Effects of Delay, Nonlinearity and Feedback on the Overall Complexity of a Stock Management Game [1]

Onur Özgün and Yaman Barlas

Boğaziçi University

Department of Industrial Engineering, 34342 Bebek, Istanbul, Turkey

Tel: +90 212 3597343, Fax: +90 212 2651800

E-mail: onur.ozgun@boun.edu.tr, ybarlas@boun.edu.tr

Abstract

The aim of this study is to test statistically the effects of delay, nonlinearity and feedback factors on the complexity of a stock management task. The task requires the player to bring the inventory to a target level and keep it there. Each of the individual complexity factors brings different challenges to the game. Using a slightly modified Latin square experimental design, we test the factors at different strength levels. We use two measures of game complexity: game scores and players subjective difficulty ratings. The results show that, with respect to the simple base game, only delay creates worsening in game performances. Also, with increased delay duration and delay order, subjects' performances deteriorate. However, nonlinearity and feedback do not deteriorate the game performance. Repeated trials of games involving all three factors allow performance improvement. However, learning can be transferred to the base game, only if the repeated trials are with nonlinear and feedback games. The subjective complexity ratings of the players yield parallel results, the overall correlation of game scores with the subjective difficulty ratings being +0.58*.*

Keywords: simulation games, stock management, systemic complexity, gaming experiments

1 Introduction

Simulation games are useful tools for training. However, as they replicate real-world dynamic problems, they involve nonlinear relations, time delays and coupled feedback loops. By understanding how these dynamic complexity factors affect the simulation games, we can derive better conclusions about real dynamic problems and design better gaming procedures that enhance learning from simulation games.

The effects of systemic complexity factors like delay, nonlinearity and feedback have been analyzed in the literature. Various studies have examined the relationship between *delay* and game performance and many report a negative effect of delay on performance (Broadbent and Aston, 1978; Diehl, 1989; Sterman, 1989b; Paich and Sterman, 1993; Diehl and Sterman, 1995; Brehmer, 1995; Atkins et al., 2002; Barlas and Özevin, 2004; Arango, 2006). Research also shows that *strength of feedback* in the simulator can be effective on the game performance (Diehl, 1989; Kampmann, 1992; Paich and Sterman, 1993; Diehl and Sterman,

[1]Research supported by Boğaziçi University Research Grant no 09HA301D

1995; Young et al., 1997; Langley et al., 1998). Likewise, *nonlinearity* has been shown to deteriorate the performance (Sterman, 1989a,b; Paich and Sterman, 1993).

While most papers analyze one factor at a time with a few exceptions (Paich and Sterman, 1993; Diehl and Sterman, 1995; Atkins et al., 2002), our research attempts to test the effects of multiple factors on the overall complexity of a simulation game. It is distinct from earlier line of work in the sense that three factors are tested in many (four or eight) levels. Moreover, we use a second performance measure based on players' subjective difficulty assessments in addition to a game score measure. In line of this goal, in an earlier paper, we presented en experimental study in which we tested the effects of three factors on the complexity of a growth management game (Özgün and Barlas, 2011). As a continuation of that study, in this paper we present a similar experiment for a stock management game.

One may intuitively expect each complexity factor to deteriorate game performance. However, although this may be true for raw measures of performance, performance relative to a benchmark may not deteriorate with increased complexity factors. Indeed, the results of the growth management game experiments (Özgün and Barlas, 2011) suggest that these factors do not necessarily worsen the game performance, which was measured in terms of cumulative profit normalized with respect to a benchmark behavior. There are also other studies reporting complexity factors being not effective on performance (Diehl, 1989; Atkins et al., 2002). Therefore, there are grounds for a systematic experimental analysis of the effects of complexity factors on the game performance in a stock management task. This way, we can also have a better understanding about generalizability of the results of the growth management game. In this study, we use a stock management task because similar games are widely used in dynamic decision-making experiments (Sterman, 1989b; Diehl, 1989; Bakken, 1993; Barlas and Özevin, 2004).

Section 2 presents the details of the task environment used in the experiments, the experimental design and the game protocol. Section 3 discusses how the benchmark behaviors are obtained. The findings of the experiments are presented in section 4. The paper concludes with discussion of the results and future work.

2 Method

2.1 Overview of Methodology

We focus on the effects of four dynamic complexity factors: *delay duration*, *delay order*, *nonlinearity* and *feedback strength*. Delay is analyzed in two dimensions (delay order and duration) because both may contribute to the complexity at different scales. *Delay duration* and *feedback strength* are analyzed in eight levels, while four levels of *delay order* and *nonlinearity* are tested.

