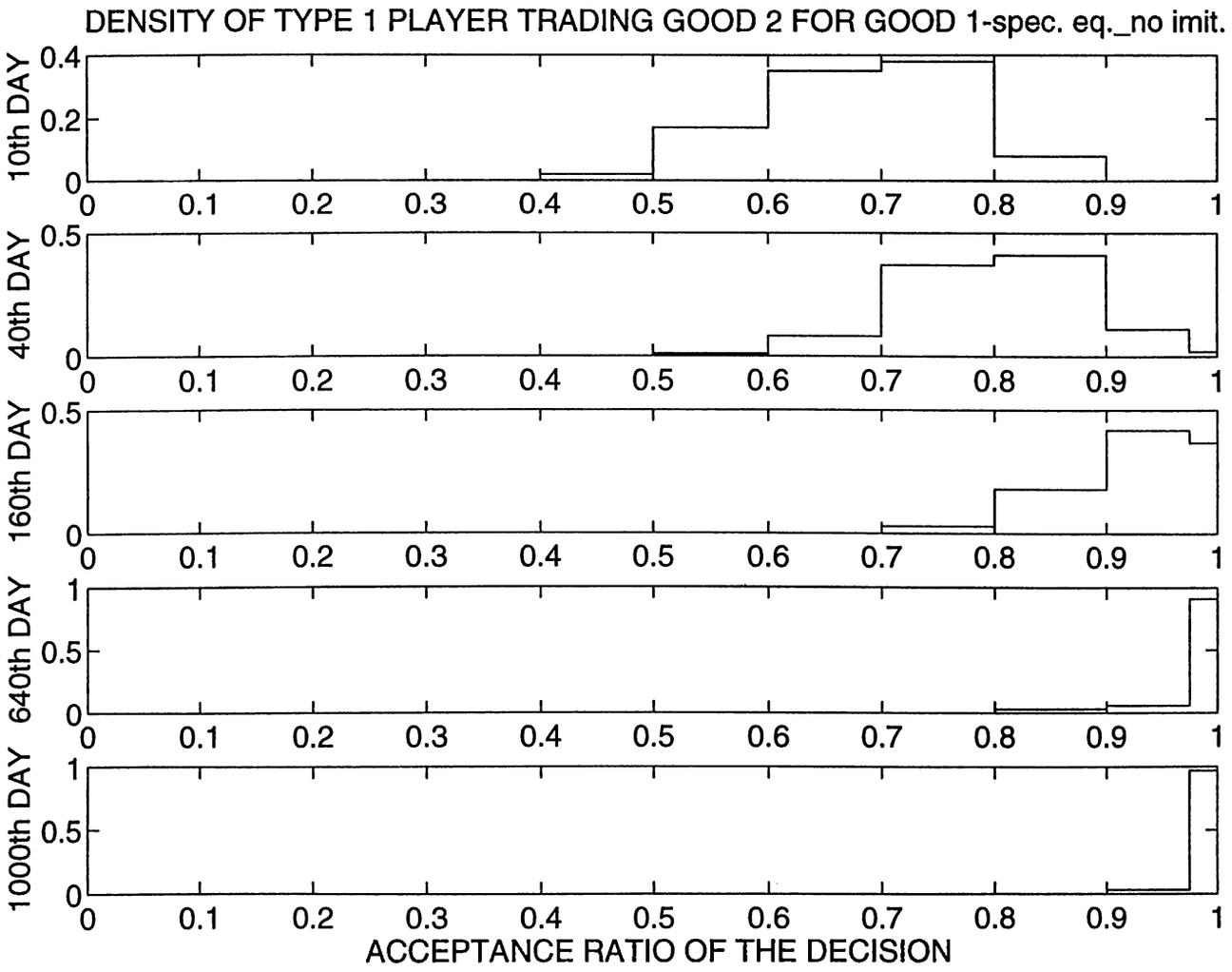

STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 2 - fun. eq._full imit.
mean
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.2
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
TIME

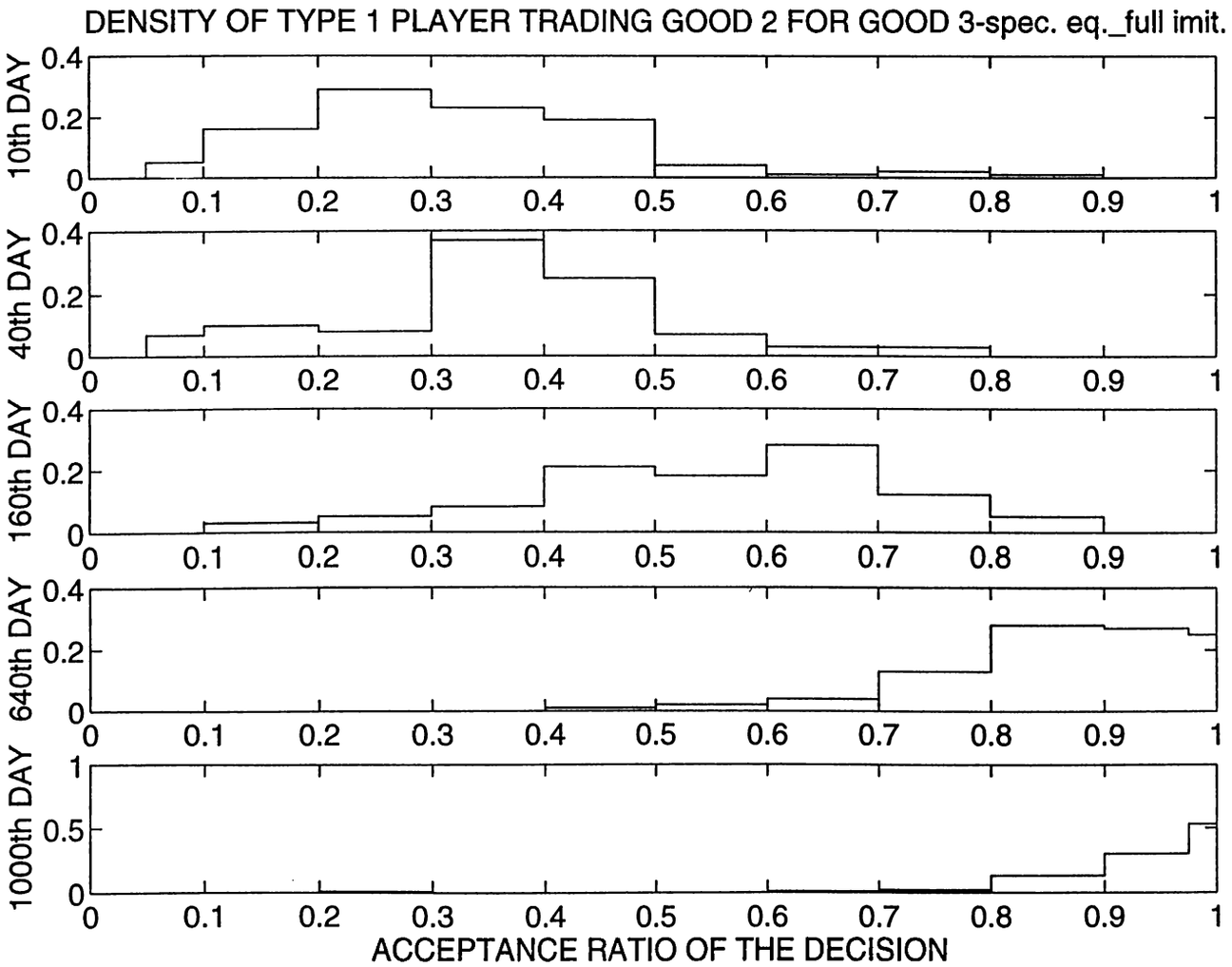

STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 2 - fun. eq._half imit.
mean
0.16
0.14
0.12
0.1
std. dev.
0.12
0.1
0.08
0.06
0
100
200
300
400
500
600
700
800
900
1000
TIME

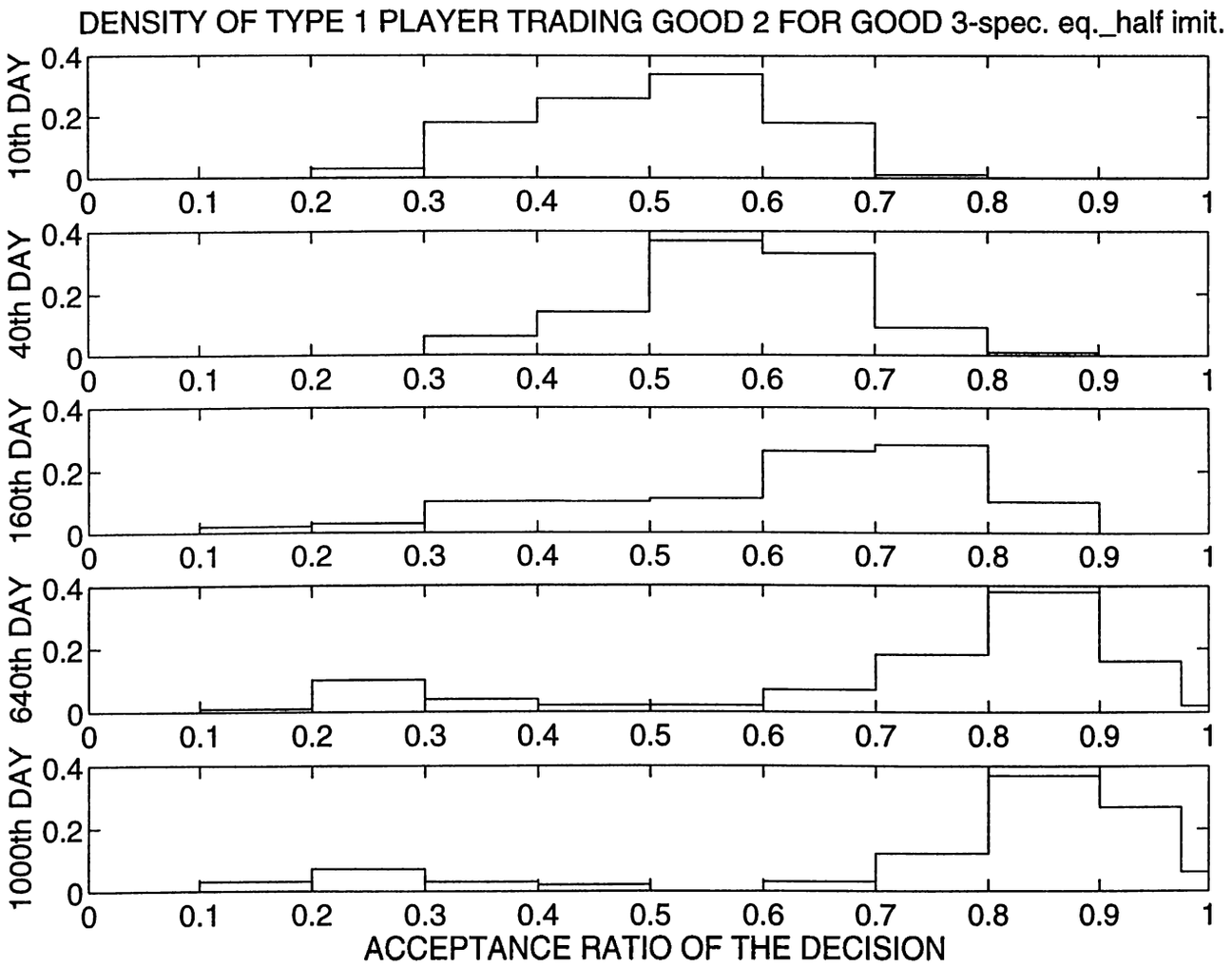

STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 2 - fun. eq._no imit.
mean
0.2
0.15
0.1
0.05
0
std. dev.
0.08
0.07
0.06
0.05
0.04
0
100
200
300
400
500
600
700
800
900
1000
TIME

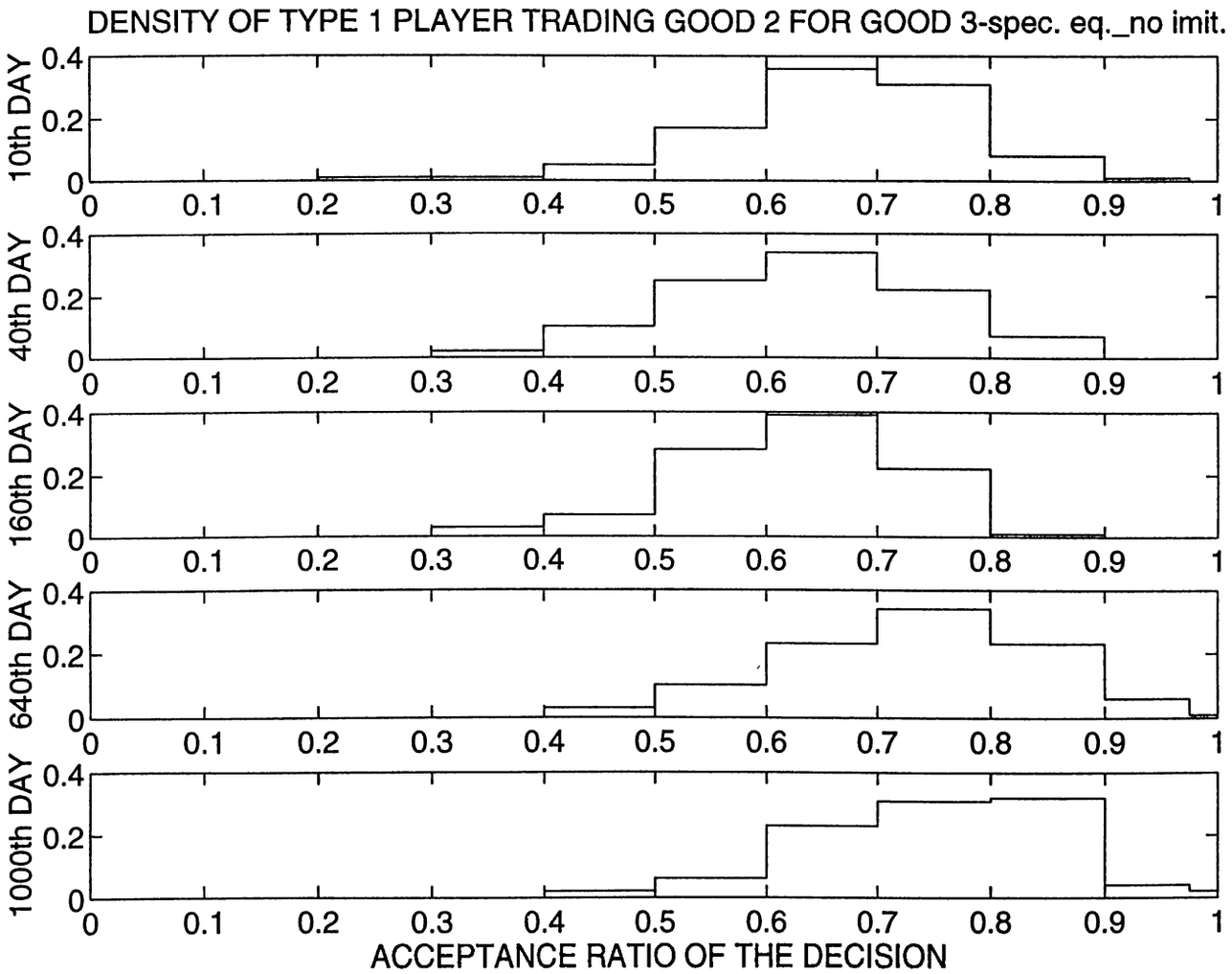

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

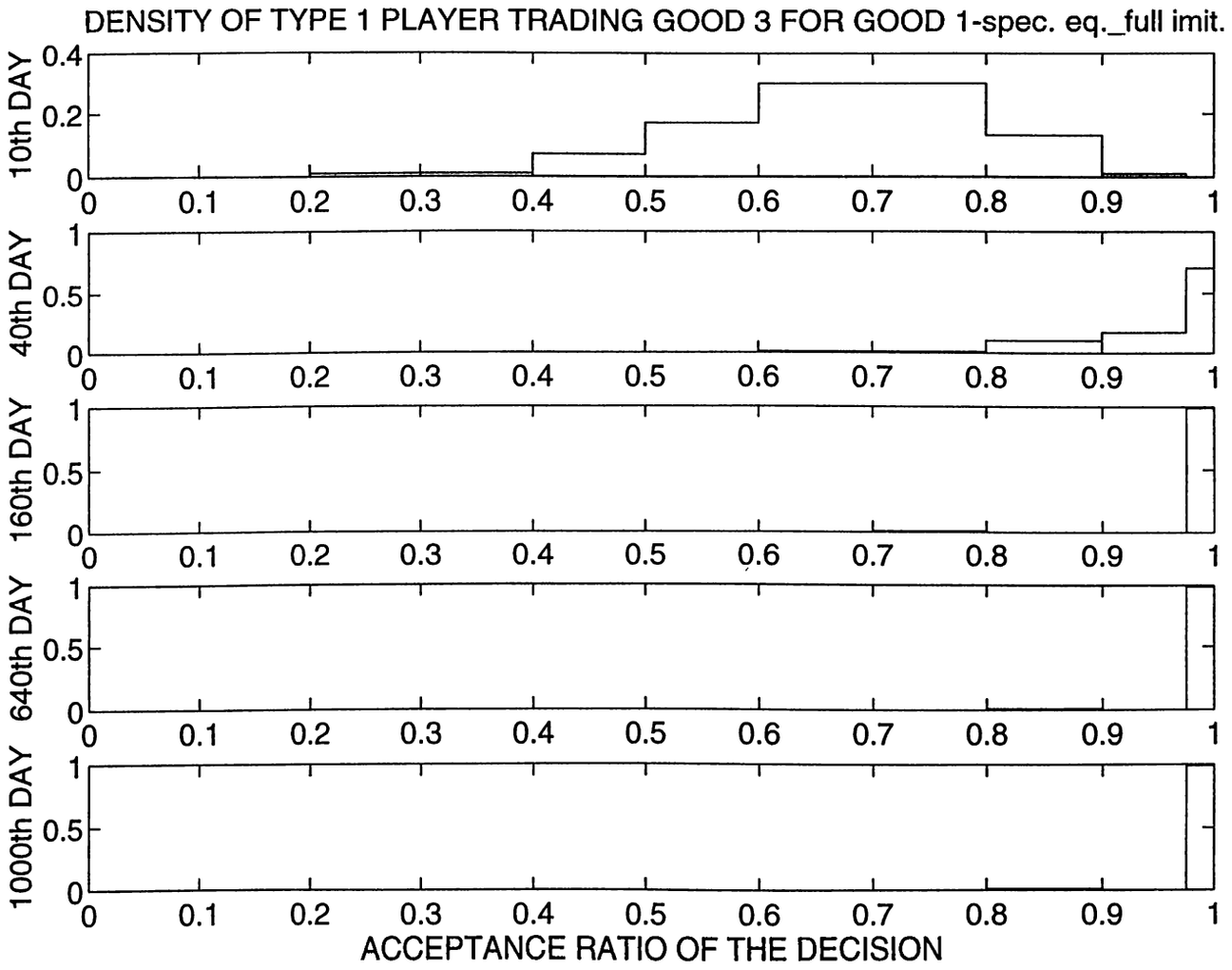

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

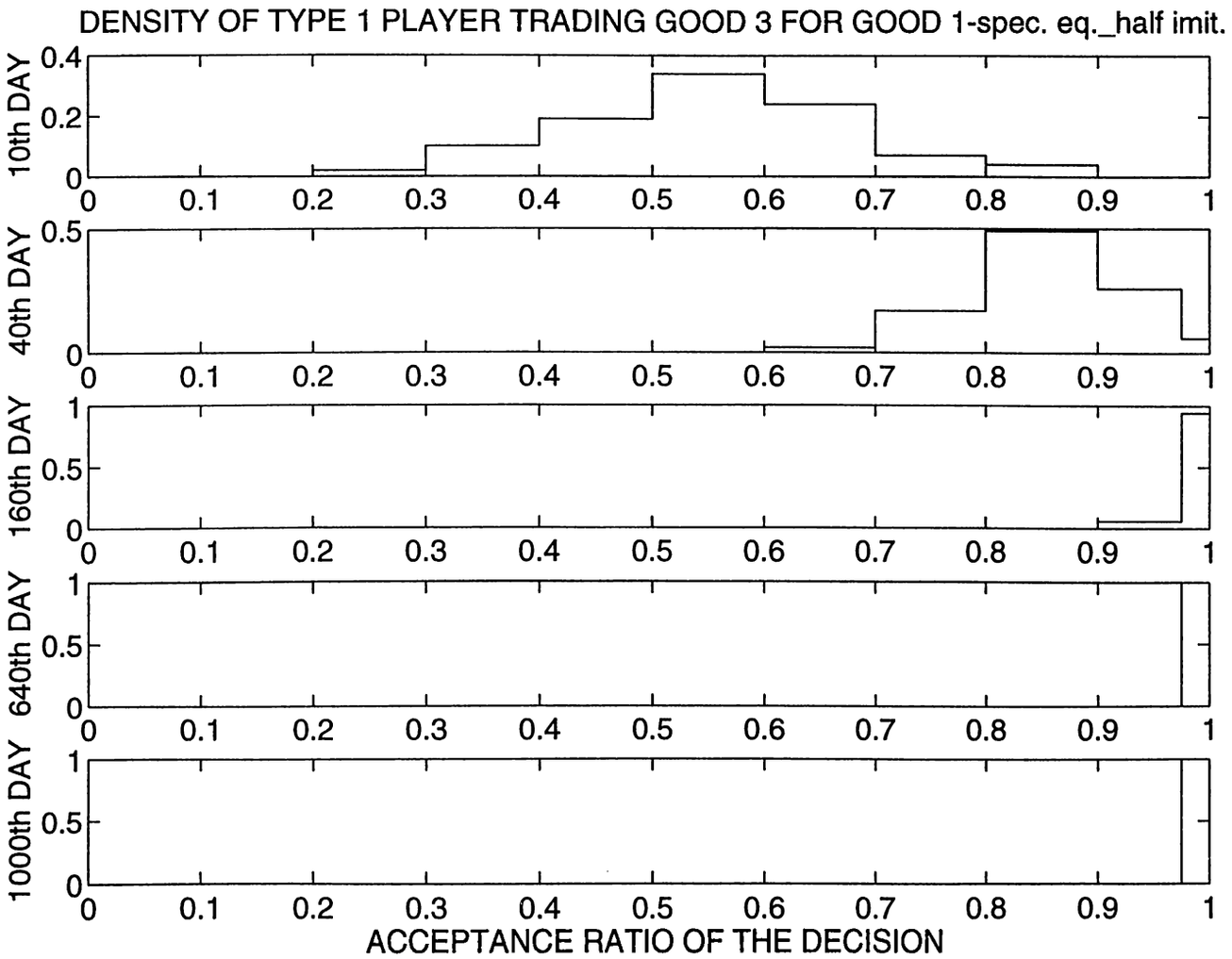

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 1-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