Two measures of game complexity are used: game performance and players' subjective difficulty assessments. Game performance is measured by the total deviation from the constant target. Players' subjective difficulty assessments are recorded on a scale from 1 to 9. In order to make subjective measures as consistent as possible, all players play two initial reference games (one easy and one difficult game) with predefined difficulty ratings. Subjects are asked to assess the difficulty of each game just after playing it, with respect to these two reference games (see game instructions in Appendix A).

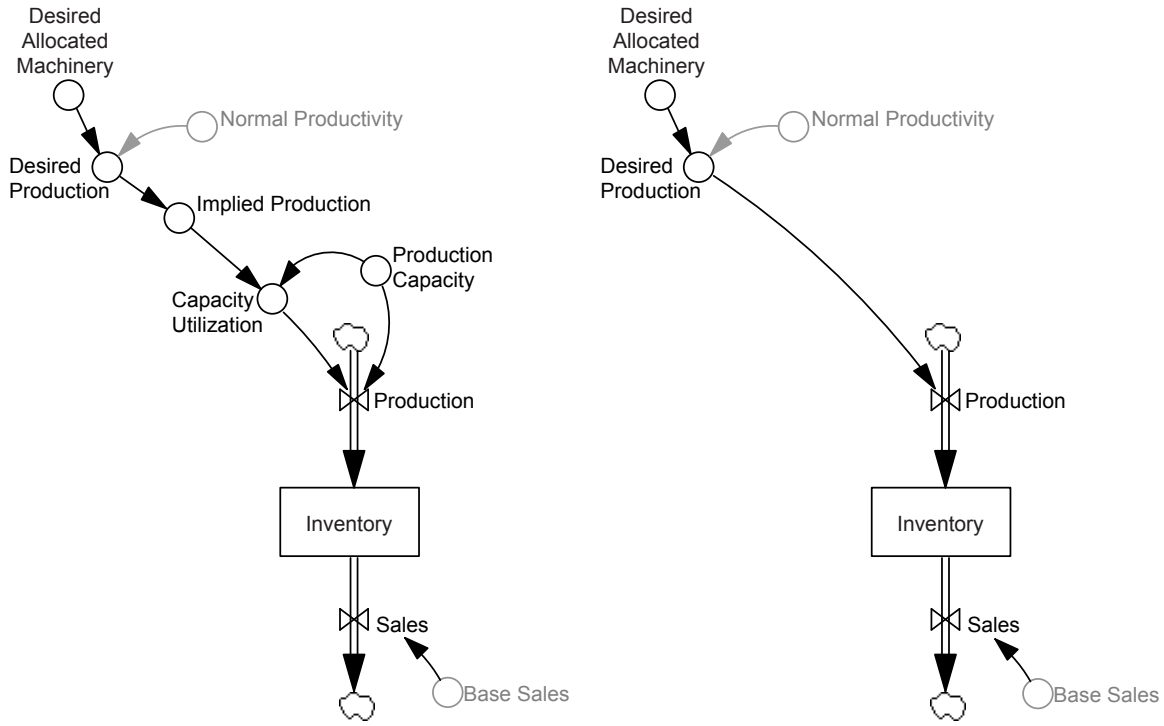
2.2 The Task

The stock management game takes place in a textile production company. The subjects play the role of a production manager who is responsible for t-shirts. Their objective is to bring *inventory* level to a target level as soon as possible and keep it there. They are allowed to change the number of machines desired to be allocated to production of t-shirts: *desired allocated machinery*. *Desired production* is calculated in the model by simply multiplying *desired allocated machinery* and *normal productivity*. In some game versions, actual *production* may be different than *desired production* due to decisions of some hypothetical production planners who may allocate a different number of machines to t-shirts than desired by the player or due to capacity constraints imposed by some other factors. The *inventory* stock grows by *production* and diminishes by *sales*. *Sales* is normally distributed around a constant mean (*base sales*) with a modest standard deviation, unknown to the player. The purpose of the noise is to make the easy versions of the game less trivial and and prevent fast learning from game to game. Initially, the *inventory* level is different from *target inventory*, and *production* is not equal to the mean *sales*. The time unit of the model is *days* and $dt = 1$. The time horizon is 40 days. The subjects know the general structure of the model but they do not know the parameter values (see Appendix A).

2.2.1 Base Game

The base structure of the stock management game is shown in Figure 1(a). Since the base game will constitute a reference point which the results of other games are compared against, we included some variables that do not have a function in the base game, but that become active in the other versions. In the base game, both *implied production* and *production* are equal to *desired production*. The production capacity is unlimited and *capacity utilization* is a linear function returning its input unchanged. Effectively the structure of the base game is as simple as shown in Figure 1(b). *Normal productivity* has a value of 1 lot/machine/day in all versions, and only serves as a unit converter. Its value is known by the players. The only unknown in the base game is random *sales* (which is normally distributed with mean 28 and standard deviation 3). The players' challenge is to bring *inventory* to the target and to discover the *sales* level by trial-and-error. Once *inventory* comes to the target inventory, it can stay there with minor movements due to the noise in the *sales*.

The game has a simple interface shown in Figure 2. The input devices are a slider for the *desired allocated machinery* decisions and an Advance button to simulate the game after giving a decision. Players are not allowed to change the initial conditions in the first period. The output devices are plots of *target inventory* (which is always constant at 200) and *inventory* (which is initially at 150 or 250), the numerical displays of *inventory*, *target inventory*, and *total deviation* of inventory from target. The vertical scale of the plot is identical in all game versions. However, some versions require a larger scale. The players can manually switch to a larger vertical scale if they need.



(a) Base game structure.

(b) Simplified diagram of the base game structure.

Figure 1: The base structure of the stock management game.

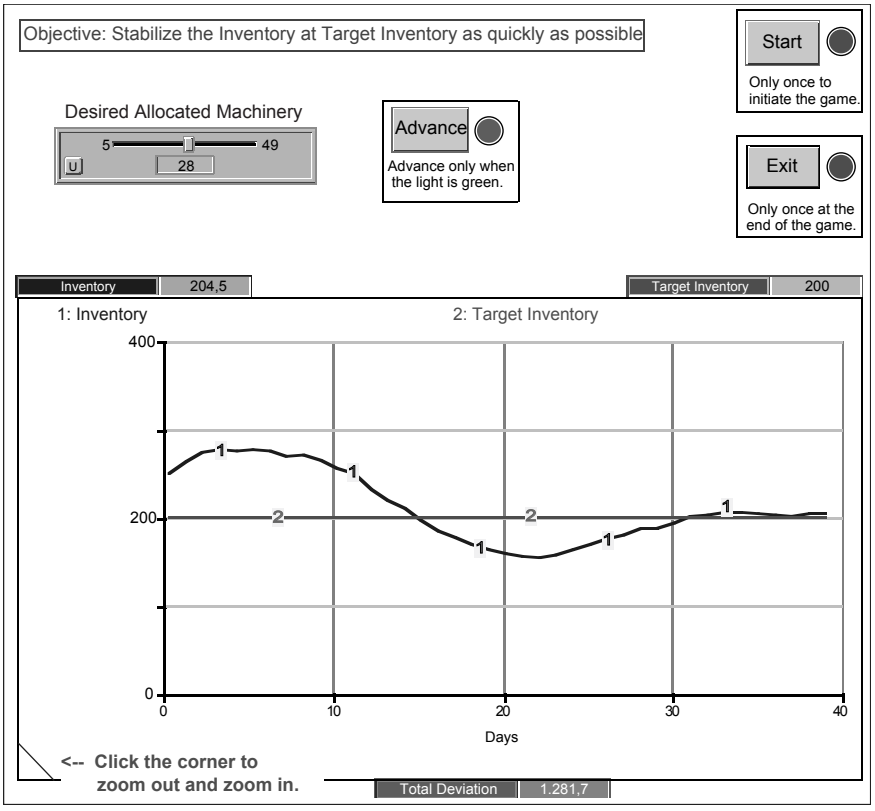
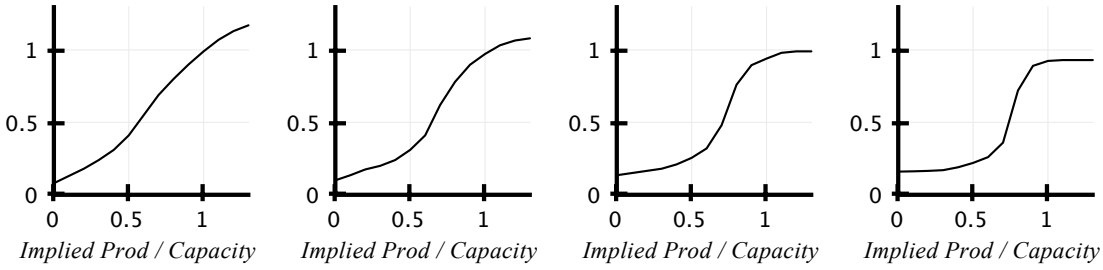


Figure 2: The user interface of the stock management game.