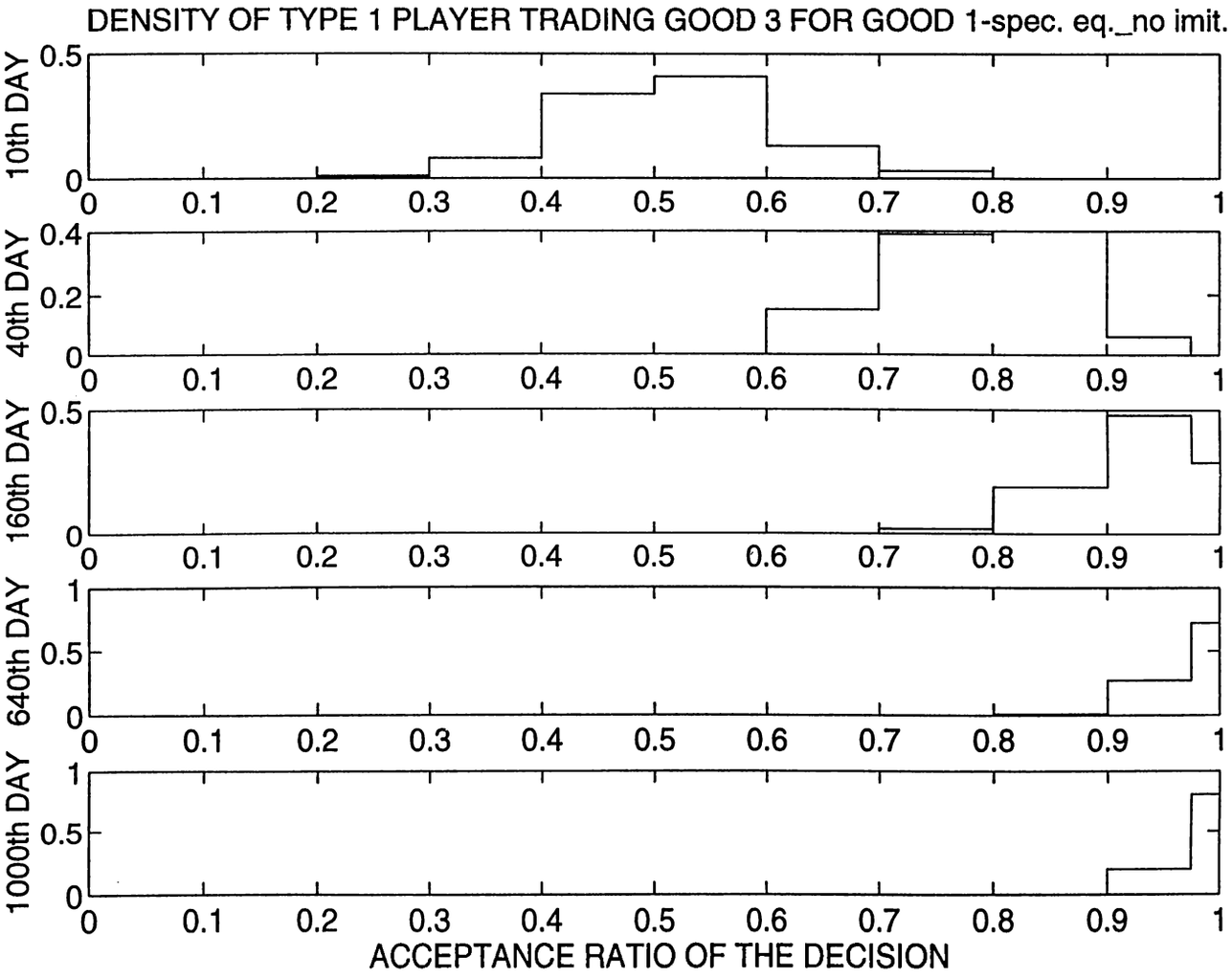

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

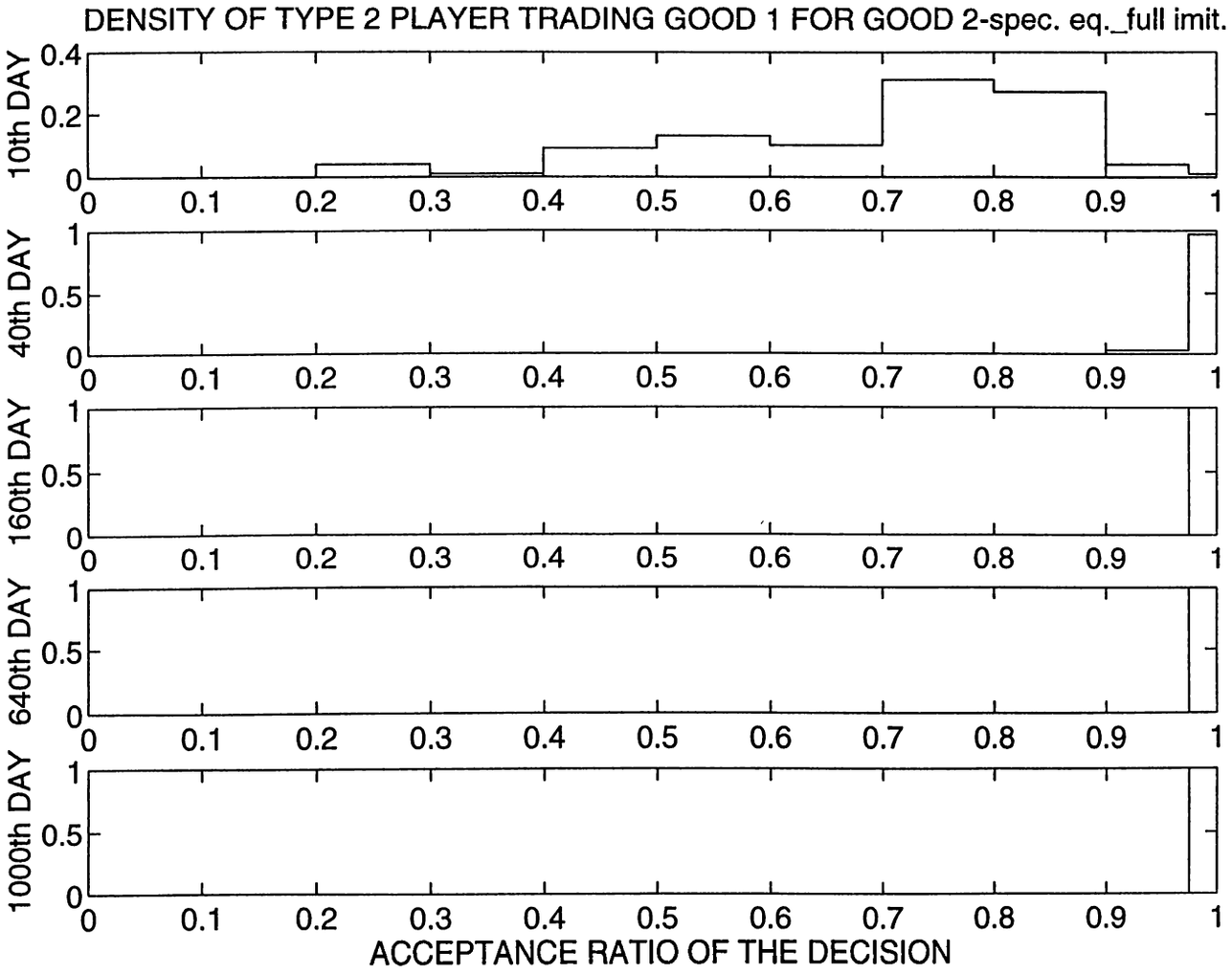

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

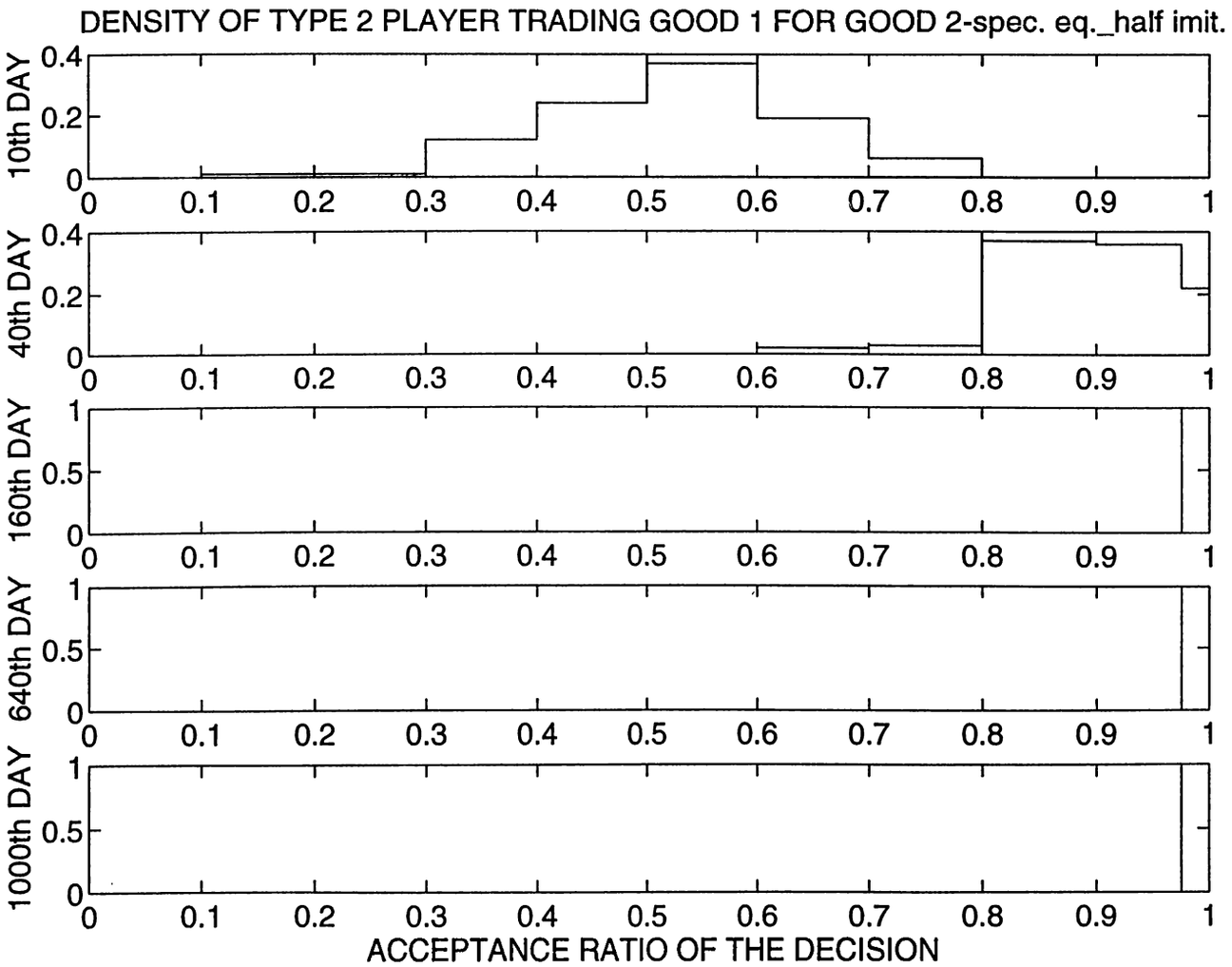

DENSITY OF TYPE 1 PLAYER TRADING GOOD 2 FOR GOOD 3-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

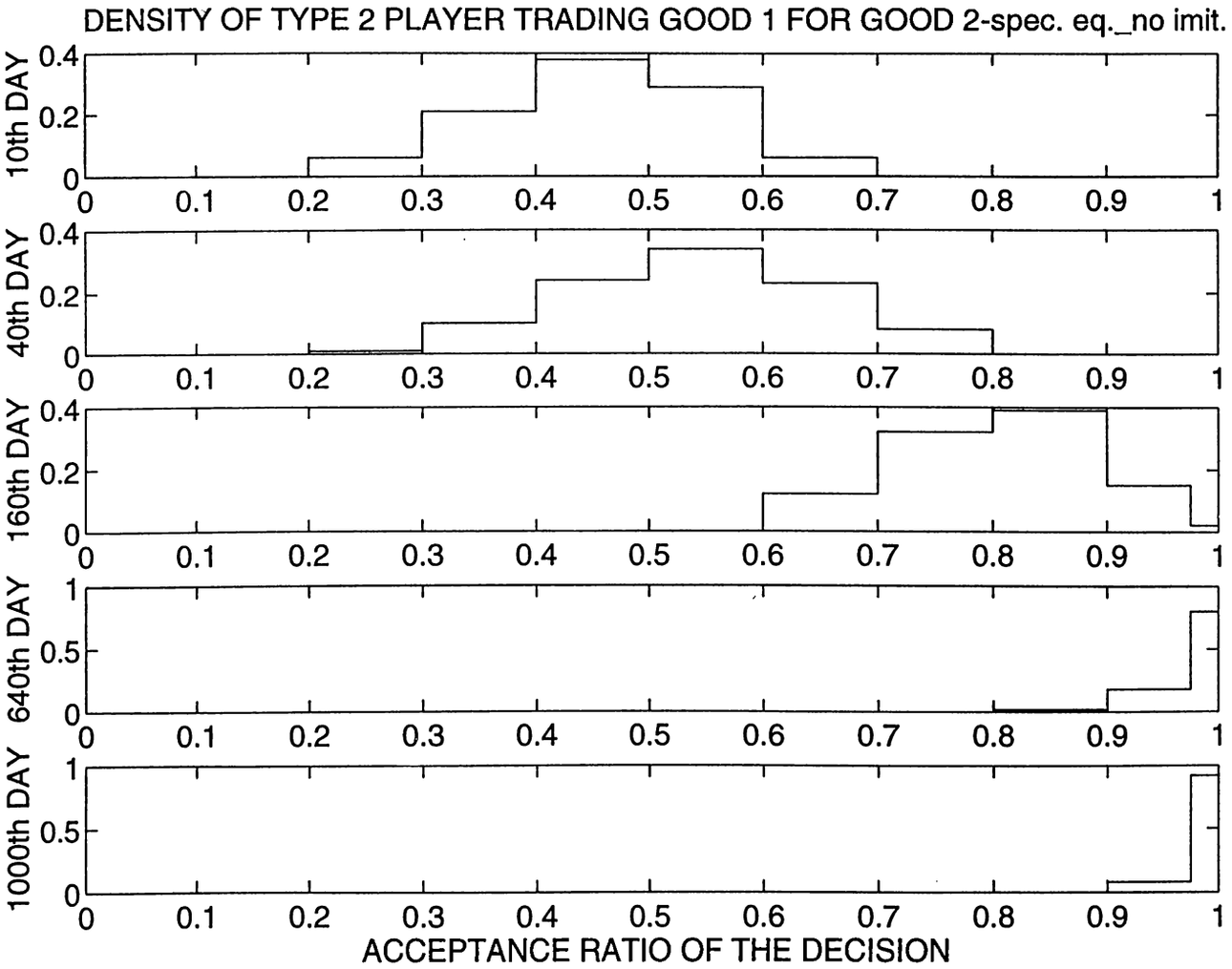

DENSITY OF TYPE 1 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

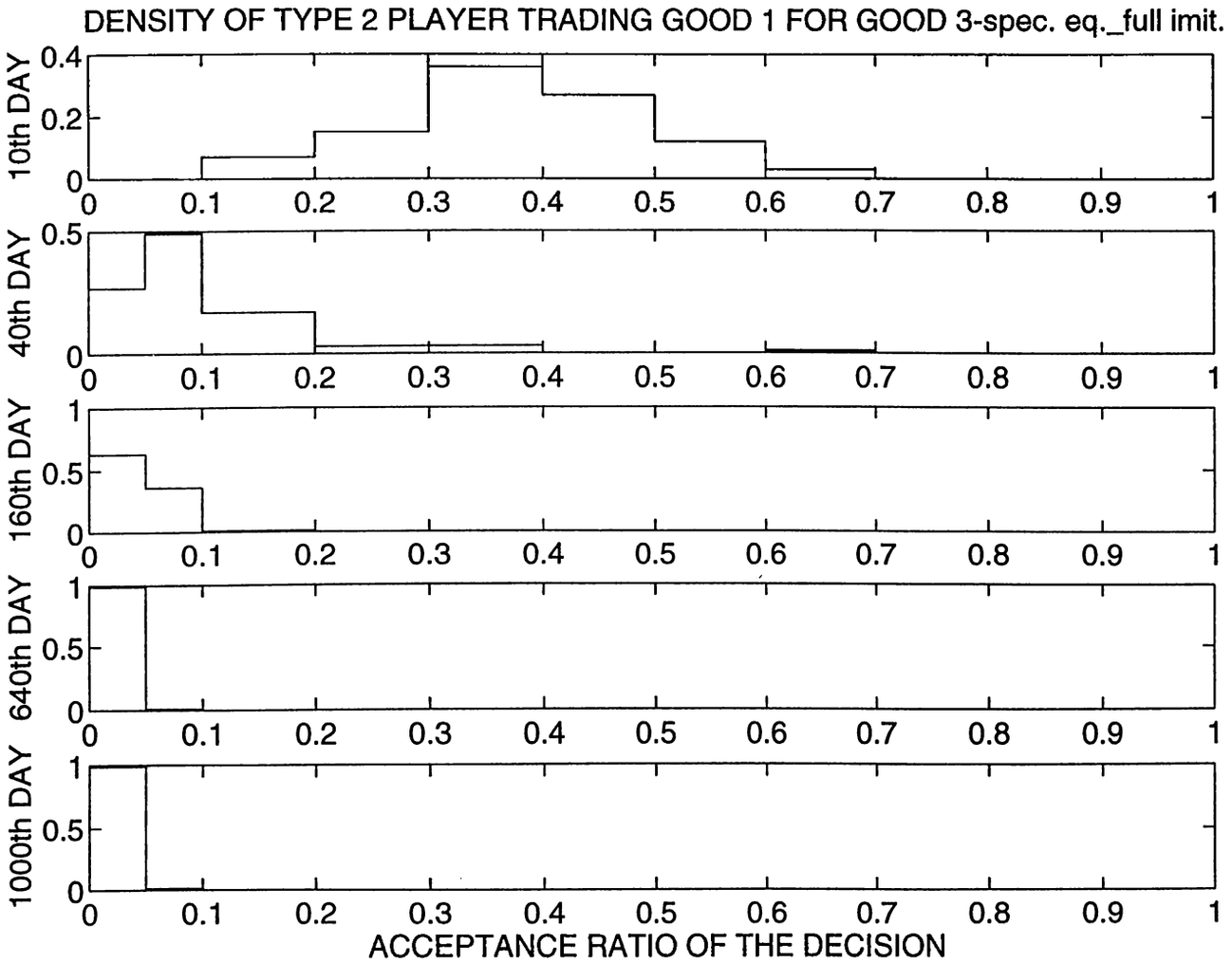

DENSITY OF TYPE 1 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

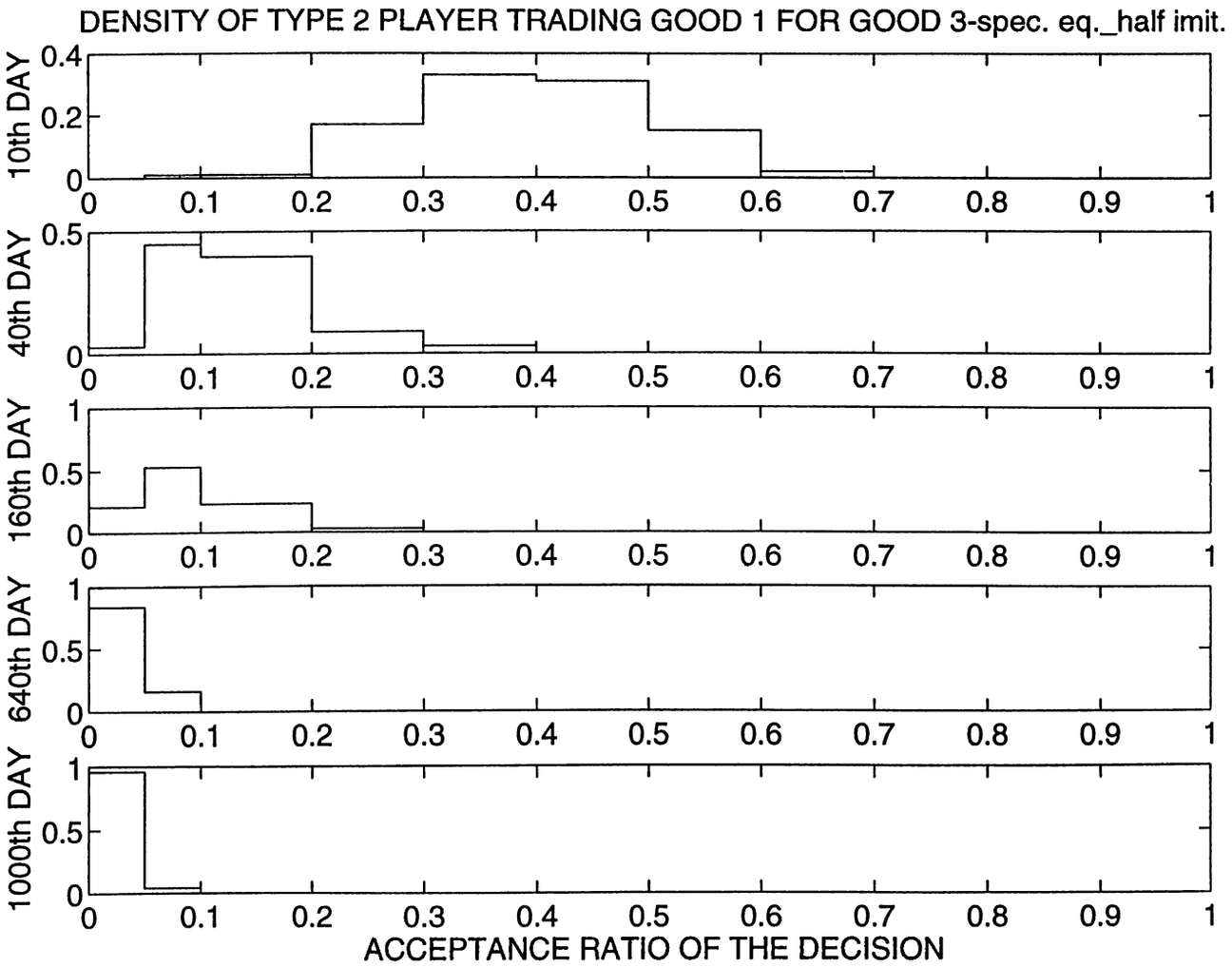

DENSITY OF TYPE 1 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

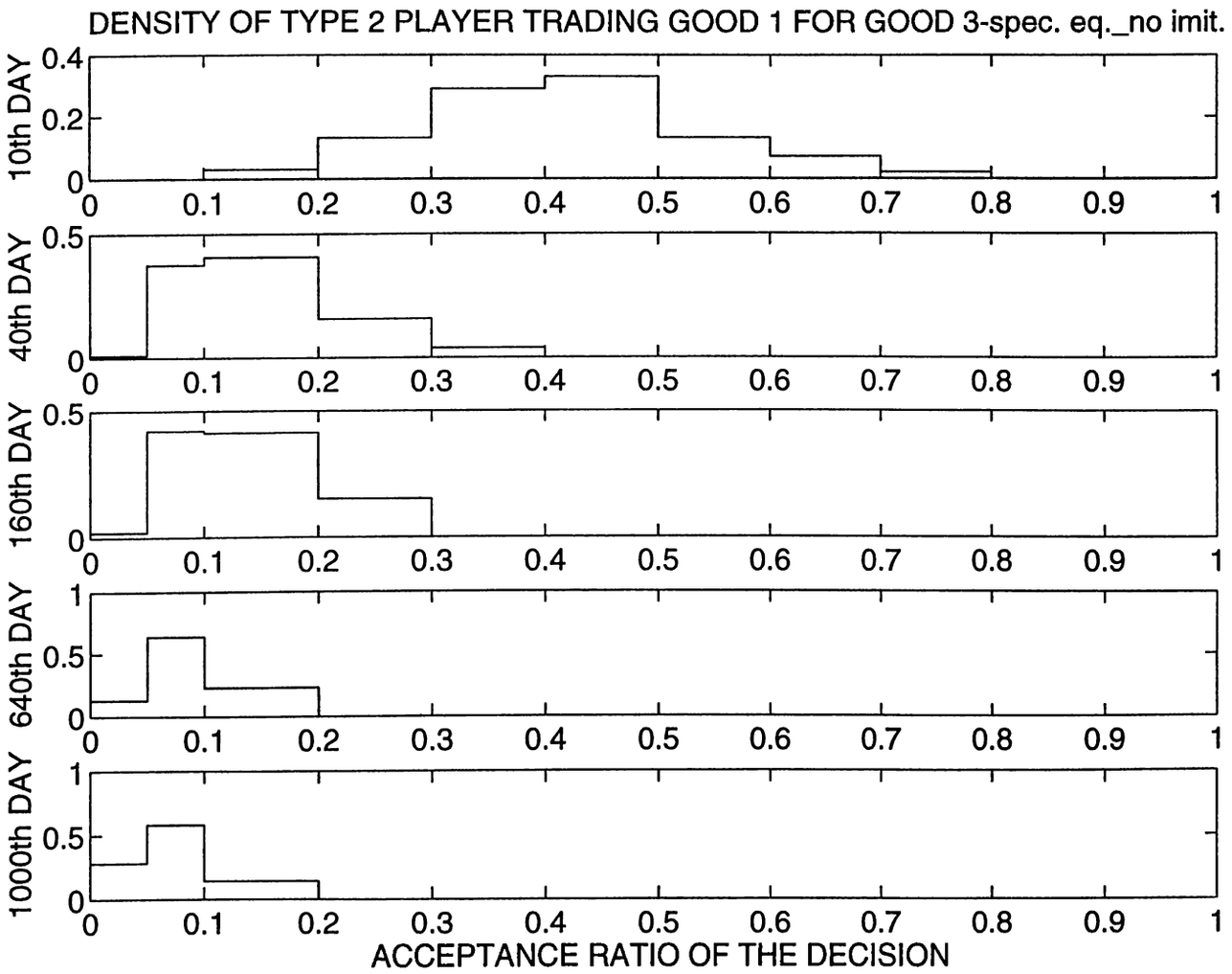

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 2-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