2.2.2 Nonlinearity Factor

The nonlinearity factor is added by making *capacity utilization* a nonlinear function of *implied production* and *production capacity*. Figure 3 shows the forms of nonlinear functions used for utilization function. Four levels of *nonlinearity* are tested: mild (denoted by N1), moderate (N2), high (N3) and extreme (N4). Nonlinearity brings two different challenges. First, there is a limit on maximum and minimum possible production, so players' *desired productions* are not always realized. Second, as *nonlinearity* increases, the shape of *capacity utilization* function becomes more uneven. As experienced by the player, the game behaves as if it is unresponsive to changes in player's decisions in a certain region, while it shows abrupt changes at a certain point.



(a) Mild (N1) (b) Moderate (N2) (c) High (N3) (d) Extreme (N4)

Figure 3: Nonlinear *capacity utilization* functions.

2.2.3 Delay Factor

In the game version involving *delay*, *implied production* at day t is equal to the delayed version of *desired production*. This can be regarded as a delay due to procedural and technical arrangements in production planning. *Delay* is analyzed in two components: *order of delay* and *delay duration*. *Order of delay* has four levels while *delay duration* has eight levels. There are $4 \times 8 = 32$ possible combinations of all levels of these two variables. The game versions involving *delay* are given in Table 1. When there is delay, in the beginning of the game, *inventory* unavoidably moves away from the target for a number of periods, after which it responds to the actions of the player. As delay gets longer, this initial phase becomes longer. Therefore, to make a fair comparison between games involving different delay durations, we need to consider this initial unavoidable deviation from the target. In addition, delay brings an important difficulty in terms of player experience. Since the game does not respond to players' actions immediately, players cannot easily understand the consequences of their actions. Combined with the effect of random noise,

Table 1: Game versions involving *delay*.

	Delay Duration							
Delay Order	2 days	4 days	6 days	7 days	8 days	9 days	10 days	11 days
First Order	O1T2	O1T4	O1T6	O1T7	O1T8	O1T9	O1T10	O1T11
Third Order		O3T4	O3T6	O3T7	O3T8	O3T9	O3T10	O3T11
Fifth Order			O5T6	O5T7	O5T8	O5T9	O5T10	O5T11
Discrete	ODT2	ODT4	ODT6	ODT7	ODT8	ODT9	ODT10	ODT11

Oi: i^{th} order exponential delay, OD: discrete delay, Tn: delay duration = n days.

it becomes difficult for the players to assess the strength of the delay (Players know that a game involves delay, but they do not know its duration or order).

2.2.4 Feedback Factor

A reinforcing feedback loop is created between *sales* and *production* as shown in Figure 4. As *sales* increases, *implied production* increases even if *desired allocated machinery* is not changed. This can be thought of a managerial policy or company rule that automatically adjusts production to keep up with the changed demand (Figure 5(b)). The rise of *implied production* increases *production*, and as a result the *inventory* stock. The feedback loop is completed through the causal link between *inventory* and *sales*. In other game versions *sales* was external, in this version, it is affected from *inventory*. This is an equally likely scenario, meaning that in this model the firm adjusts its sales effort depending on its inventory level. As *inventory* accumulates, *sales* starts to accelerate, but in a decreasing rate. The shape of the function representing *effect of inventory on sales* is seen in Figure 5(a). When *inventory* is equal to the target *inventory*, the effect function becomes 1. At this point, *sales* formulation becomes equivalent to the formulation in the other versions. From the player's perspective, it becomes more challenging to bring the *inventory* to the target as it moves away from the target. Higher the distance between the *inventory* and the target, stronger the effect that pushes *inventory* away. When *inventory* reaches the neighborhood of the *target inventory*, feedback becomes ineffective and the game resembles the base game.

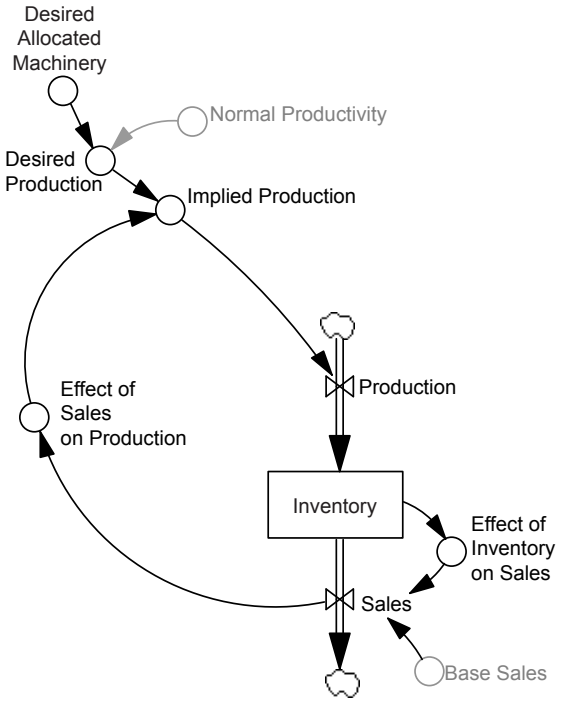
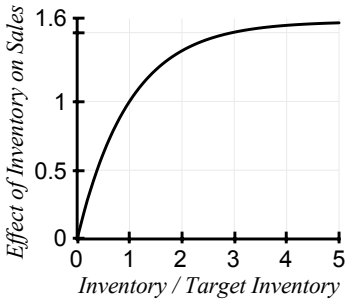
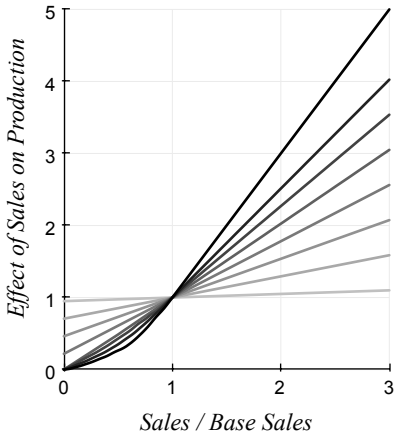