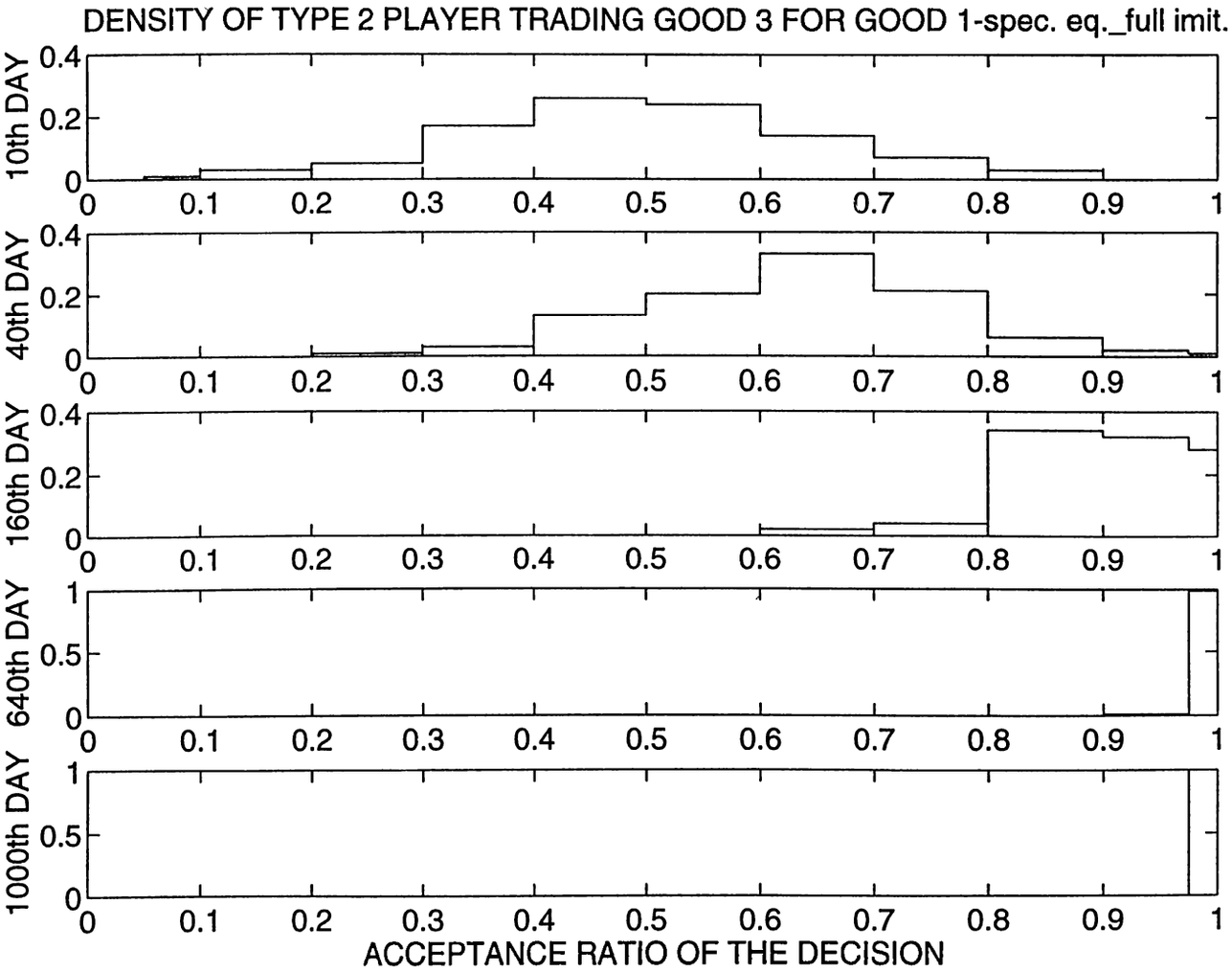

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 2-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

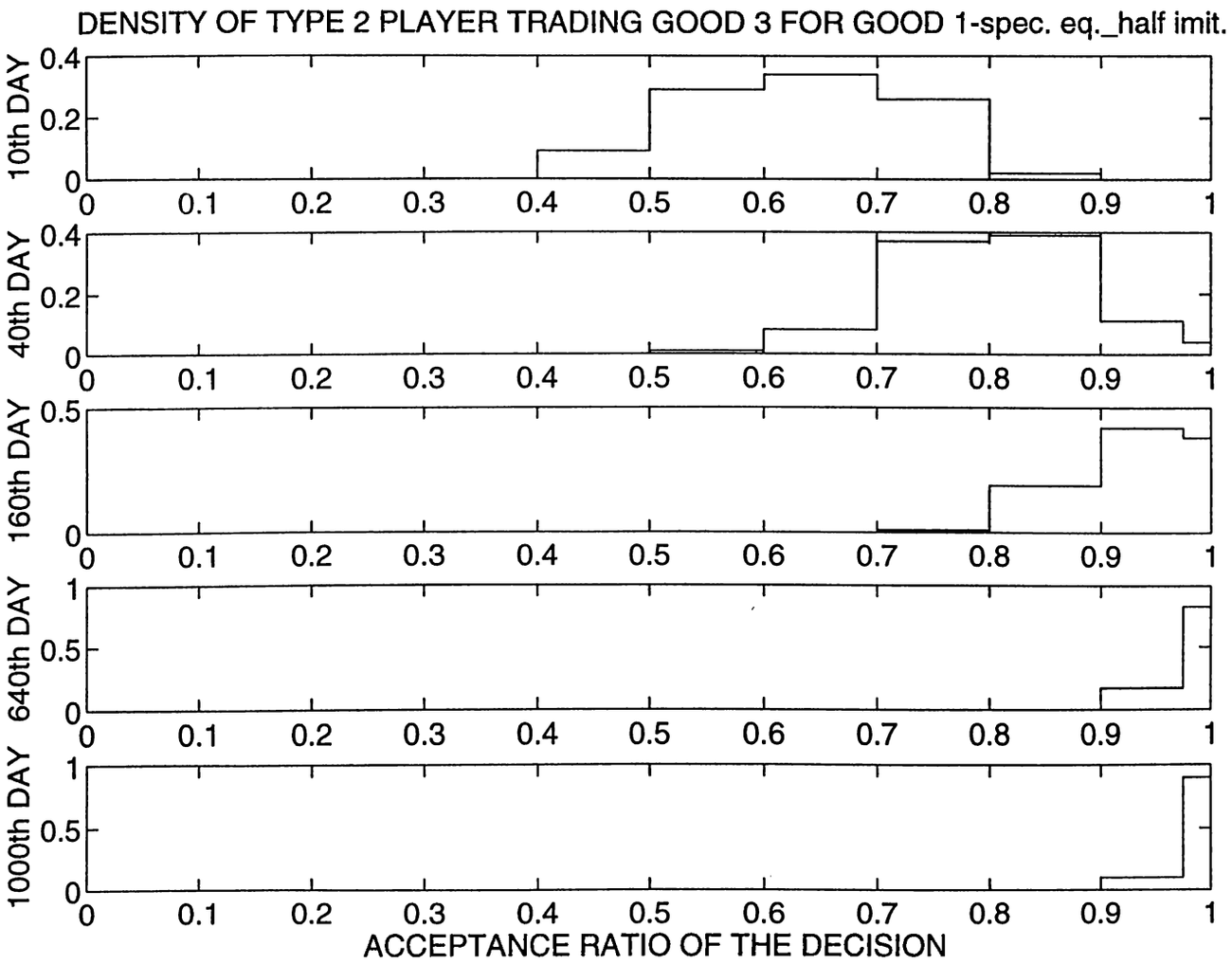

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 2-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

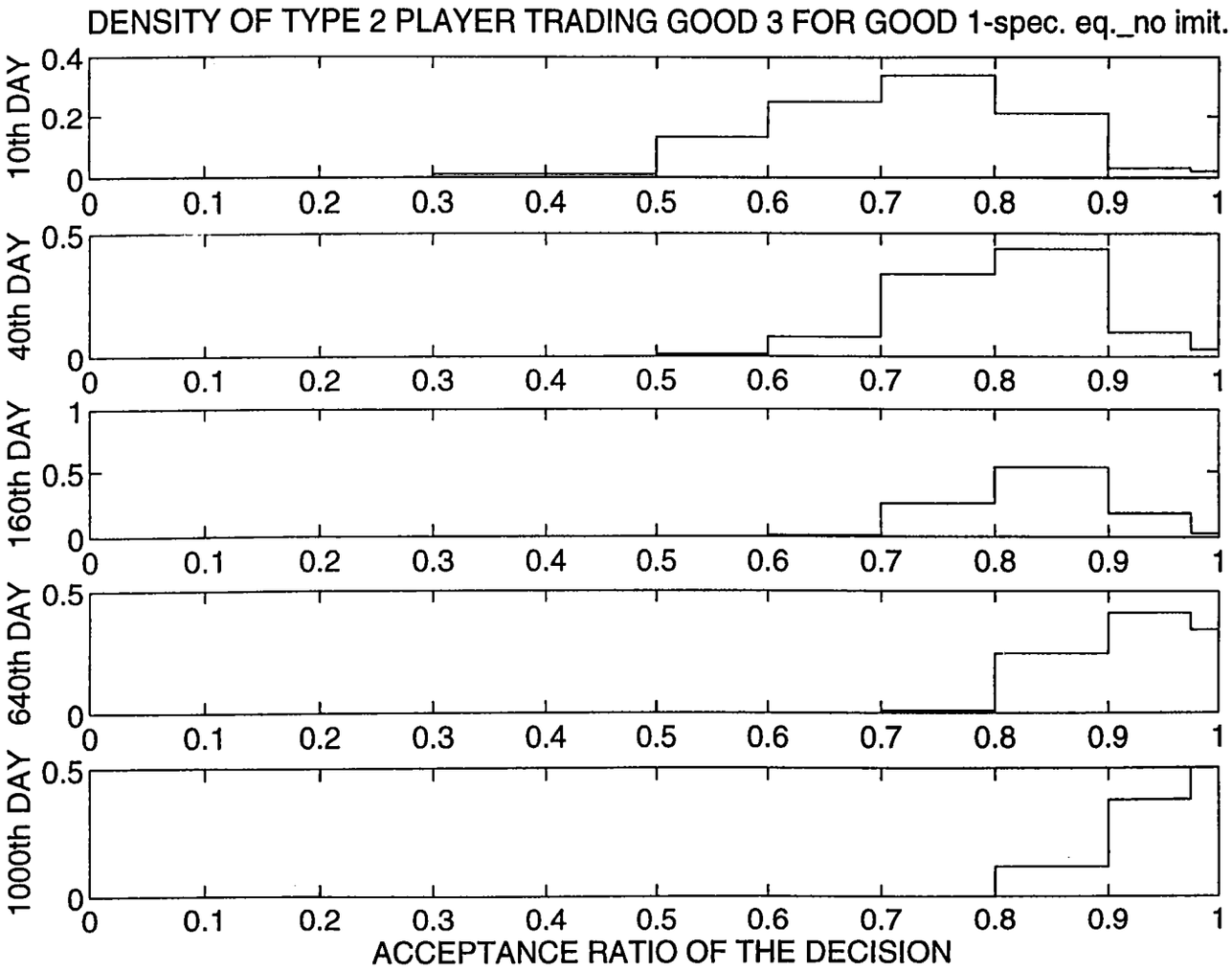

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 3-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

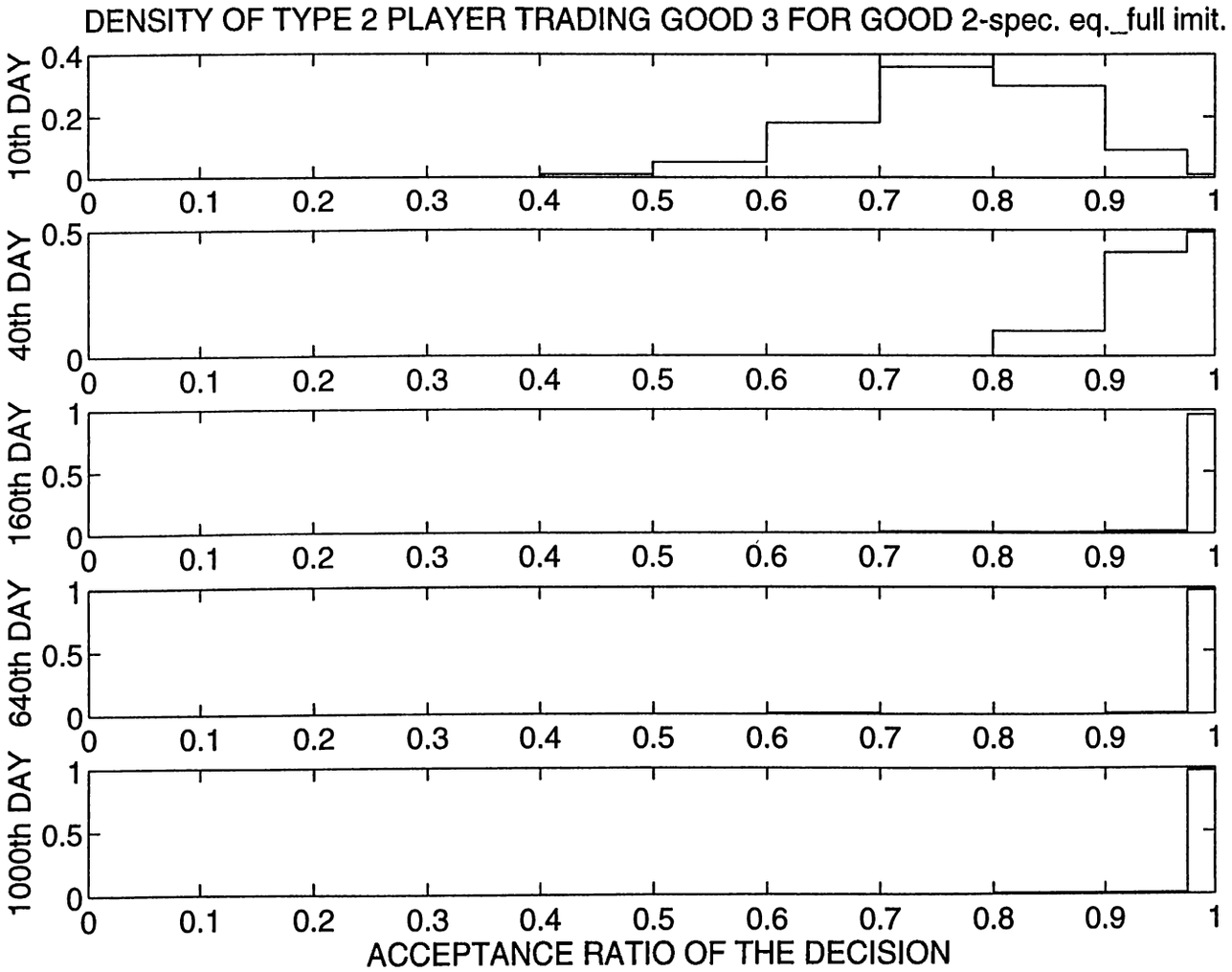

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 3-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