Figure 4: The structure of the stock management game with feedback.

Eight forms of the effect function for eight feedback strength levels are shown in Figure 5(b). These eight forms of functions correspond to the eight levels of feedback strength factor and they are coded as F1–F8. The shape of the effect function becomes nonlinear for small values in the curves with high slopes. This is to avoid the risk of getting caught in a position where it becomes impossible to recover from a decreasing inventory trend.



(a) Function defining *effect of inventory on sales*.



(b) Function defining *effect of sales on production*. Darker the color stronger the feedback.

Figure 5: Effect functions used in the feedback version.

2.3 Experimental Design

There are eight levels of *delay duration* and *feedback strength* and four levels of *delay order* and *nonlinearity*. We use a modified version of Latin square design as shown in Table 2. A total of 24 subjects is used. All players play the simplest base game in the beginning and at the end. They play the difficult game as the second game. The aim of the difficult game is to help subjects in their difficulty assessments by providing another reference point with a predefined difficulty rating (the other extreme reference is the base game). The difficult game has a fifth order nine-day *delay*, strong *nonlinearity*, and a *feedback strength* level of 7. This configuration is determined after pilot experiments. *Delay order* and *delay duration* are treated jointly (players 1–8). On top of the Latin square design for *delay duration*, the levels of *delay order* are embedded such that: (1) each player plays each *delay order* twice, and (2) each *delay order* is combined with each *delay duration*, twice. *Nonlinearity* design (players 9–16) is composed of four repeated Latin square designs. Trials 3–6 of Subjects 9–12 is repeated in the following trials and sequence of Subjects 9–12 is repeated by Subjects 13–14. The experimental design of *feedback* (subjects 17–24) is a pure Latin square design.

Our pilot studies and several other studies have shown that as the subjects repeat a game, the familiarity increases and scores improve. We want to slow down the procedural learning in order to make sure that even in the last trials players still face a challenge and do not just copy their actions in earlier trials. To this end, we make minor modifications to the game interface between game versions. We slightly change the limits of the slider and alter the initial position of *inventory*. At the same time, we adjust the base sales (in the range of 22–39) and initial *desired allocated machinery* so that we do not cause an artificial difficulty by these modifications. Since all players play all the versions of these factors (yet in different orders), they all face the same interface conditions, in different orders.

Table 2: The experimental design

	Delay Group							
Trial	Sbj1	Sbj2	Sbj 3	Sbj4	Sbj5	Sbj6	Sbj7	Sbj8
1	base	base	base	base	base	base	base	base
2	diff.	diff.	diff.	diff.	diff.	diff.	diff.	diff.
3	ODT6	O5T7	O5T6	ODT8	O5T10	O5T11	O1T4	O5T9
4	ODT9	ODT4	O1T10	O5T6	O1T8	O3T4	ODT11	O1T7
5	O1T2	O3T8	O1T4	O5T9	O3T11	ODT10	ODT7	O5T6
6	O3T7	O3T4	ODT11	O3T10	ODT9	O3T8	O3T6	O3T4
7	O1T8	O1T6	O3T9	O1T11	O3T7	ODT4	O5T6	O3T10
8	O5T6	O1T9	O5T8	O1T7	O1T2	O1T6	O1T10	O1T11
9	O3T11	ODT10	O3T6	ODT2	O5T6	O5T7	O3T9	ODT8
10	O5T10	O5T11	ODT7	O3T4	ODT6	O1T9	O5T8	ODT2
11	base	base	base	base	base	base	base	base
	Nonlinearity Group							
Trial	Sbj9	Sbj10	Sbj11	Sbj12	Sbj13	Sbj14	Sbj15	Sbj16
1	base	base	base	base	base	base	base	base
2	diff.	diff.	diff.	diff.	diff.	diff.	diff.	diff.
3	N2	N1	N3	N4	N2	N1	N3	N4
4	N1	N4	N2	N3	N1	N4	N2	N3
5	N4	N3	N1	N2	N4	N3	N1	N2
6	N3	N2	N4	N1	N3	N2	N4	N1
7	N2	N1	N3	N4	N2	N1	N3	N4
8	N4	N3	N1	N2	N4	N3	N1	N2
9	N3	N2	N4	N1	N3	N2	N4	N1
10	N1	N4	N2	N3	N1	N4	N2	N3
11	base	base	base	base	base	base	base	base
	Feedback Group							
Trial	Sbj17	Sbj18	Sbj19	Sbj20	Sbj21	Sbj22	Sbj23	Sbj24
1	base	base	base	base	base	base	base	base
2	diff.	diff.	diff.	diff.	diff.	diff.	diff.	diff.
3	F3	F4	F1	F5	F7	F8	F2	F6
4	F6	F2	F7	F3	F5	F1	F8	F4
5	F1	F5	F2	F6	F8	F7	F4	F3
6	F4	F1	F8	F7	F6	F5	F3	F2
7	F5	F3	F6	F8	F4	F2	F1	F7
8	F2	F6	F5	F4	F1	F3	F7	F8
9	F8	F7	F3	F1	F2	F4	F6	F5
10	F7	F8	F4	F2	F3	F6	F5	F1
11	base	base	base	base	base	base	base	base

2.4 Procedure

The subjects are recruited from undergraduate and graduate engineering students. The experiments are carried out using STELLA software. Subjects are given a written instruction (Appendix A). The instructions give an overview about the underlying model structure, the game objective and instructions about the subjective difficulty assessment. During the experiments, they are asked to rate the difficulty of each game they play, on a scale 1 to 9, where 1 corresponds to an extremely easy and 9 corresponds to an extremely hard game (See instructions in Appendix A for the complete scale).

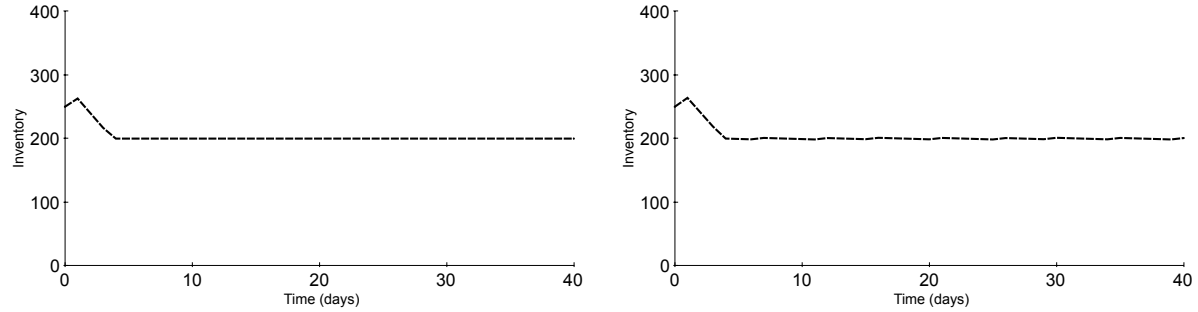
Subjects are told that the first (base) game has pre-assigned difficulty of 1 and the second (difficult) game has a difficulty of 7. After the instructions, subjects play a trial game for getting familiar with the game interface, the software and the procedure. The trial game has different parameters than other games and is designed not to reveal specific subtleties of the actual games. After the trial game, subjects play the base game followed by the difficult game, after which they play eight games involving one of the complexity factors (Table 2). At the end of each game, they record their score and rate the difficulty of the game they played. At the very end, they play the base game once more. After completing all games, they have a chance to revise their difficulty ratings and write down the factors and

critical information they used in making their decisions. The players are given monetary rewards according to their performances. The reward function is based on *total deviation* of inventory from the target (see Appendix A for more specific information).

3 Benchmark Behaviors

Benchmark behavior is defined as the best possible decisions yielding the minimum total inventory deviation from the target level. While determining the benchmark behaviors, we remove the noise in *sales*.

Figure 6(a) shows the benchmark behavior for the base game. Remember that the players are not allowed to change the initial values in their first decisions. Because of this rule, *inventory* increases in the first day. Starting from the second day, we apply decisions that would bring the *inventory* to the target as fast as possible. Since we know the *sales* level exactly, this is a trivial task. Similarly, for the nonlinear and feedback versions, we can easily find the behavior yielding minimum deviation from the target. Figure 6(b) shows the resulting benchmark behavior for a *feedback* game. The benchmark behavior of the nonlinear game is very similar.