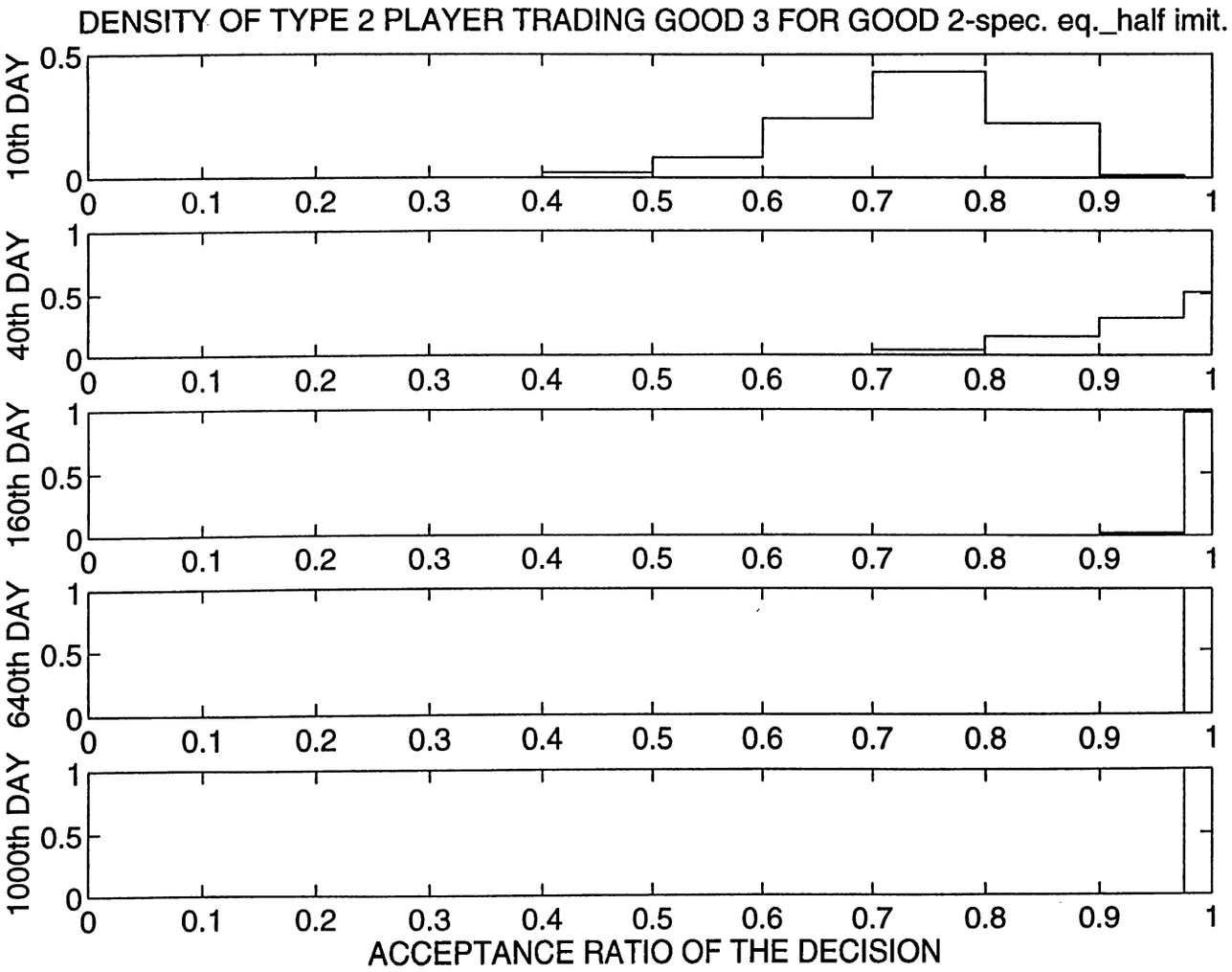

DENSITY OF TYPE 2 PLAYER TRADING GOOD 1 FOR GOOD 3-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

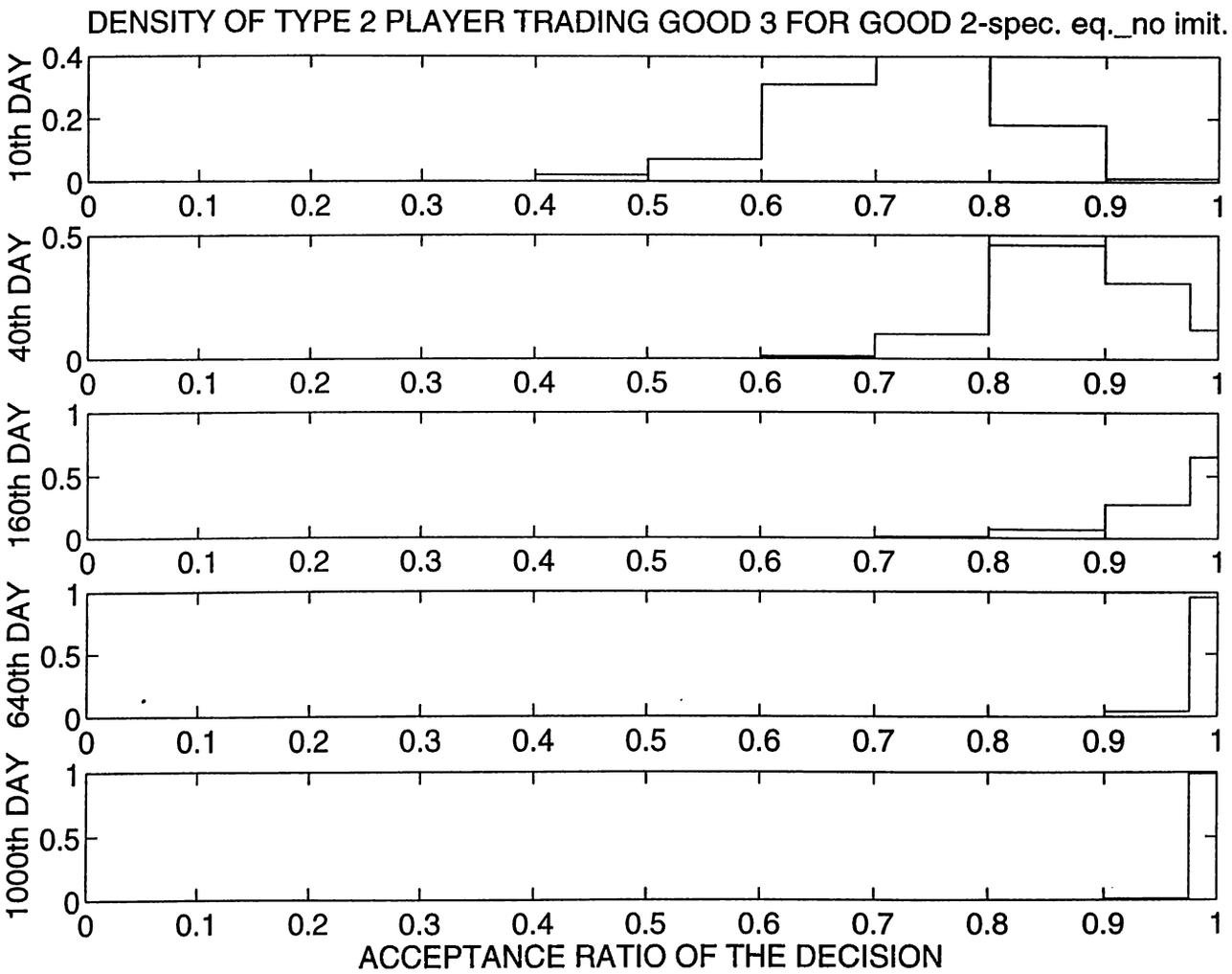

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

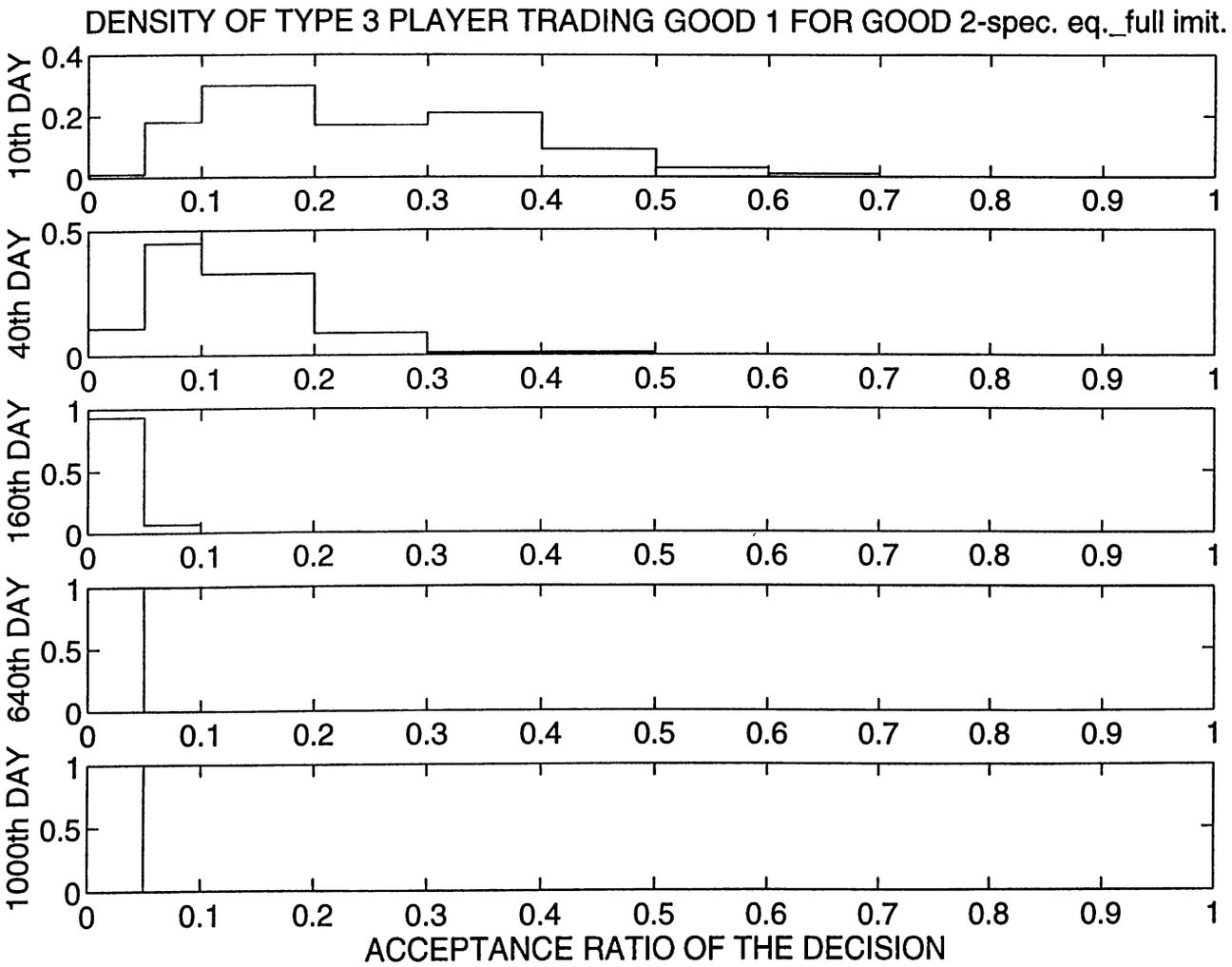

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

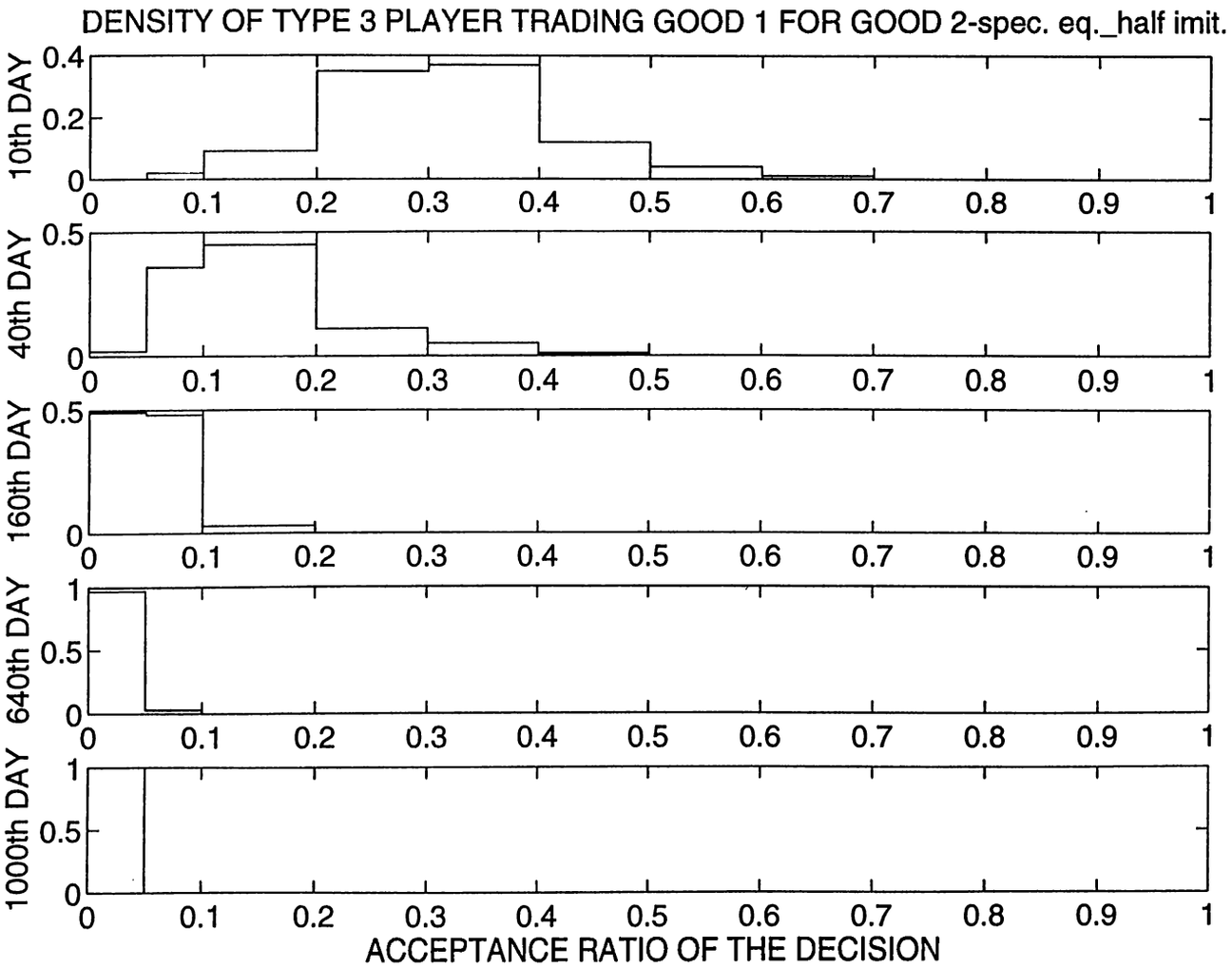

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 1-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

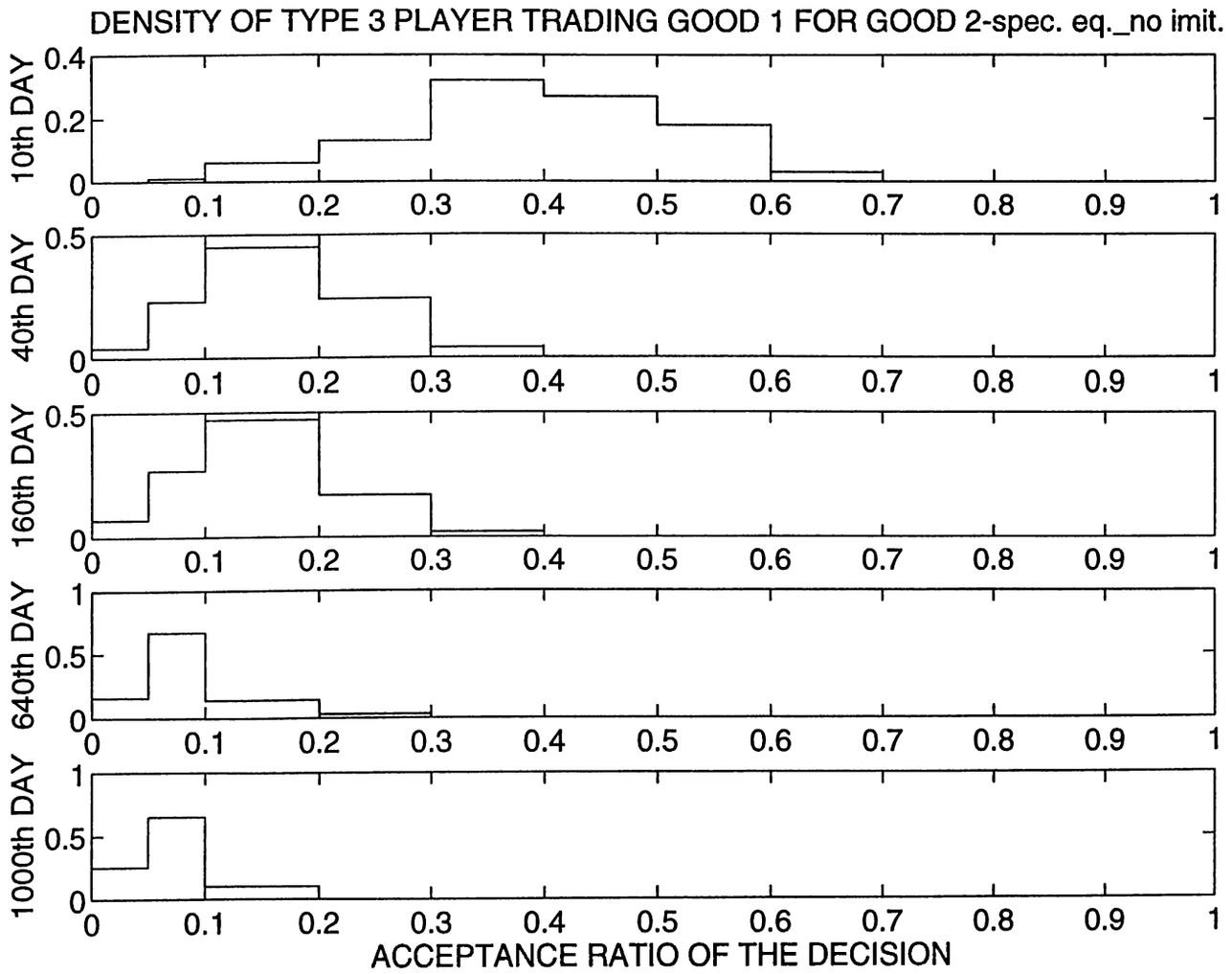

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 2-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

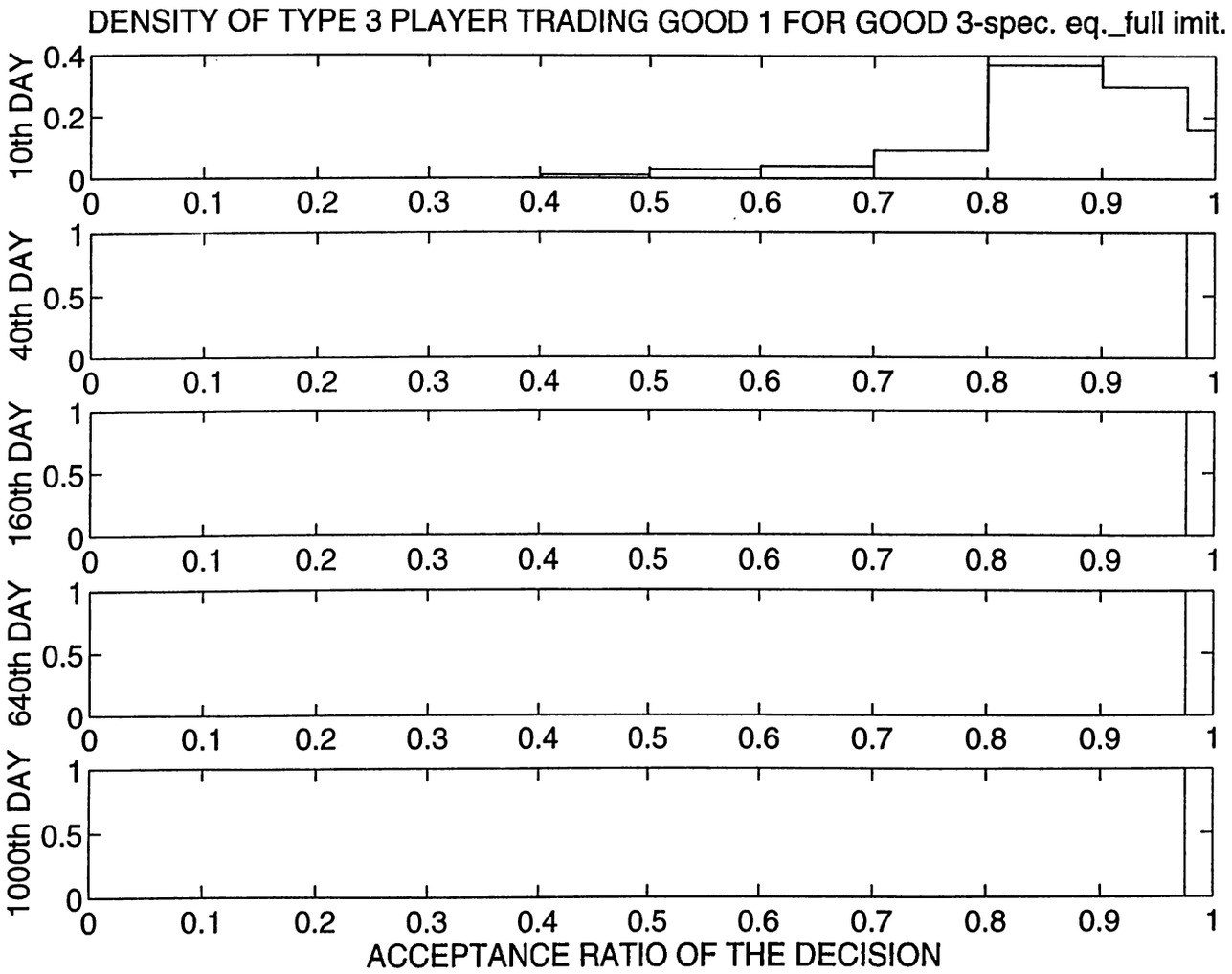

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 2-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

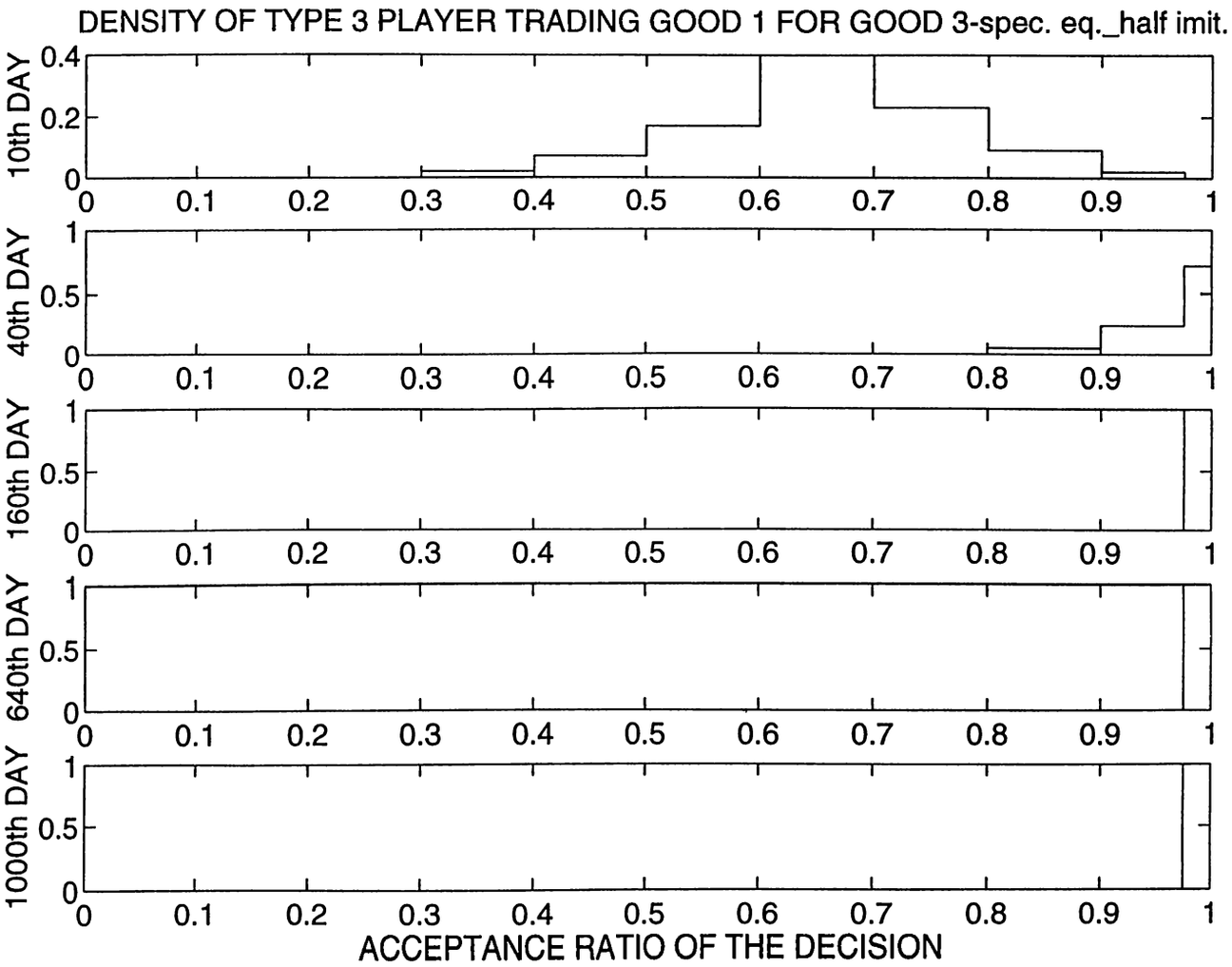

DENSITY OF TYPE 2 PLAYER TRADING GOOD 3 FOR GOOD 2-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

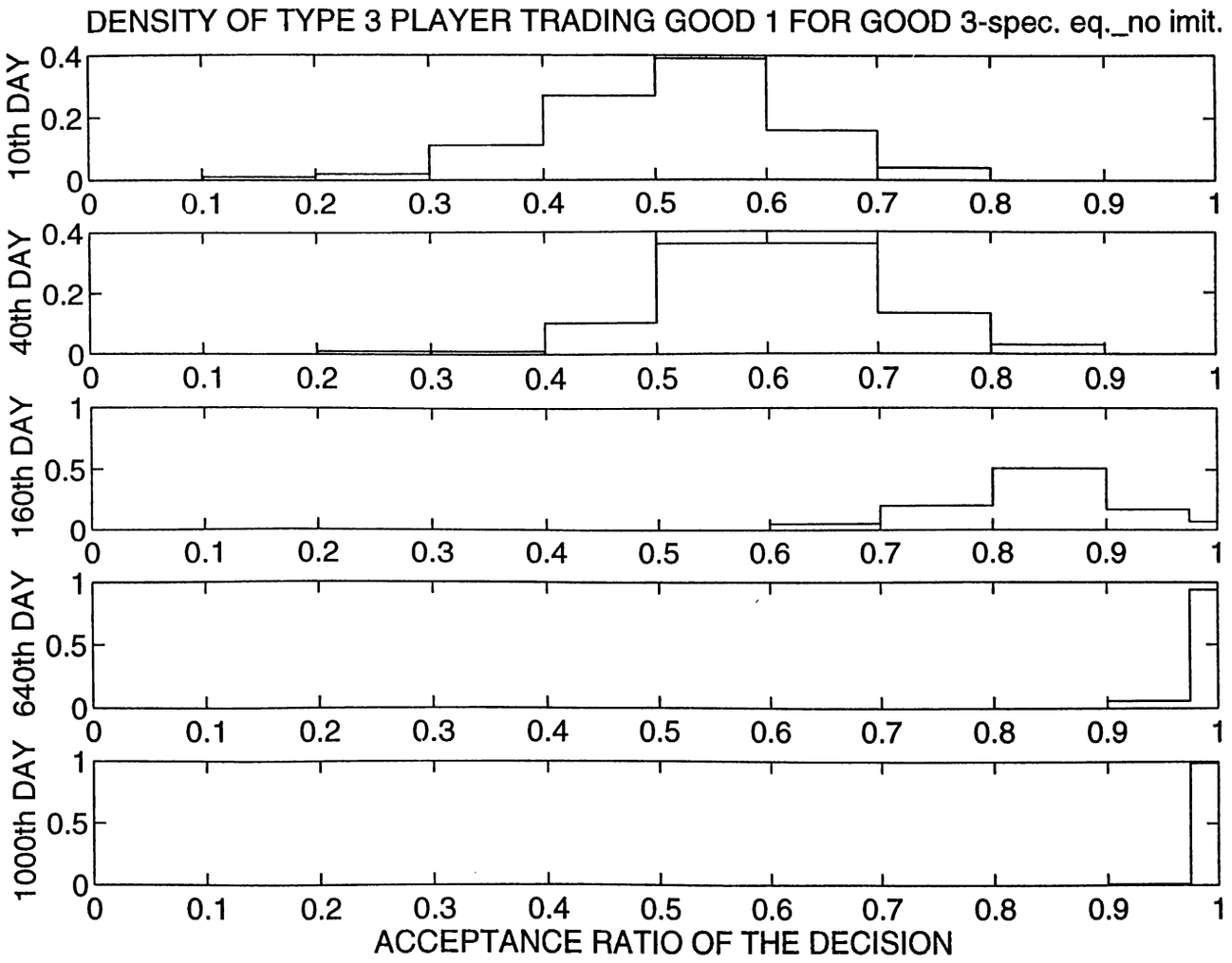

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 2-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

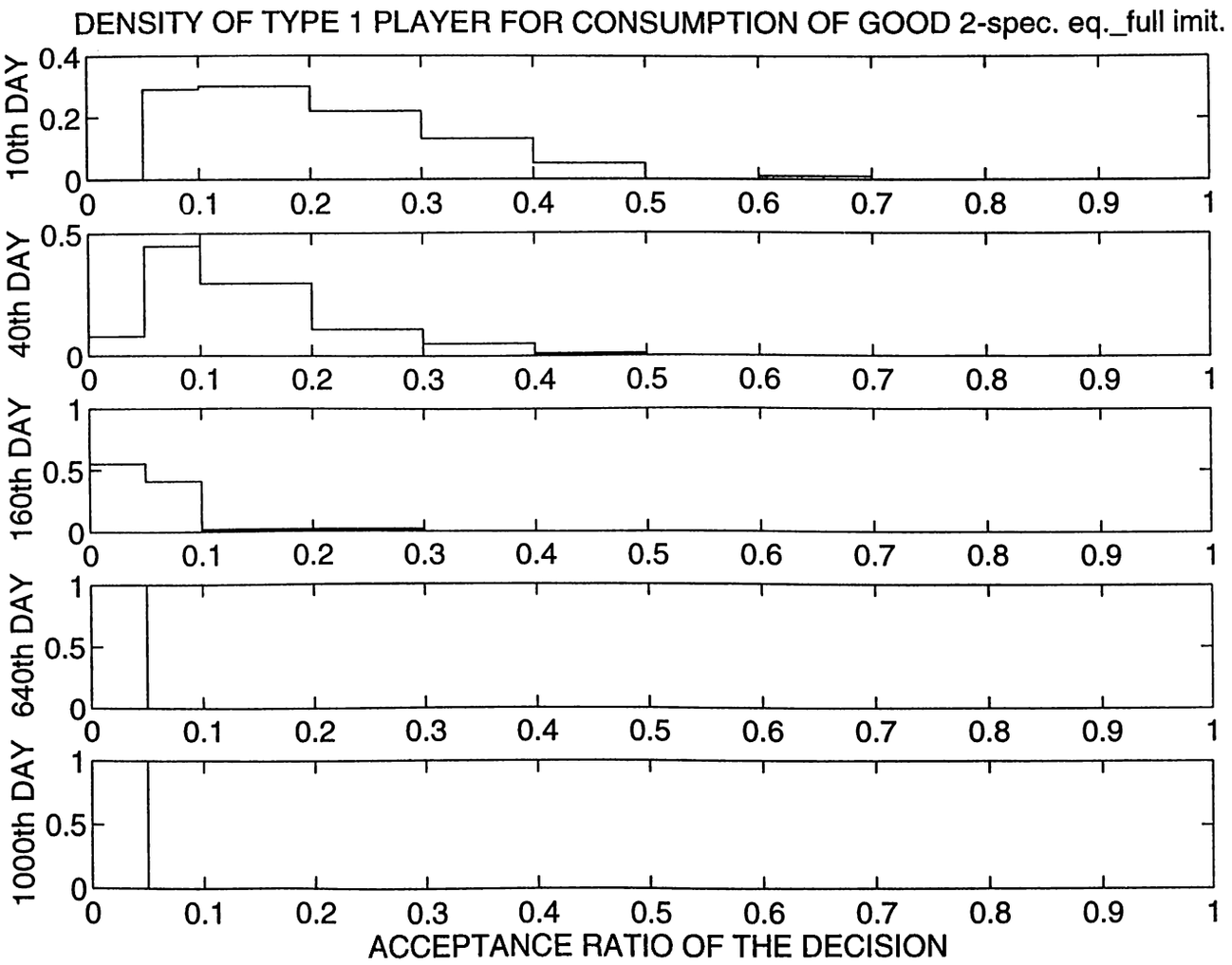

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 2-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

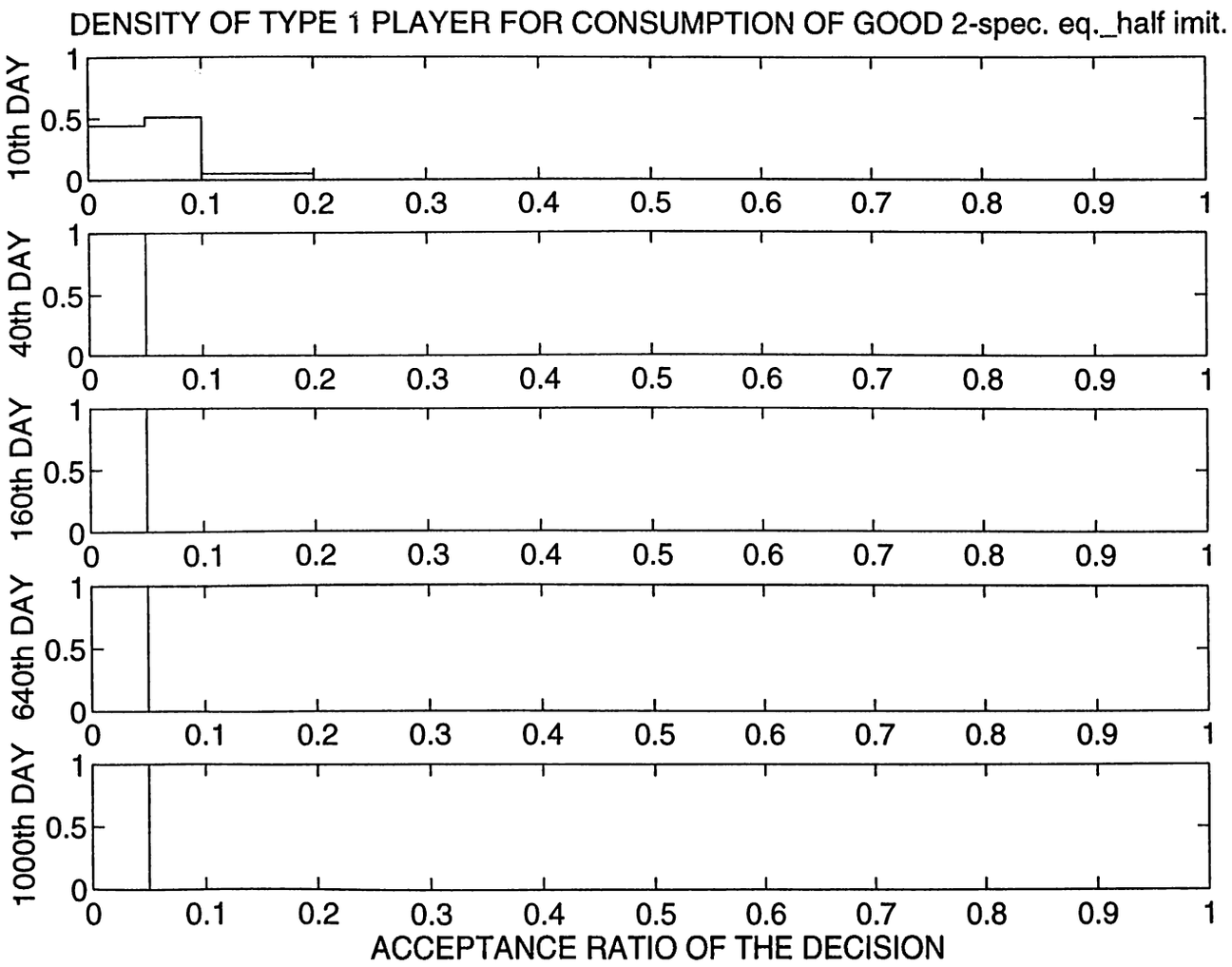

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 2-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

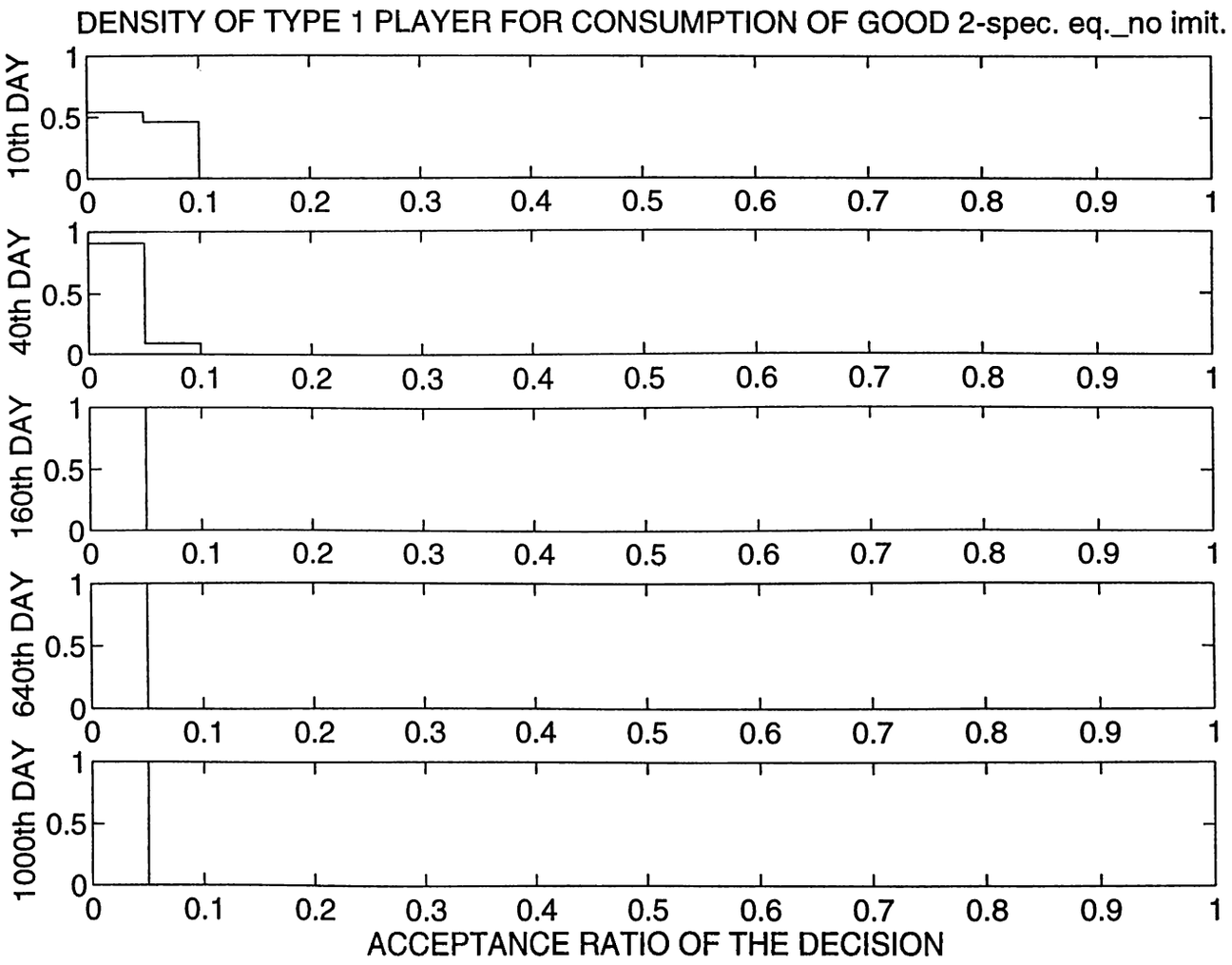

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 3-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

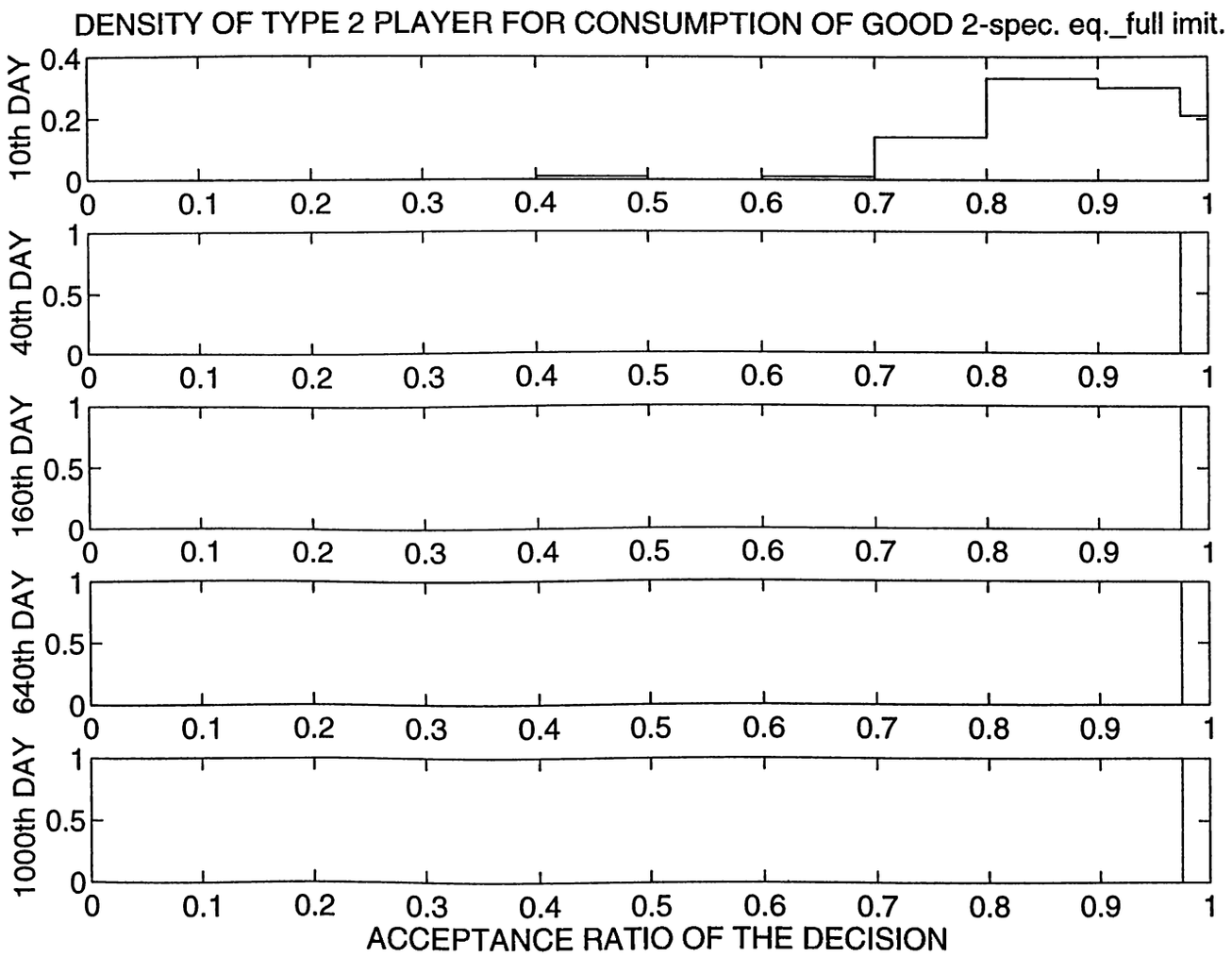

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 3-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

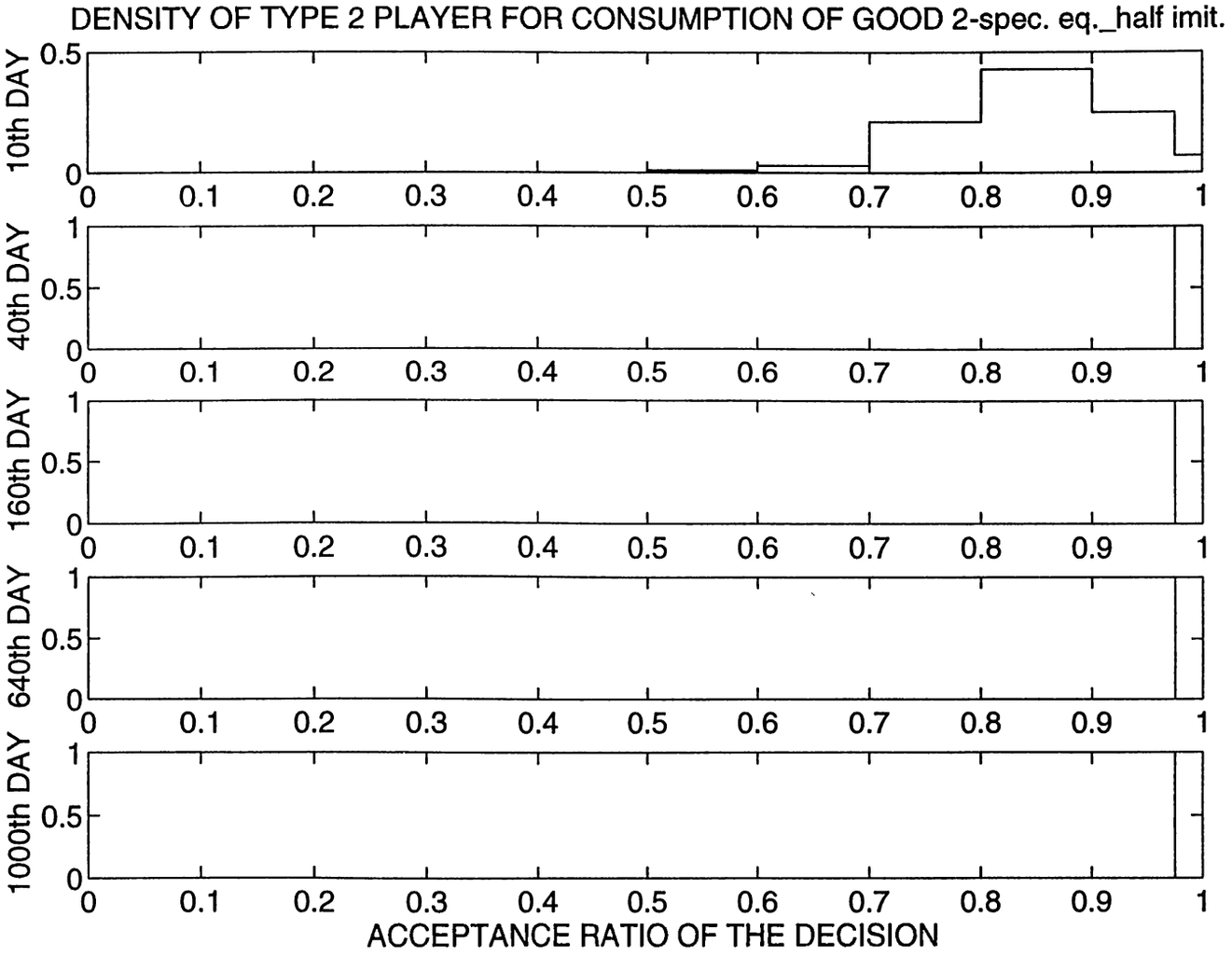

DENSITY OF TYPE 3 PLAYER TRADING GOOD 1 FOR GOOD 3-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

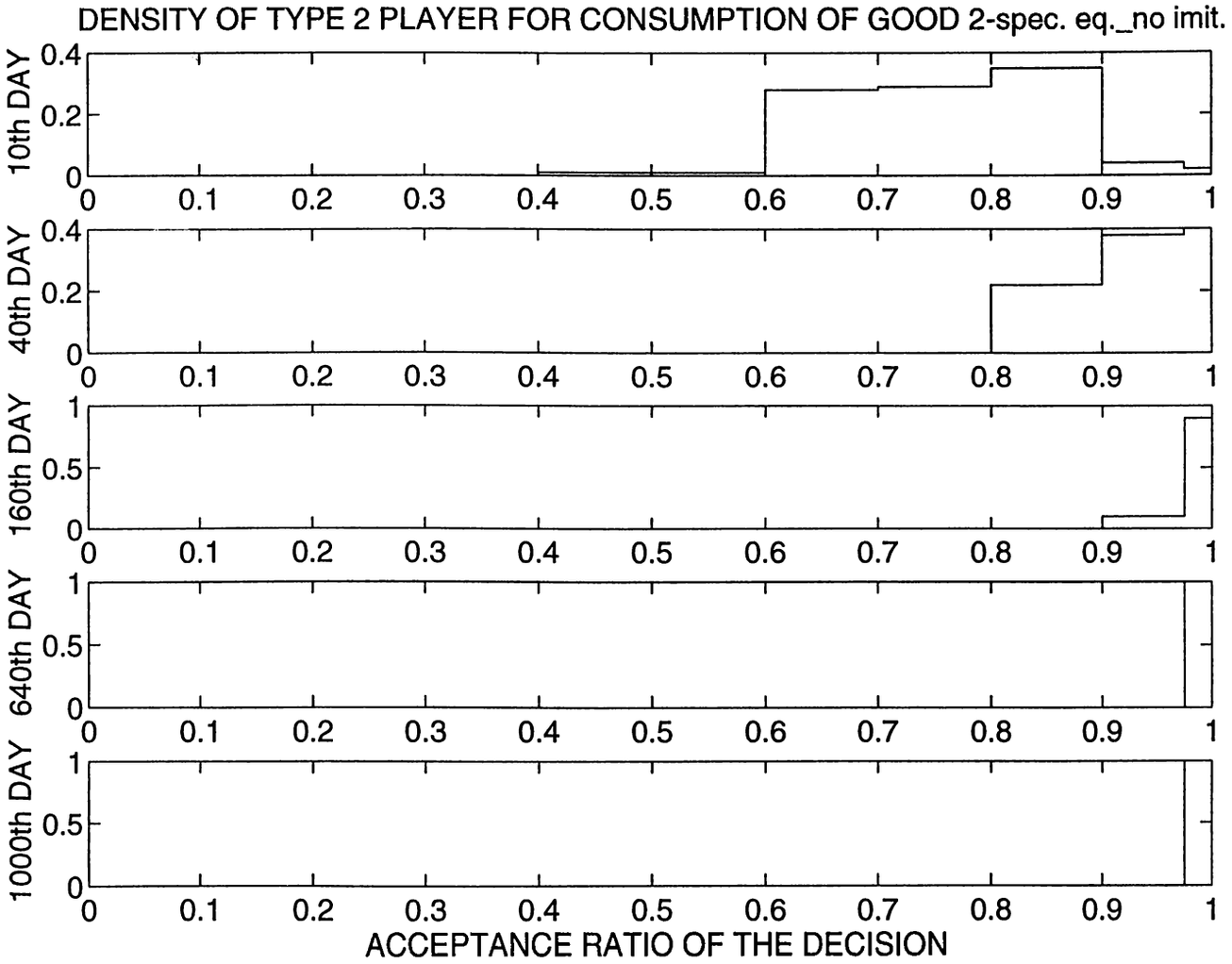

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 2-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

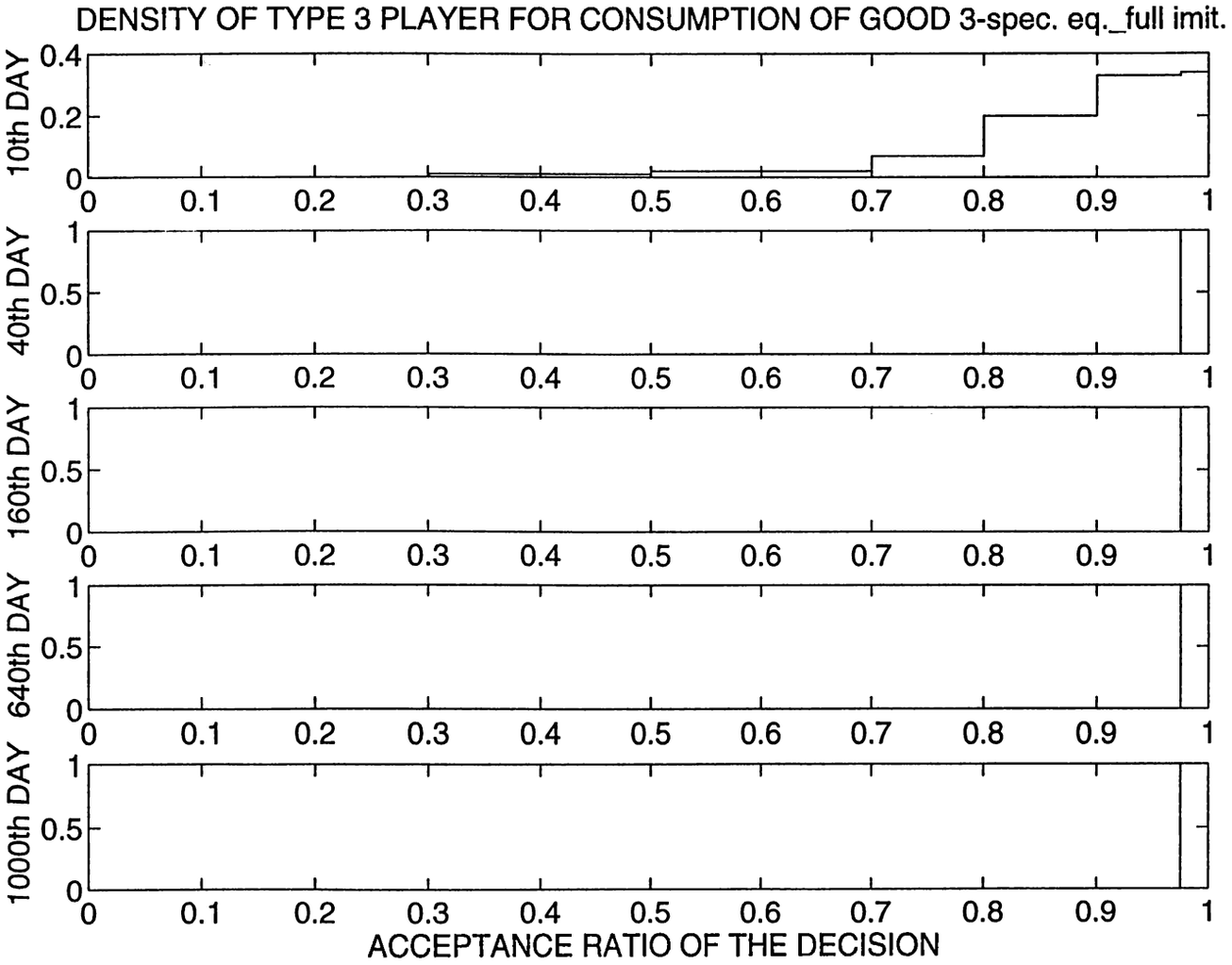

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 2-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
1
0.5
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

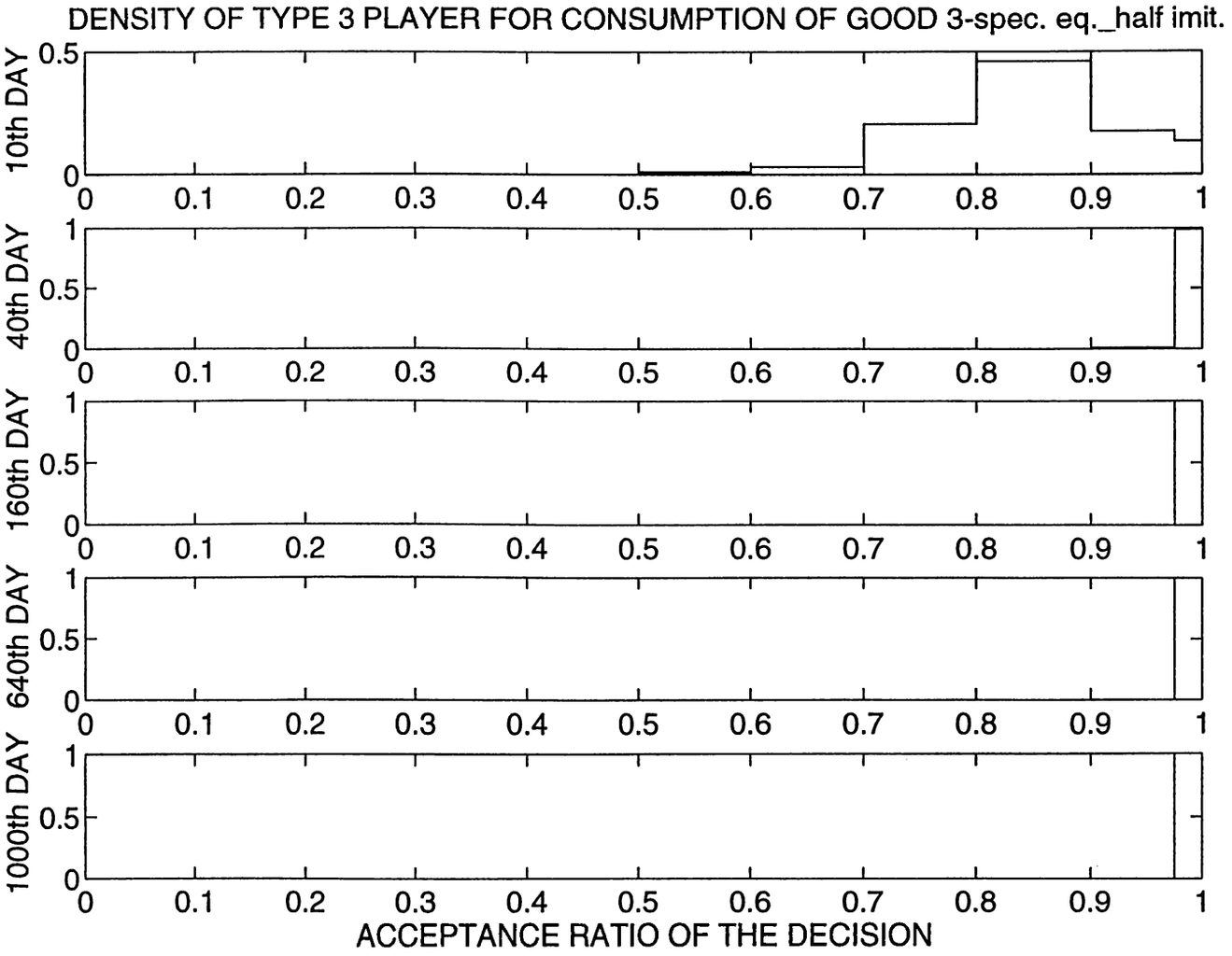

DENSITY OF TYPE 1 PLAYER FOR CONSUMPTION OF GOOD 2-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
1
0.5
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
ACCEPTANCE RATIO OF THE DECISION