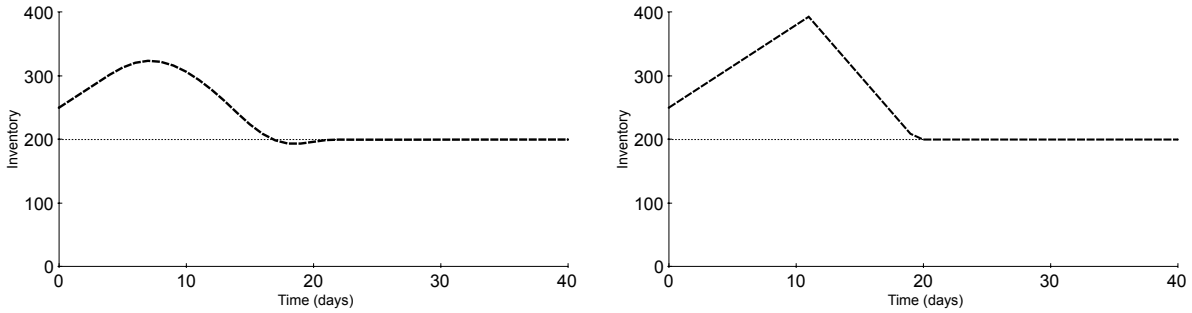


(a) Base game

(b) *Feedback* game (setting F8)

Figure 6: Benchmark behaviors for the base and *feedback* game versions.

Finding the benchmark for the games involving *delay* is not that trivial. If you are aggressive to bring *inventory* to the target very quickly, you may end up overshooting the target due to the effects of earlier decisions. Conversely, if you play conservatively not to overshoot the target, you may end up with a large deviation from the target. Fortunately, it is possible to optimize the behavior for minimum deviation from the target. We formulate the problem of finding the decision sequence that minimizes the total deviation as a mixed-integer nonlinear optimization problem. Using BARON 9.0.6 –a global optimization solver for mixed-integer nonlinear optimization problems (Tawarmalani and Sahinidis, 2005)– we find the minimum possible deviation with less than 1% optimality gap. Figure 7 shows benchmark behaviors for two different game versions involving delay, a continuous and a discrete delay. Note that even the best possible behaviors exhibit considerable deviations from the target, unlike other game versions. The amount of this unavoidable deviation is deduced from the players' cumulative deviation in statistical analysis.



(a) Setting O3T8

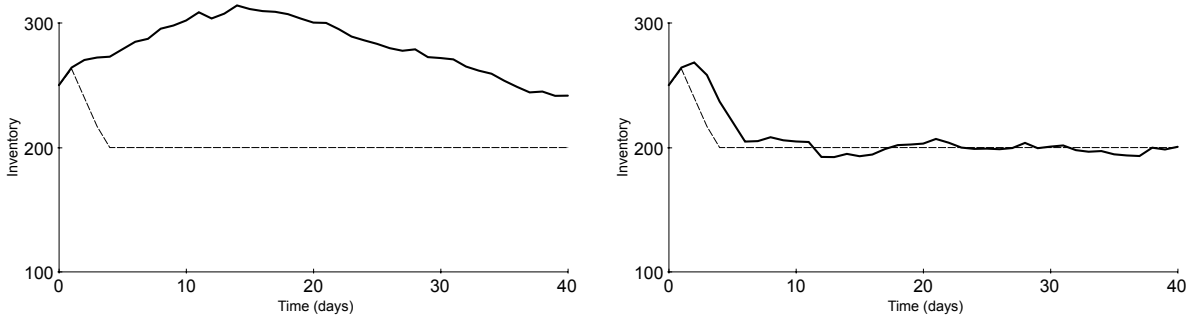
(b) Setting ODT10

Figure 7: Benchmark behaviors for the game versions involving *delay*.

4 Results

4.1 Qualitative Analysis

First, we present example behaviors to show some features of the experiments. Appendix B presents the behaviors for all trials of all players. Player characteristics have an important influence on the variance of game results. For instance, consider two subjects' performances in the first base games shown in Figure 8. Although they are given the same instructions and they play the same trial game before this game, there is a remarkable performance difference between two players.