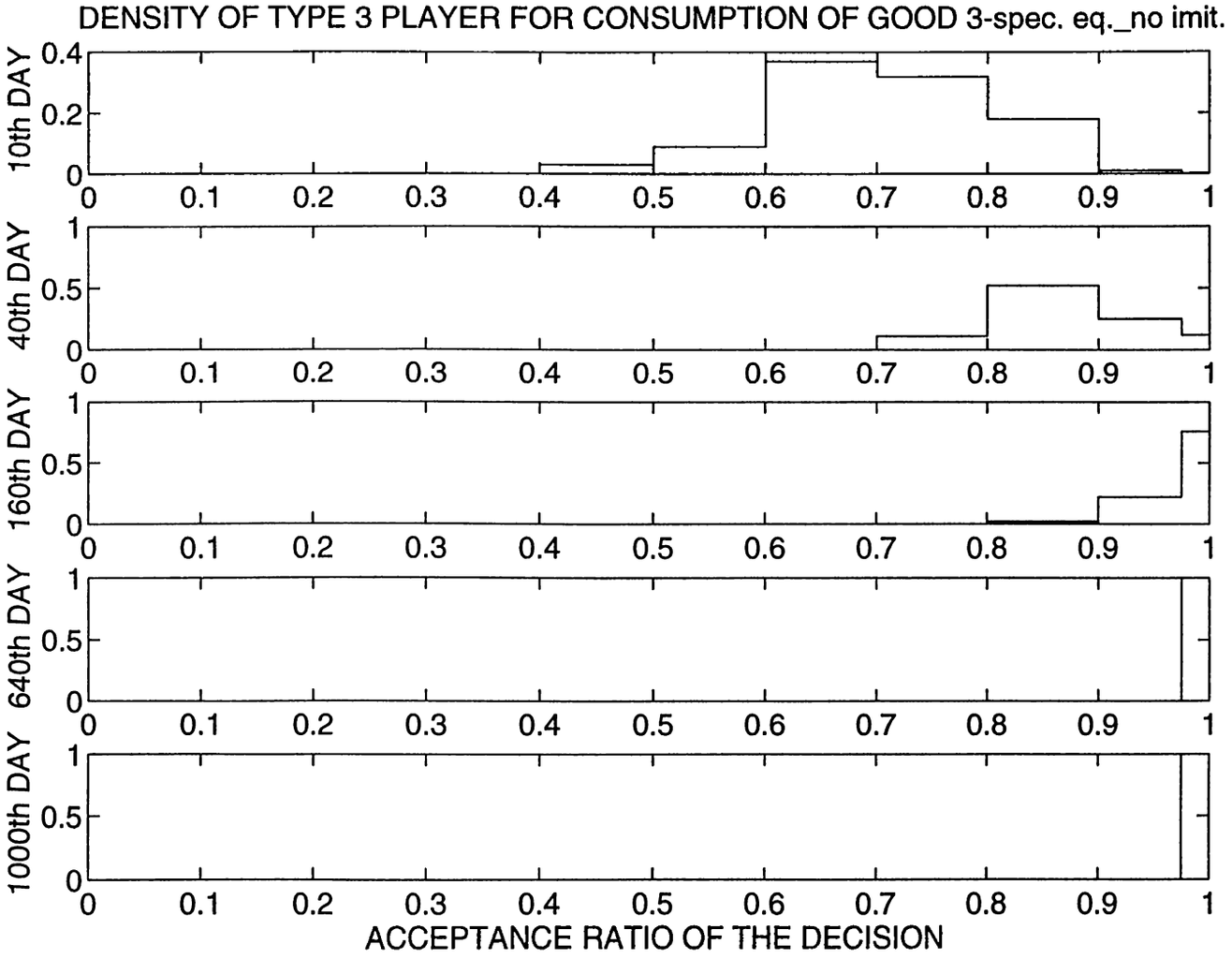

DENSITY OF TYPE 2 PLAYER FOR CONSUMPTION OF GOOD 2-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

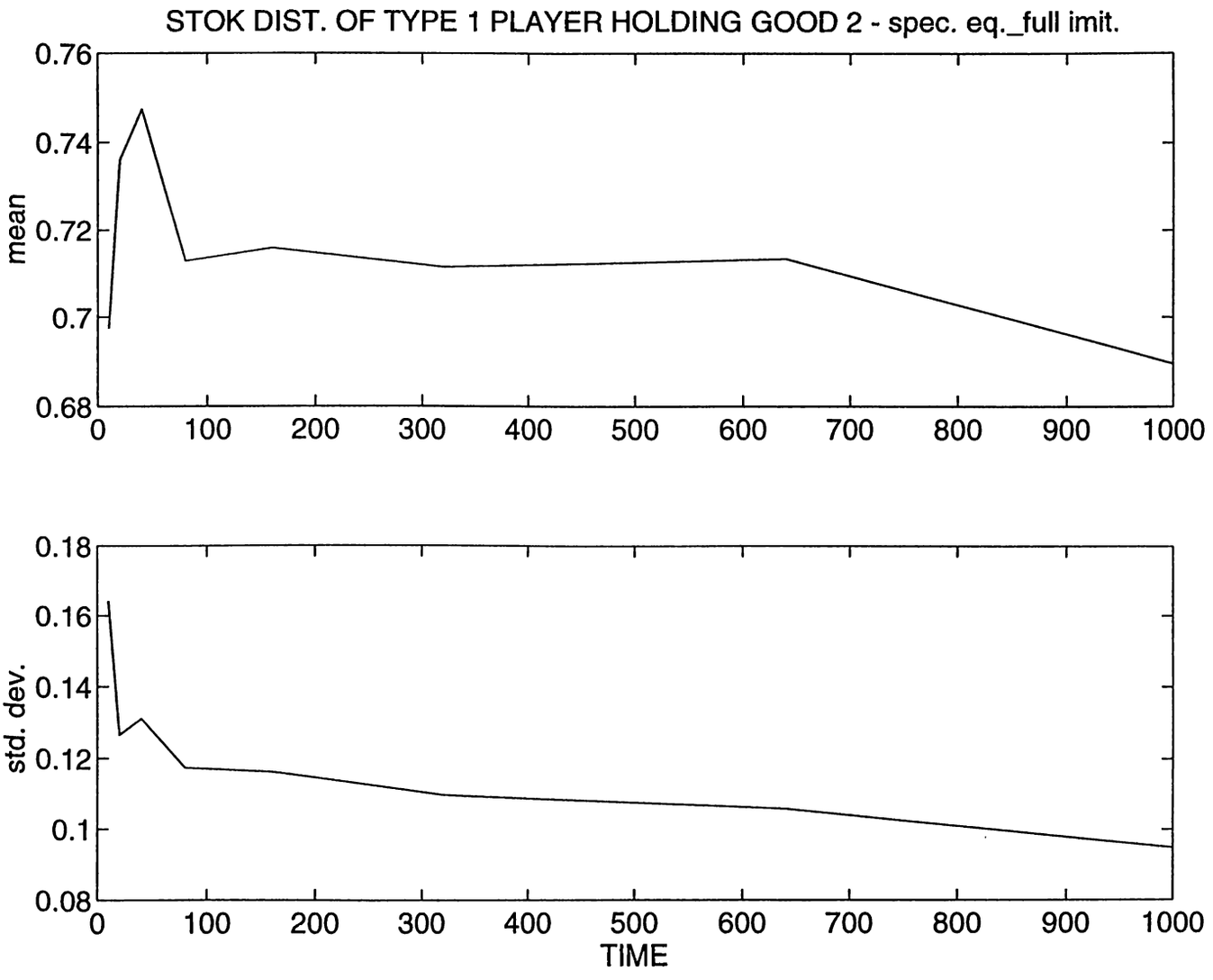

DENSITY OF TYPE 2 PLAYER FOR CONSUMPTION OF GOOD 2-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

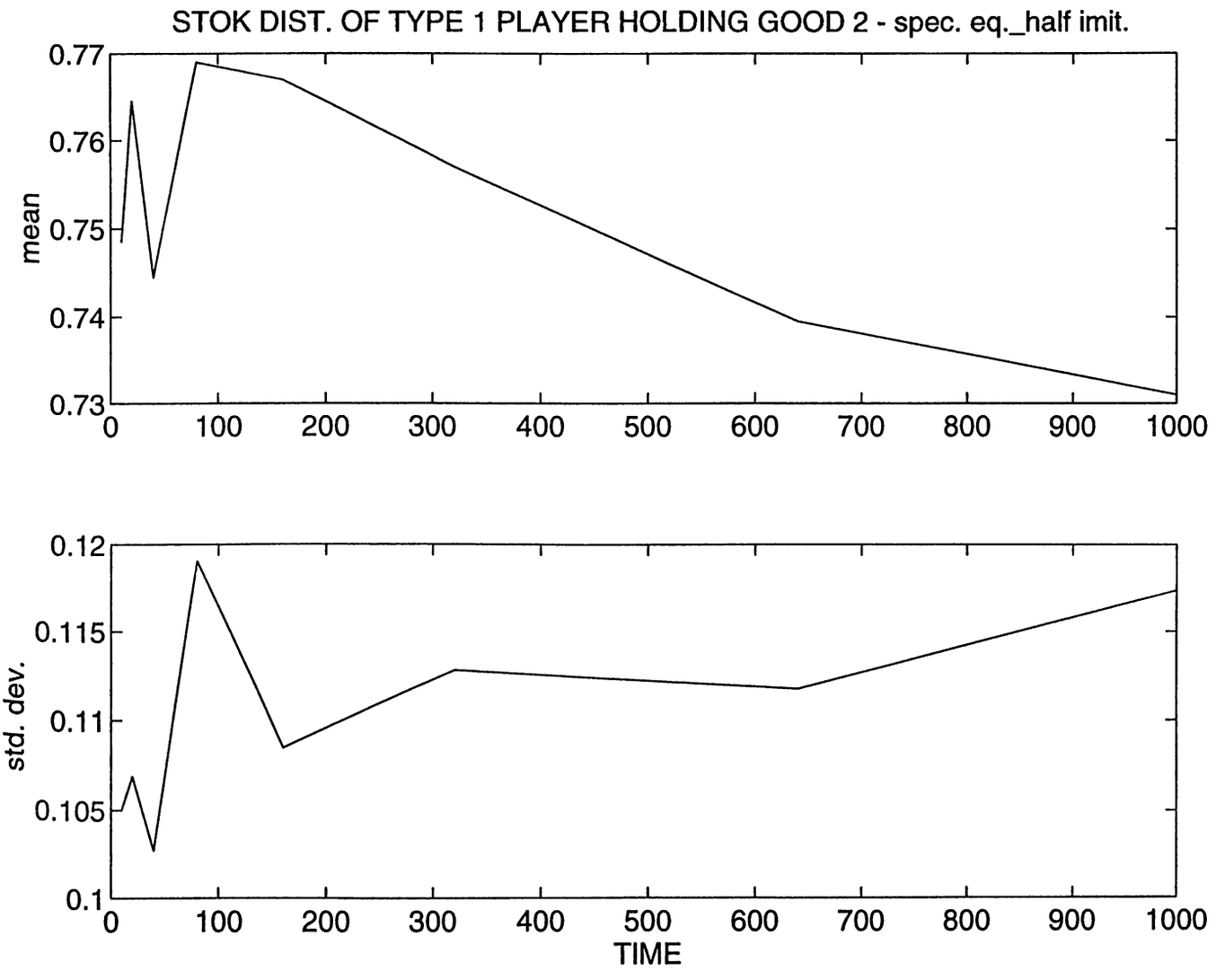

DENSITY OF TYPE 2 PLAYER FOR CONSUMPTION OF GOOD 2-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
0
1
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

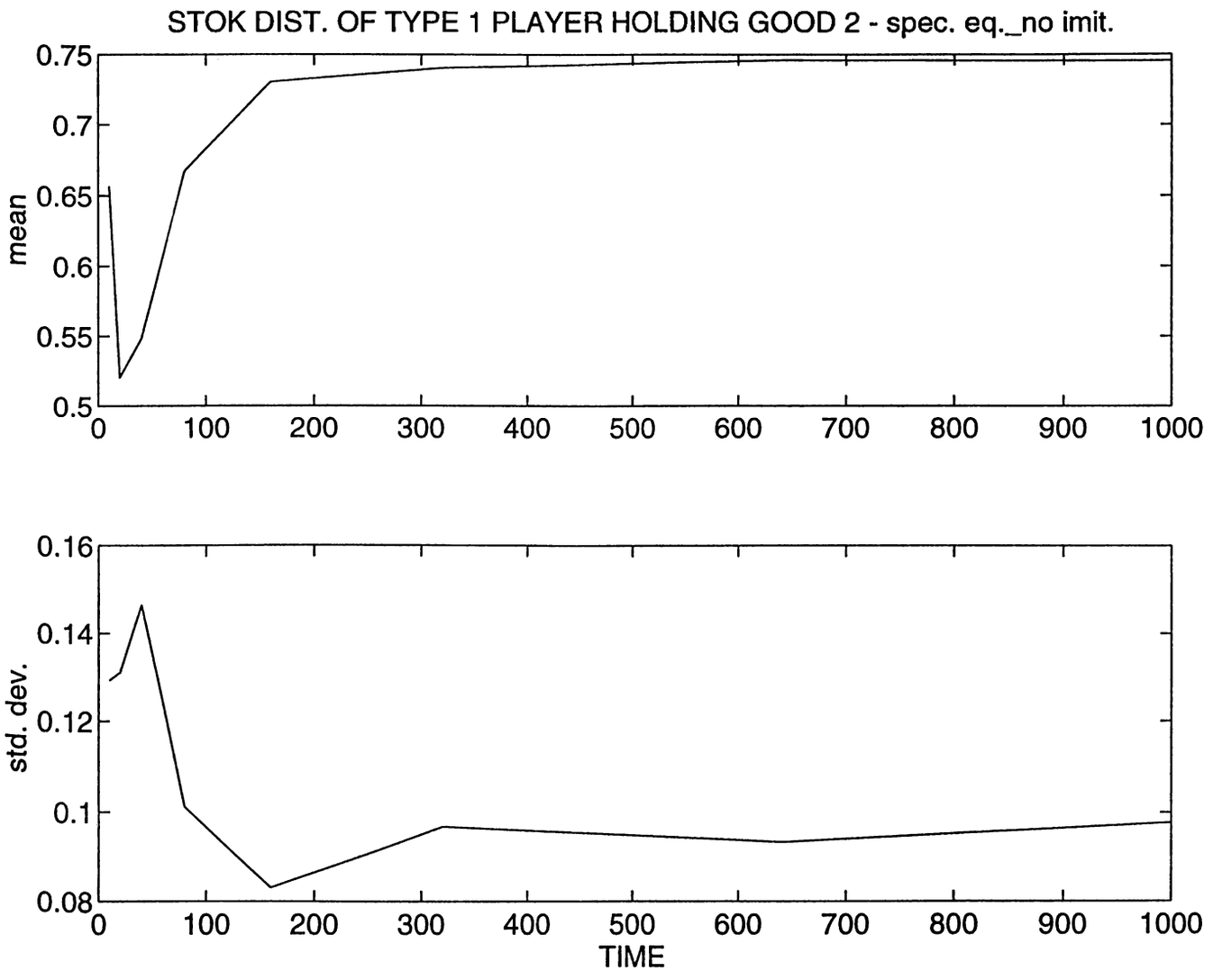

DENSITY OF TYPE 3 PLAYER FOR CONSUMPTION OF GOOD 3-spec. eq._full imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.4
0.2
1
0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

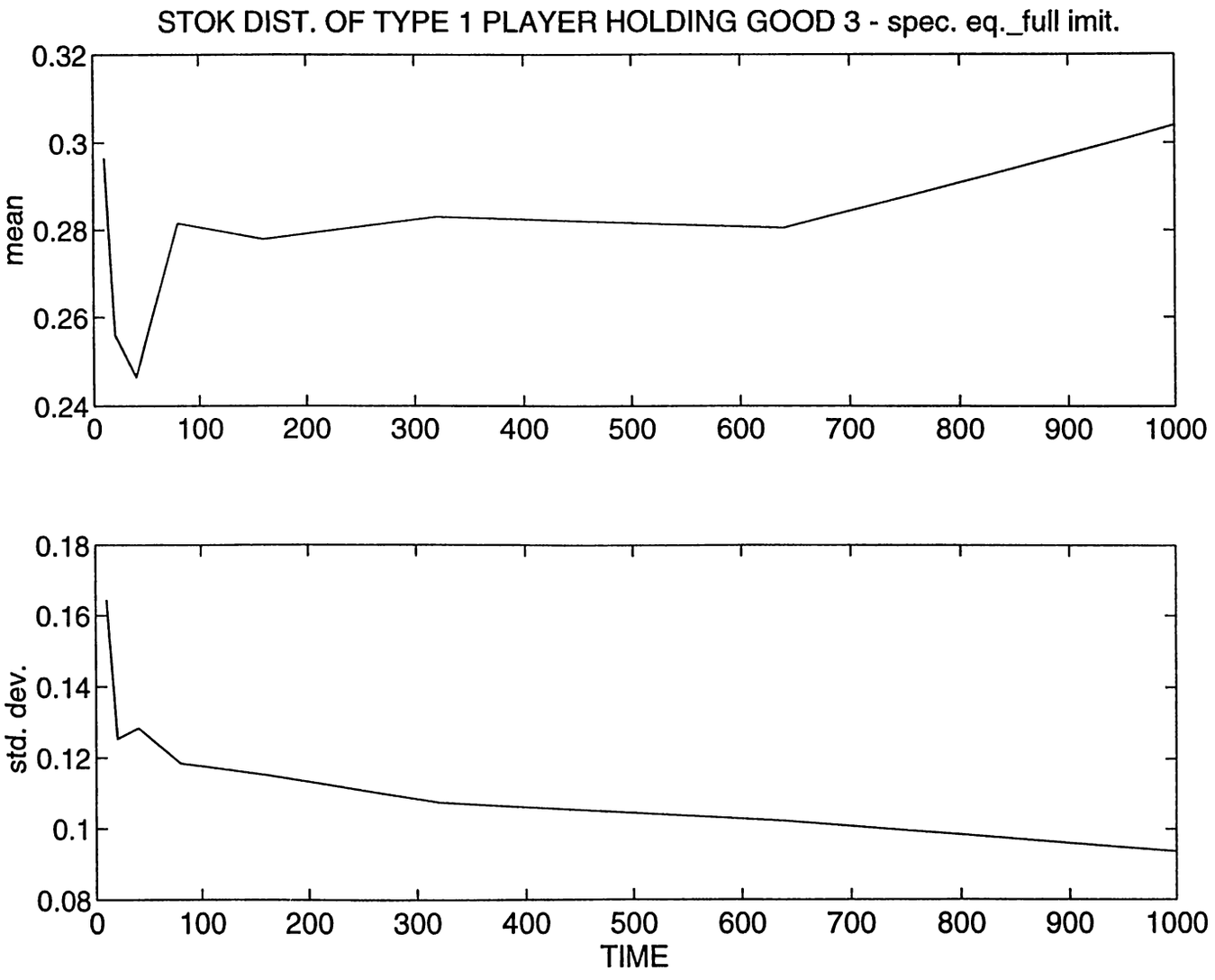

DENSITY OF TYPE 3 PLAYER FOR CONSUMPTION OF GOOD 3-spec. eq._half imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ACCEPTANCE RATIO OF THE DECISION

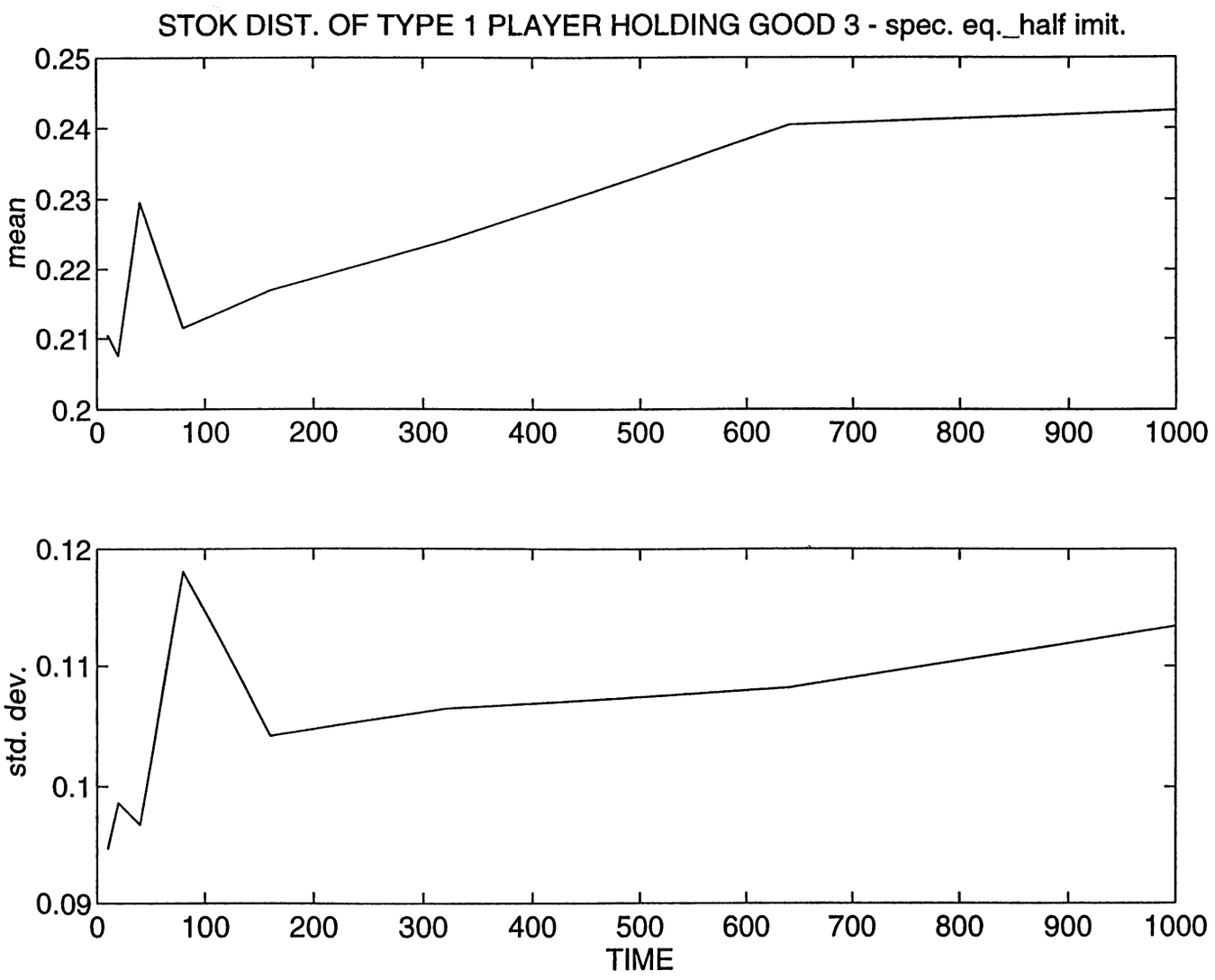

DENSITY OF TYPE 3 PLAYER FOR CONSUMPTION OF GOOD 3-spec. eq._no imit.
10th DAY
40th DAY
160th DAY
640th DAY
1000th DAY
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ACCEPTANCE RATIO OF THE DECISION

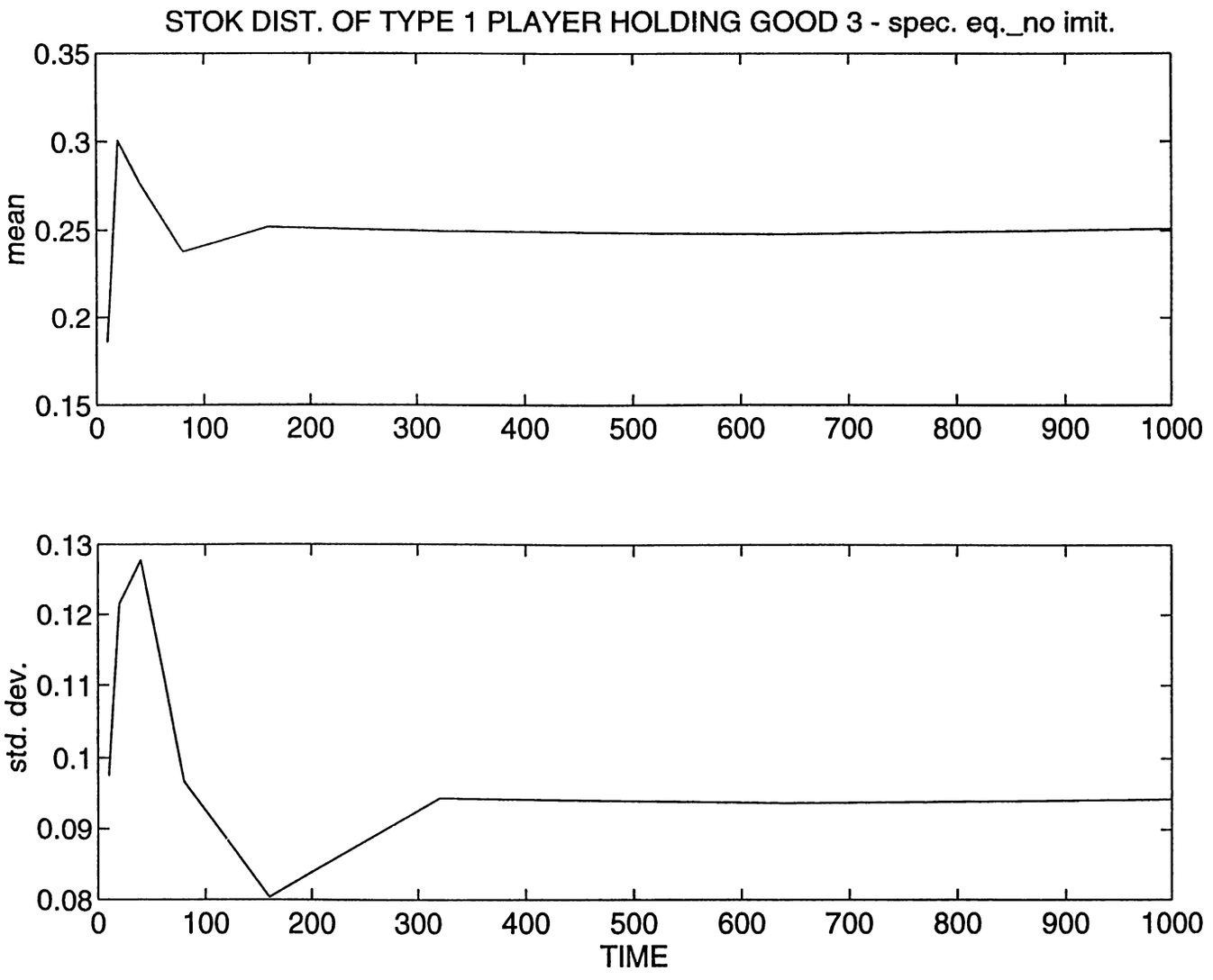

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 2 - spec. eq._full imit.
mean
0.76
0.74
0.72
0.7
0.68
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.18
0.16
0.14
0.12
0.1
0.08
0
100
200
300
400
500
600
700
800
900
1000
TIME

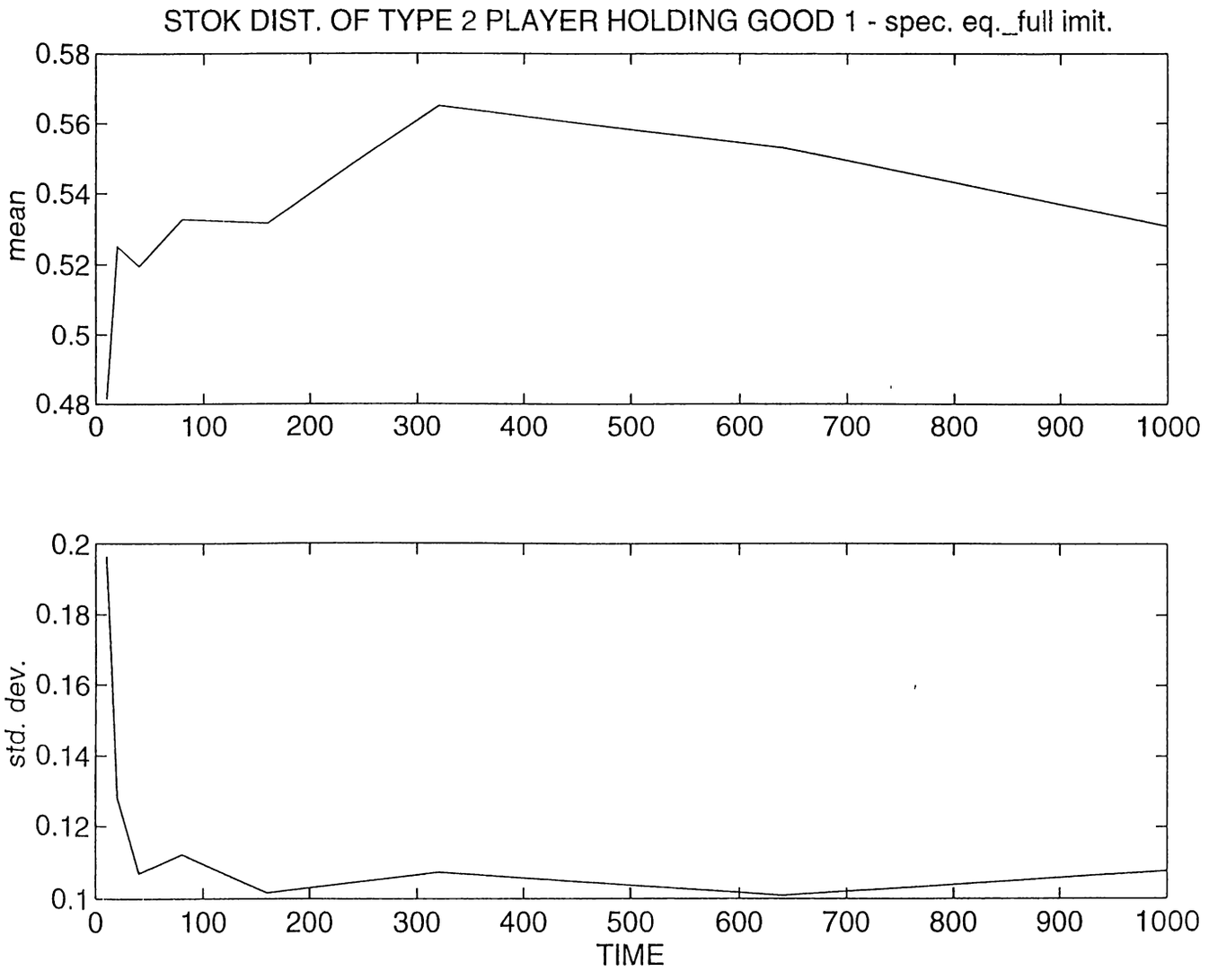

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 2 - spec. eq._half imit.
mean
0.77
0.76
0.75
0.74
0.73
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.12
0.115
0.11
0.105
0.1
0
100
200
300
400
500
600
700
800
900
1000
TIME

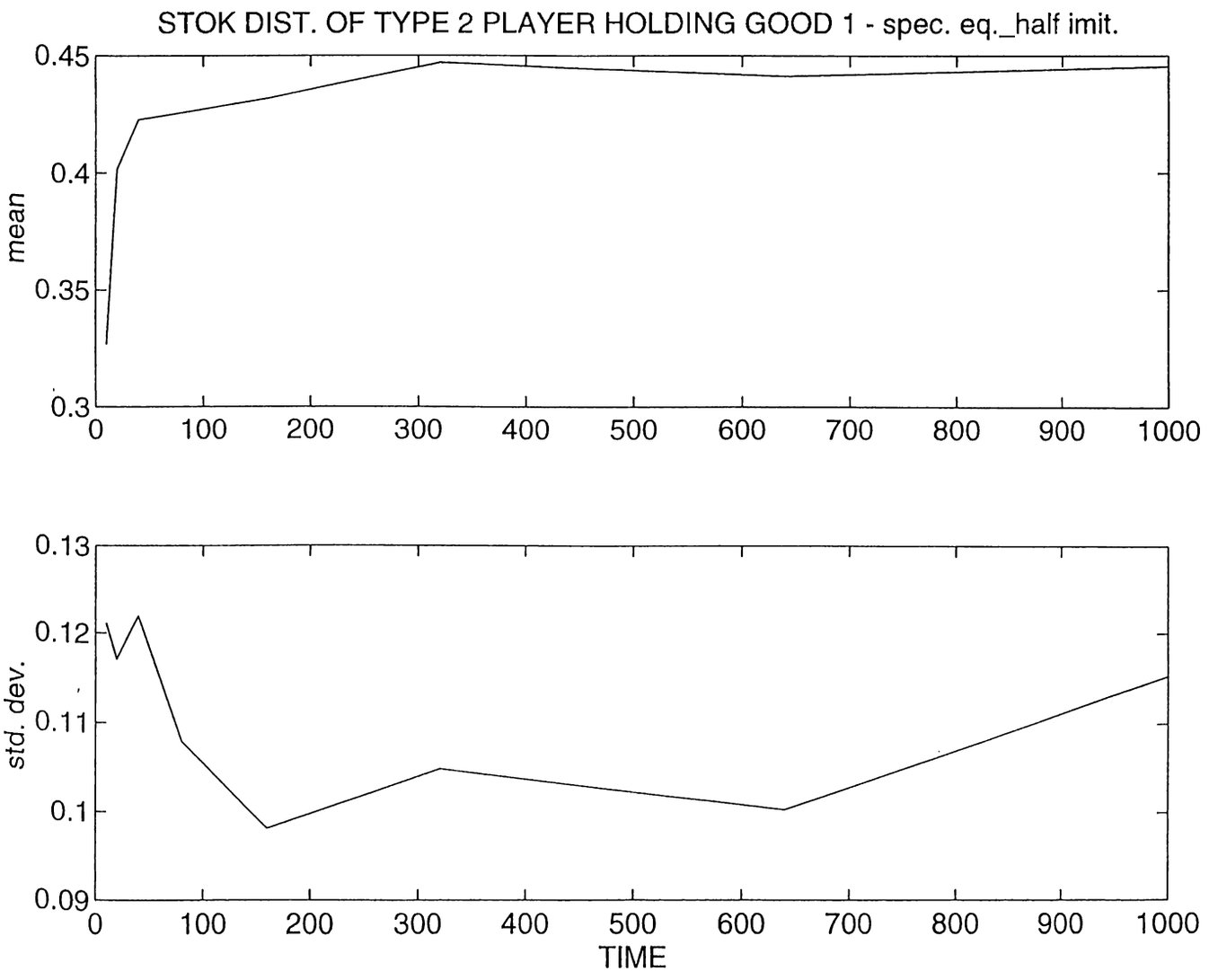

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 2 - spec. eq._no imit.
mean
0.75
0.7
0.65
0.6
0.55
0.5
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.16
0.14
0.12
0.1
0.08
TIME

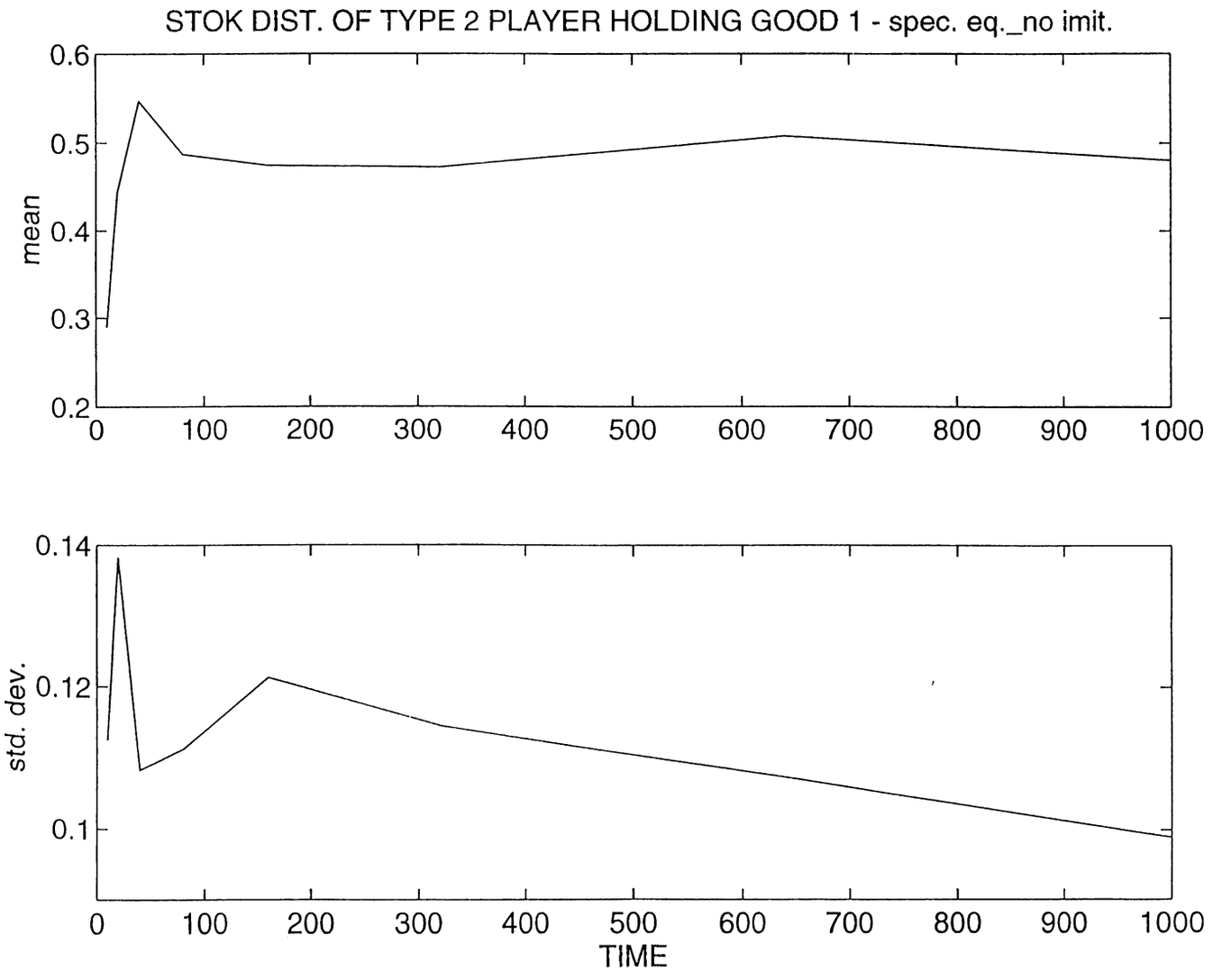

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 3 - spec. eq._full imit.
mean
0.32
0.3
0.28
0.26
0.24
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.18
0.16
0.14
0.12
0.1
0.08
TIME

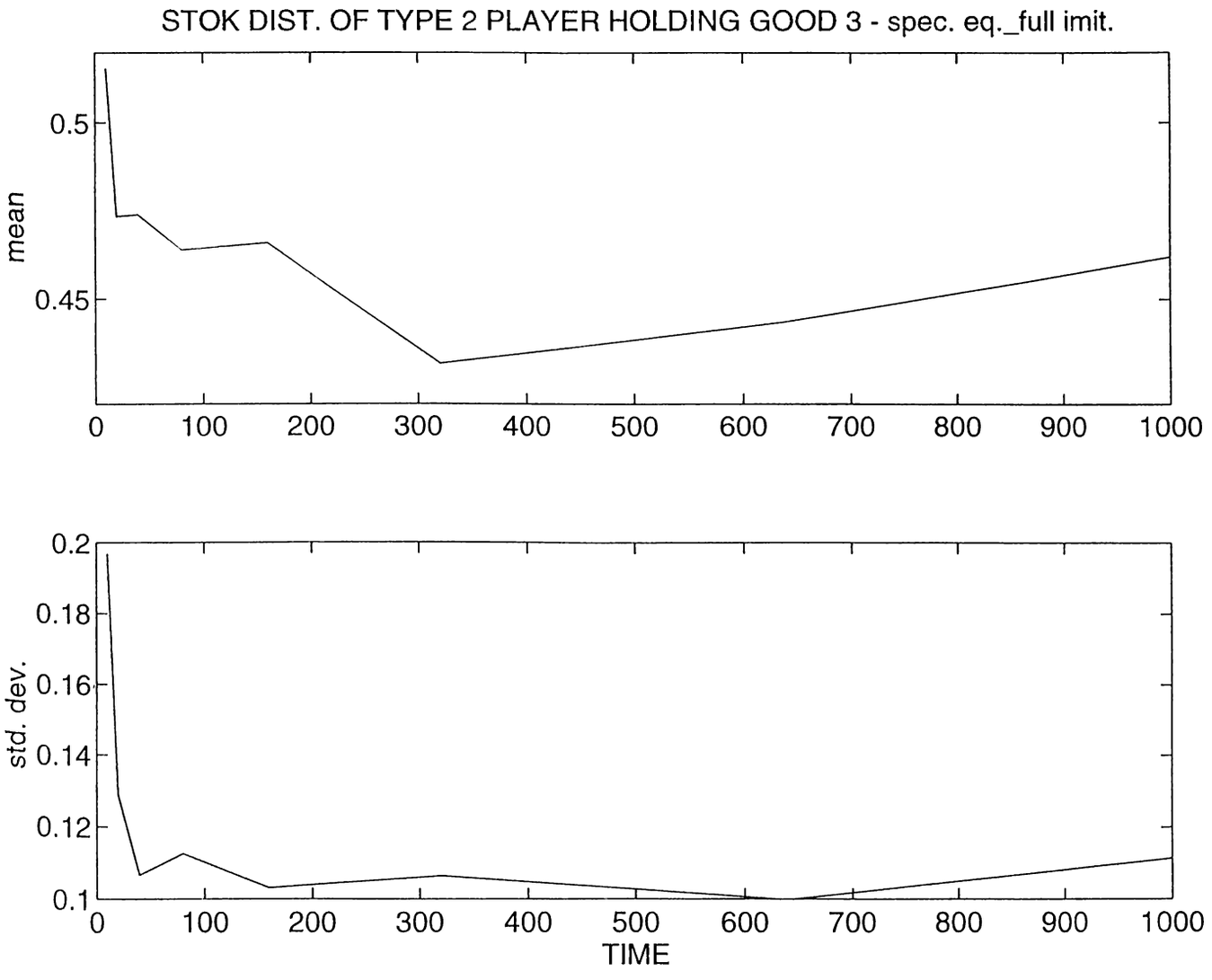

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 3 - spec. eq._half imit.
mean
0.25
0.24
0.23
0.22
0.21
0.2
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.12
0.11
0.1
0.09
0
100
200
300
400
500
600
700
800
900
1000
TIME

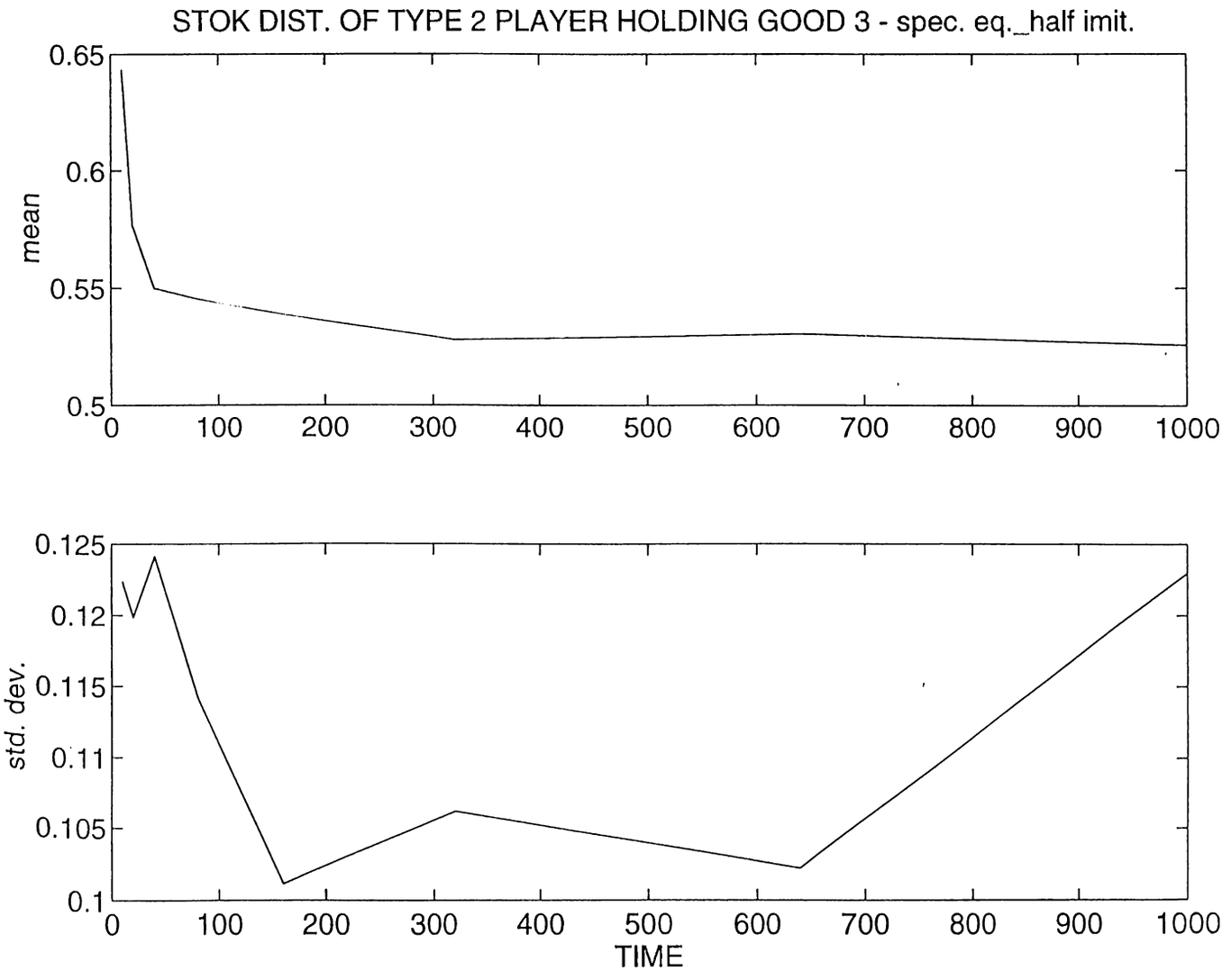

STOK DIST. OF TYPE 1 PLAYER HOLDING GOOD 3 - spec. eq._no imit.
mean
0.35
0.3
0.25
0.2
0.15
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.13
0.12
0.11
0.1
0.09
0.08
TIME

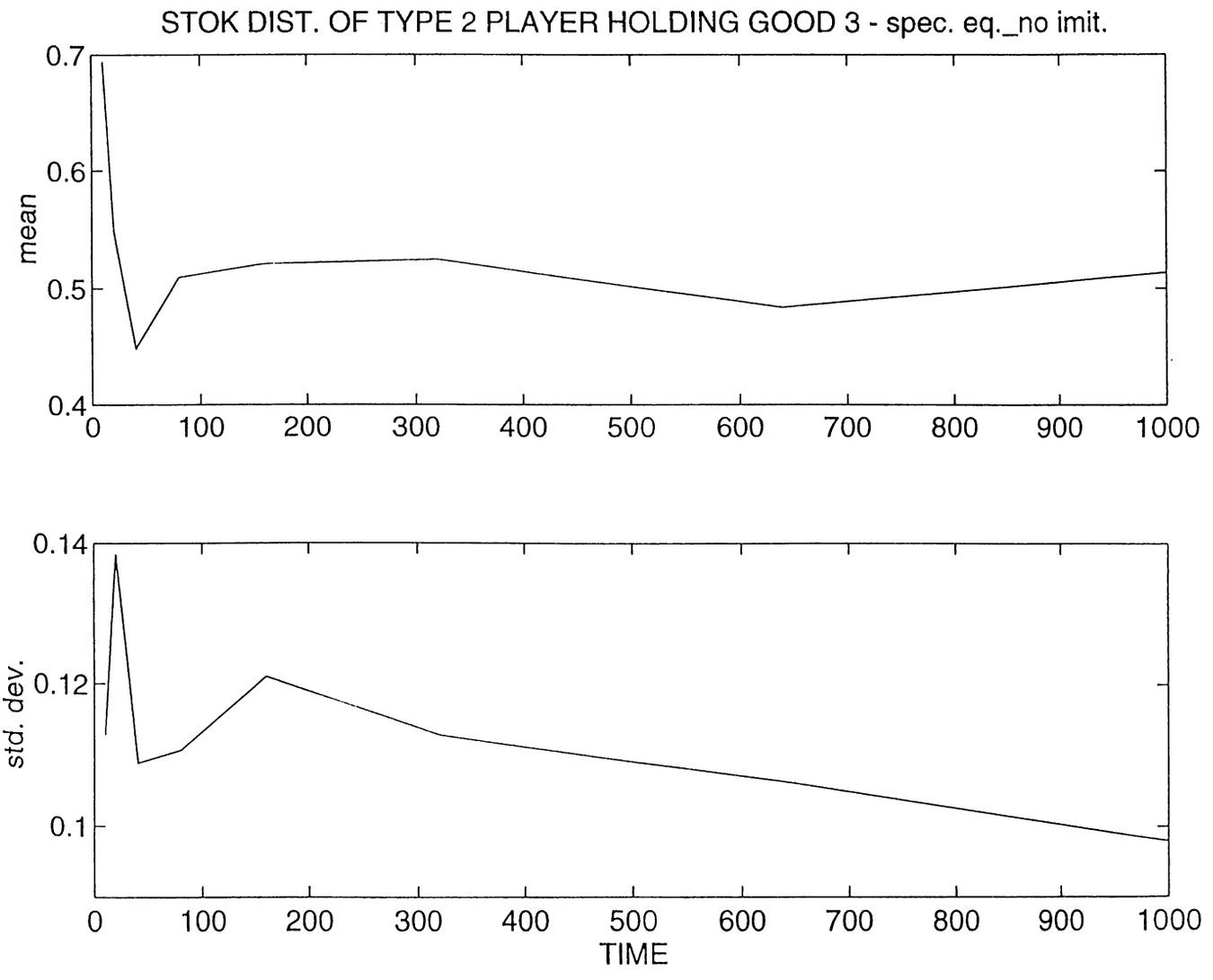

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 1 - spec. eq._full imit.
mean
0.58
0.56
0.54
0.52
0.5
0.48
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.2
0.18
0.16
0.14
0.12
0.1
0
100
200
300
400
500
600
700
800
900
1000
TIME

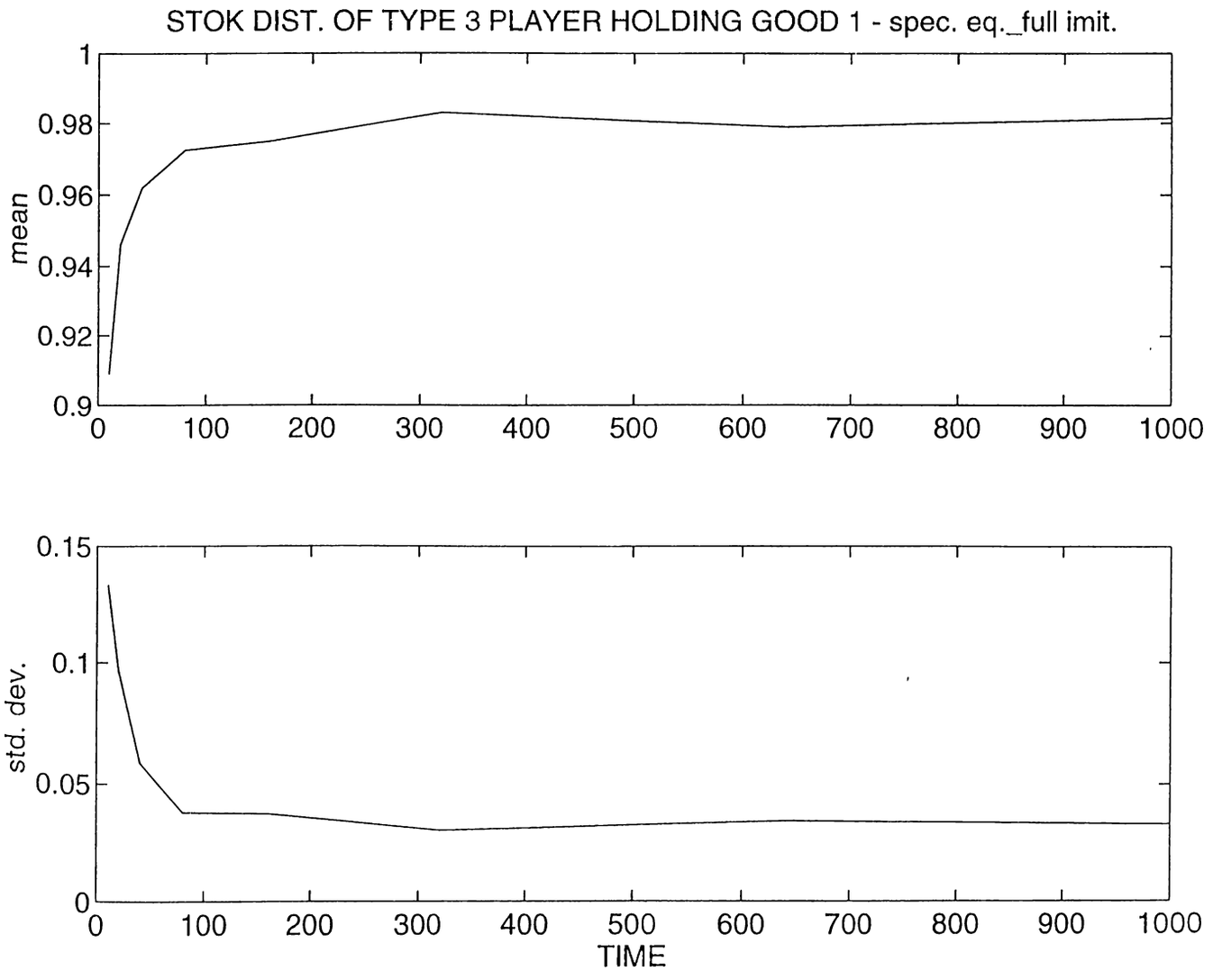

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 1 - spec. eq._half imit.
mean
0.45
0.4
0.35
0.3
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.13
0.12
0.11
0.1
0.09
0
100
200
300
400
500
600
700
800
900
1000
TIME

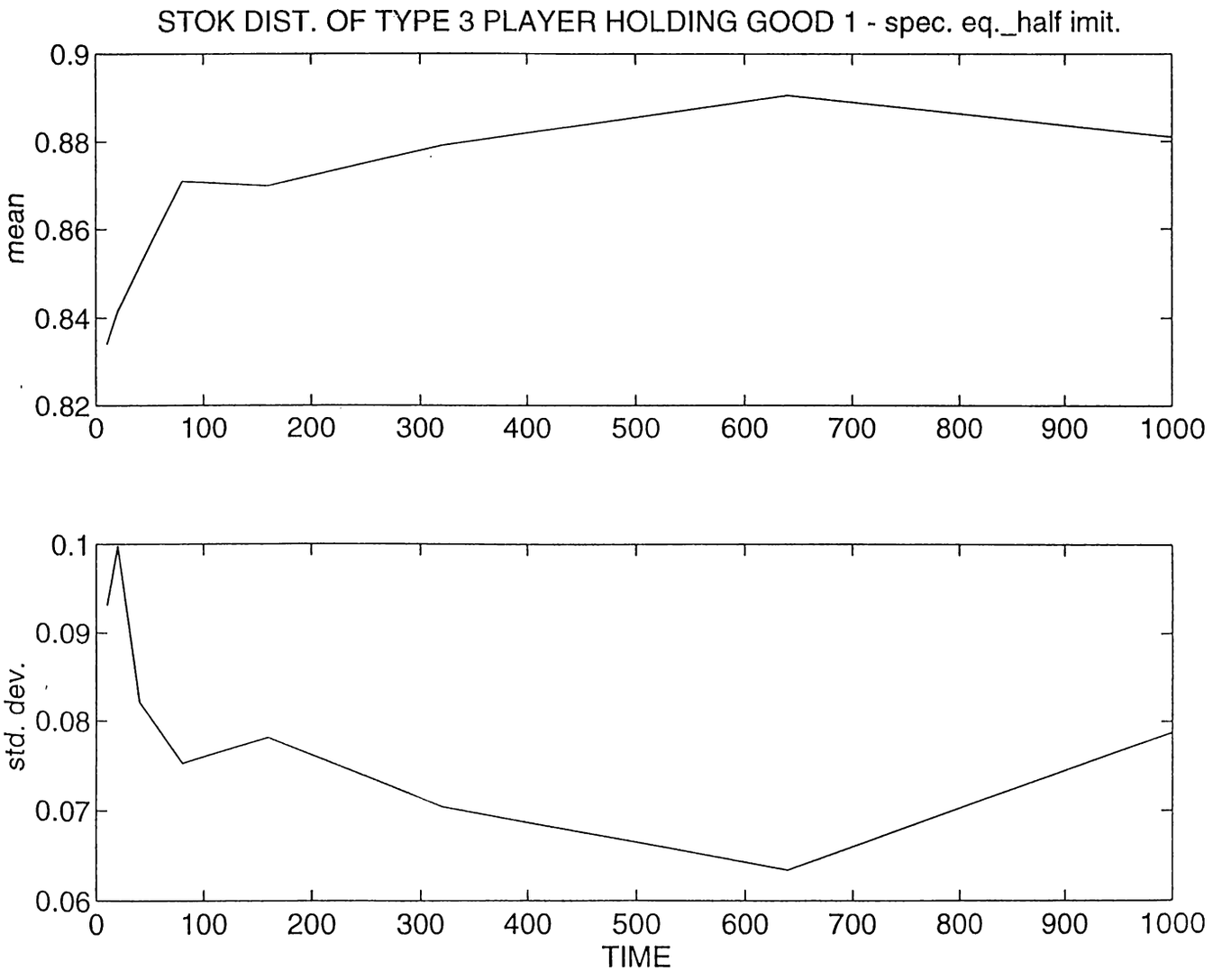

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 1 - spec. eq._no imit.
mean
0.6
0.5
0.4
0.3
0.2
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.14
0.12
0.1
0
100
200
300
400
500
600
700
800
900
1000
TIME

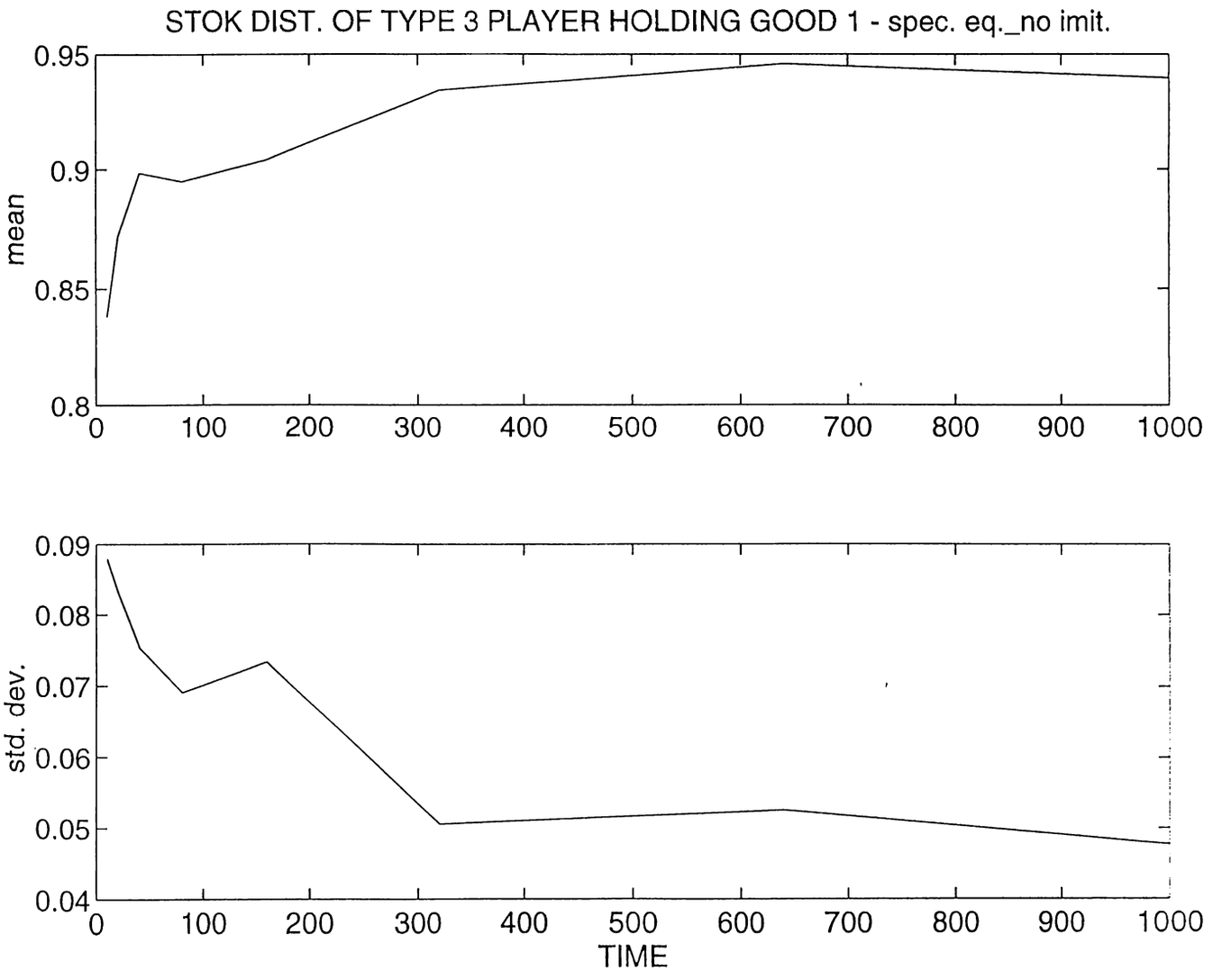

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 3 - spec. eq._full imit.
mean
0.5
0.45
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.2
0.18
0.16
0.14
0.12
0.1
TIME

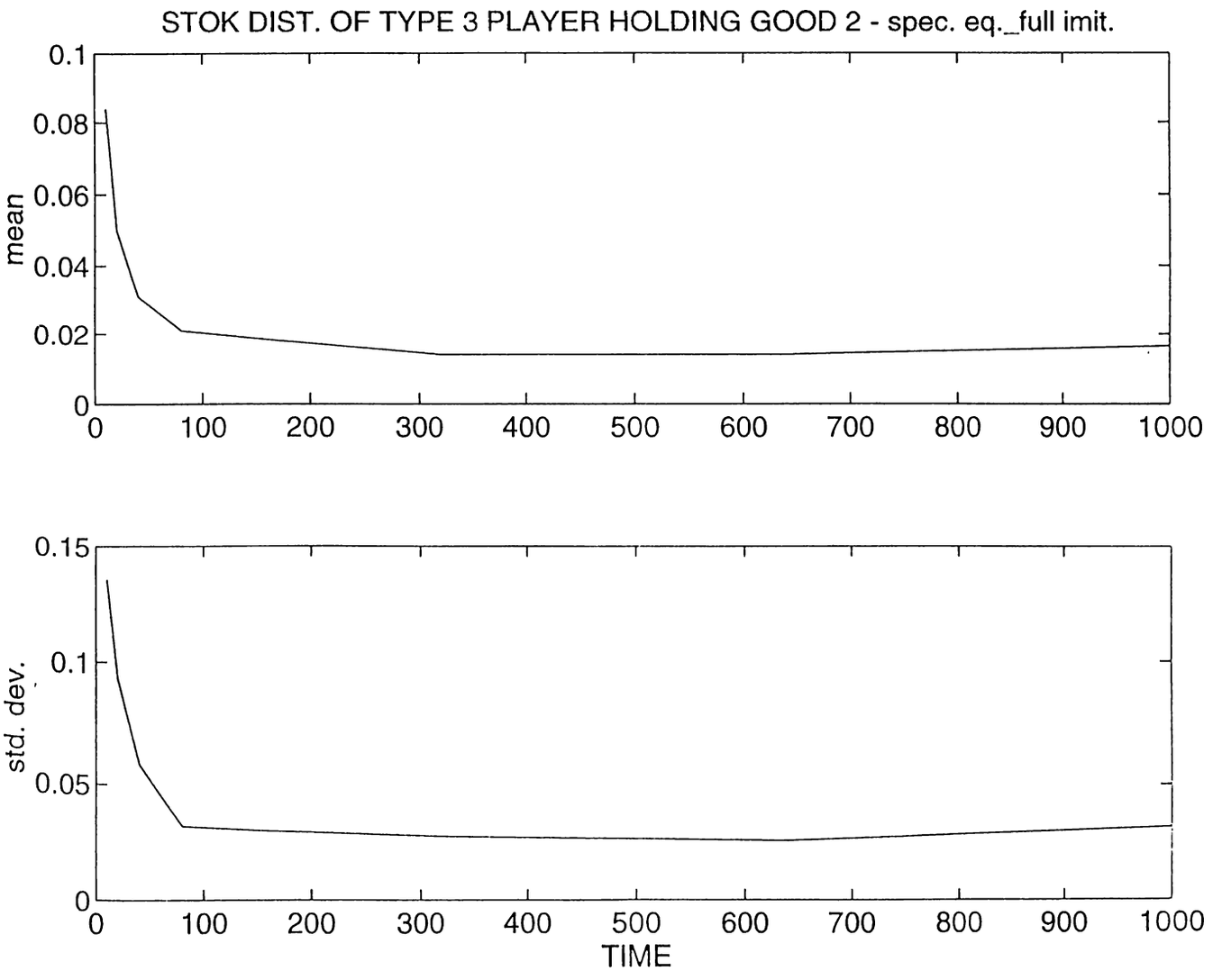

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 3 - spec. eq._half imit.
mean
0.65
0.6
0.55
0.5
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.125
0.12
0.115
0.11
0.105
0.1
TIME

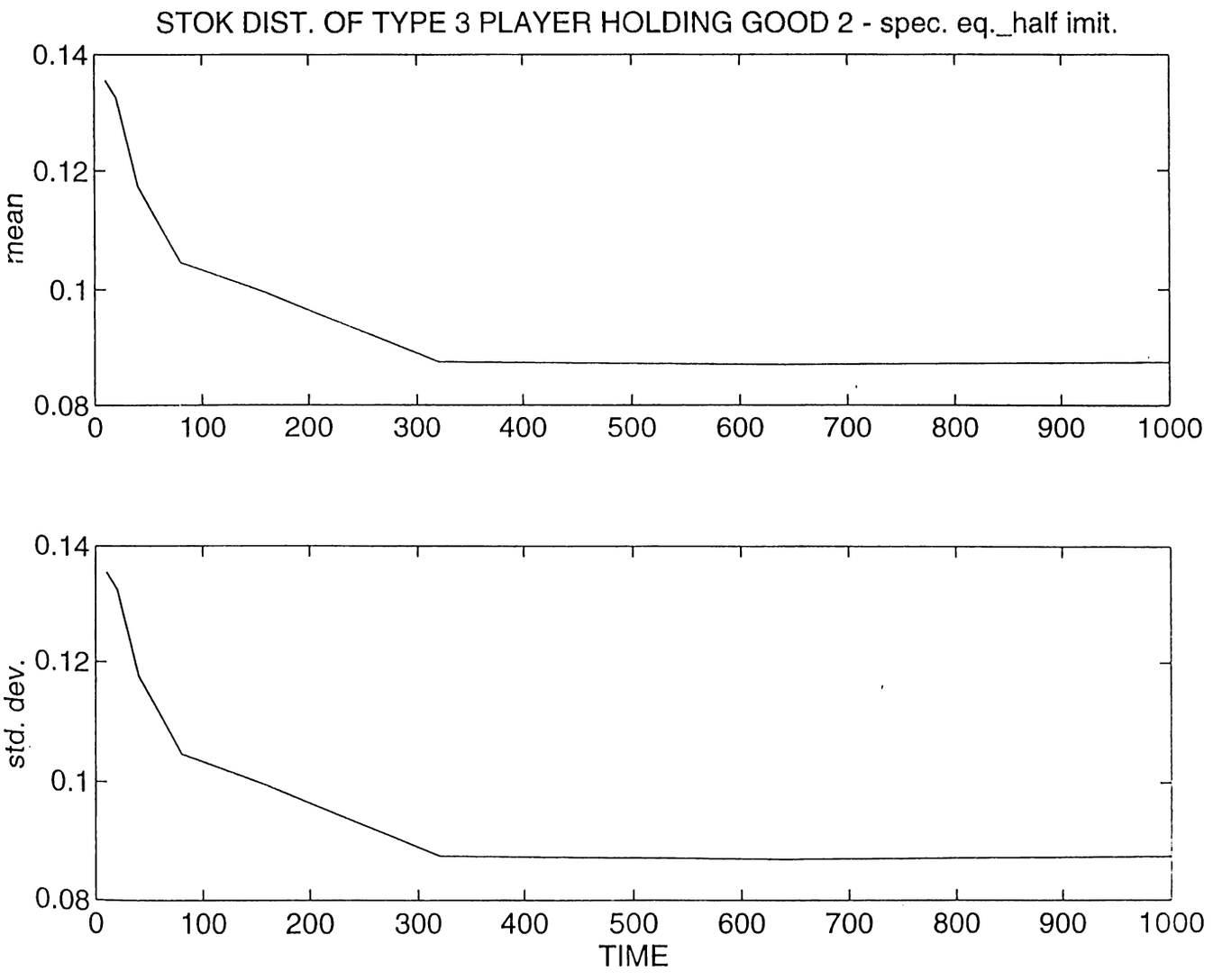

STOK DIST. OF TYPE 2 PLAYER HOLDING GOOD 3 - spec. eq._no imit.
mean
0.7
0.6
0.5
0.4
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.14
0.12
0.1
0
100
200
300
400
500
600
700
800
900
1000
TIME

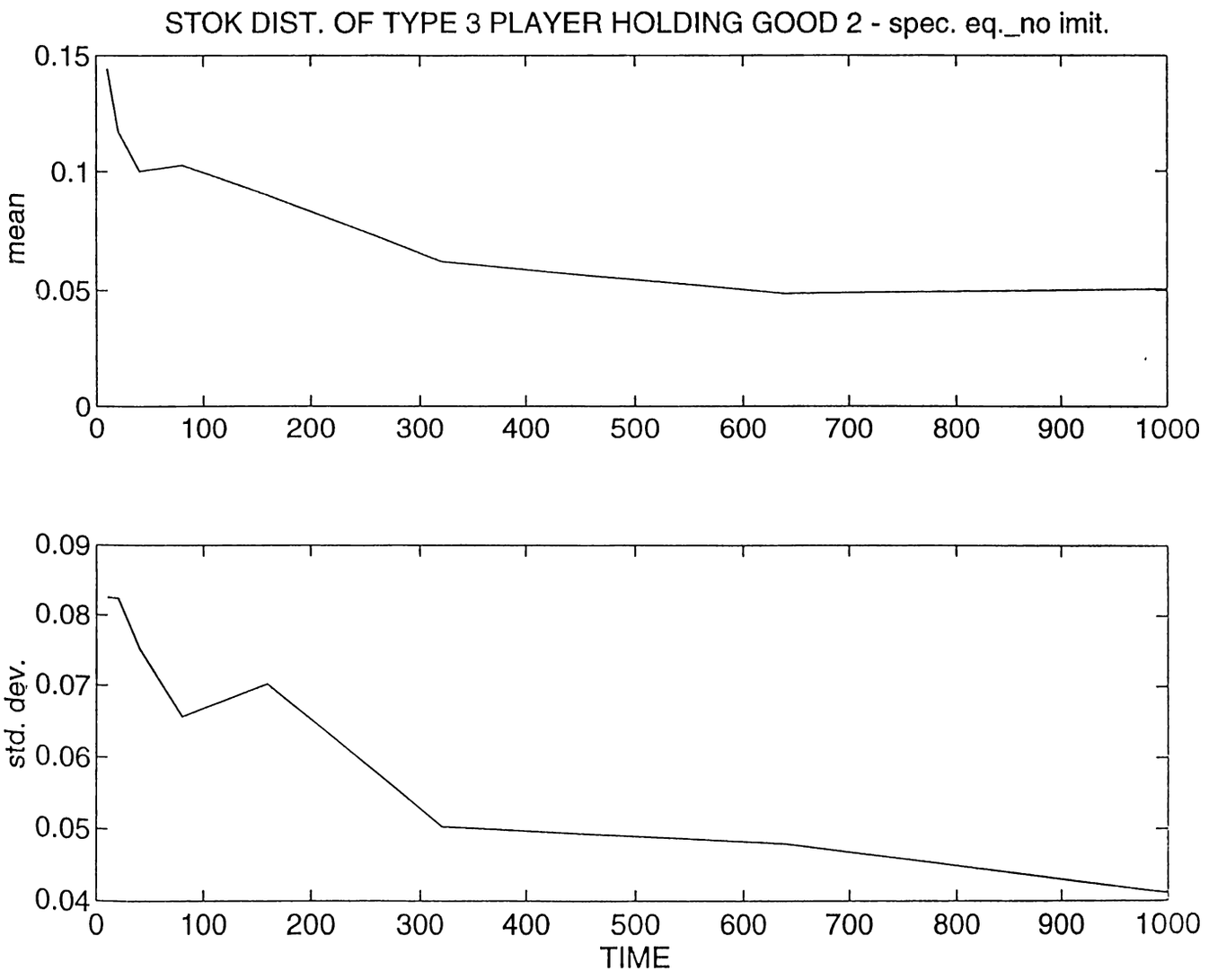

STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 1 - spec. eq._full imit.
mean
1
0.98
0.96
0.94
0.92
0.9
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
TIME


STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 1 - spec. eq._half imit.
mean
0.9
0.88
0.86
0.84
0.82
std. dev.
0.1
0.09
0.08
0.07
0.06
0
100
200
300
400
500
600
700
800
900
1000
TIME


STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 1 - spec. eq._no imit.
mean
0.95
0.9
0.85
0.8
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.09
0.08
0.07
0.06
0.05
0.04
0
100
200
300
400
500
600
700
800
900
1000
TIME


STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 2 - spec. eq._full imit.
mean
0.1
0.08
0.06
0.04
0.02
0
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
TIME


STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 2 - spec. eq._half imit.
mean
0.14
0.12
0.1
0.08
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.14
0.12
0.1
0.08
0
100
200
300
400
500
600
700
800
900
1000
TIME


STOK DIST. OF TYPE 3 PLAYER HOLDING GOOD 2 - spec. eq._no imit.
mean
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
900
1000
std. dev.
0.09
0.08
0.07
0.06
0.05
0.04
0
100
200
300
400
500
600
700
800
900
1000
TIME