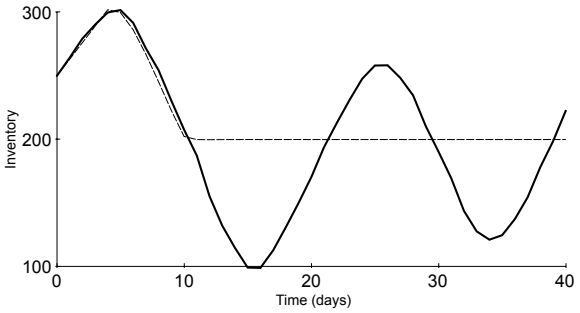


(a) Player 1: base

(b) Player 5: base

Figure 8: Two very different behaviors in the first base game (solid line) compared with the benchmark (dashed line).

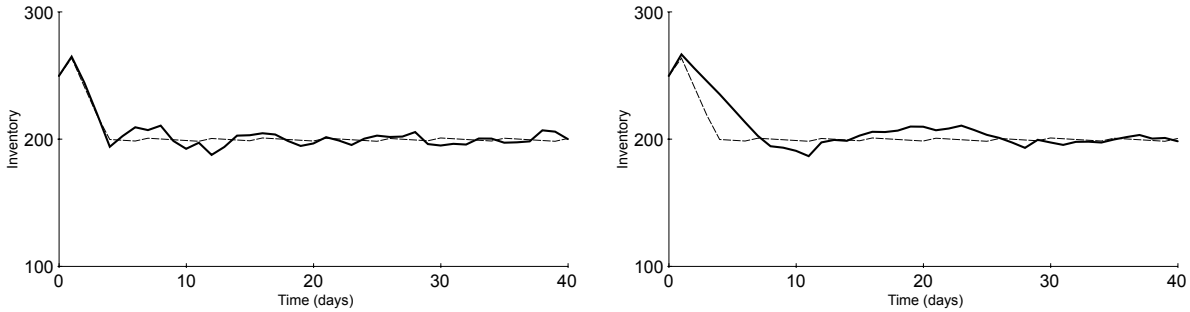
Figure 9 shows a typical behavior from a game involving *delay*. As the inventory starts to increase, the players intuitively decrease their *desired allocated machinery* to minimum. Since there is delay, initially the game does not respond. Hence, players continue to give very low *desired allocated machinery* decisions. Therefore, the inventory undershoots the target. This process continues usually by creating damping oscillations. When the delay duration is long, the period of the cycles gets larger.



(a) Player 6, Trial 4: Setting O3T4

Figure 9: A typical behavior from the game involving *delay*.

Figure 10 shows typical behaviors from games involving *nonlinearity* and *feedback*. As compared to the *delay* games, in the games involving *nonlinearity* and *feedback*, subjects show almost perfect behaviors. The capacity restriction imposed by the *nonlinearity* is not influential because the players quickly bring *inventory* to the vicinity of the *target inventory*, where the required adjustments are minor compared to the capacity. Once the players discover the *desired allocated machinery* region keeping the *inventory* around the target, they manage to control *inventory* with small adjustments. Different levels of *nonlinearity* alter the location of this optimal region of *desired allocated machinery*, but the players quickly adapt to this alteration, since there is no delay. Similarly, *feedback* cannot show its detrimental effect since the players quickly reach the target, where feedback is ineffective.



(a) Player 9, Trial 6: Setting N3

(b) Player 19, Trial 6: Setting F8

Figure 10: A typical behavior from the game involving *nonlinearity* and *feedback*.

Table 3 shows some selected subject comments collected after the completion of all games. The comments show that subjects had a good grasp of the task requirements and the involved complexity factors. Their descriptions of their decision strategies represent decision heuristics that would yield a good performance score. However, especially for the *delay* case, their lack of knowledge about the levels of the complexity factor involved in the games (i.e. duration and order of the delay) led to poor performances.

Table 3: Selected subject comments.

Player 1 (Delay Group): "I discovered that there is a delay between my decision and output. I tried to adjust the inventory with 1st, trying extremes and then gradually decreasing or increasing to an estimated equilibrium level."
Player 3 (Delay Group): "When delays were long, I kept the production at the extremes. When there were shorter delays, I played more with the slider."
Player 10 (Nonlinearity Group): "If inventory [is] drastically lower than target, use maximum allocation, if inventory [is] drastically [higher] than target level, minimize allocation. Tried to understand the trend."
Player 14 (Nonlinearity Group): "The hardest part is to bring [the inventory] close to 200 in the beginning. The difficult part is to determine the effect of an increase or decrease in the number of machines on the inventory, in these first steps. The rest moves on easier."
Player 19 (Feedback Group): "Initially, I reduced the desired allocated machinery to its minimum. Then, I seeked for a equil. value."
Player 20 (Feedback Group): "I just wanted to keep the inventory level close to desired level. When it fell below the line, I increased the desired allocated machinery and vice versa."

4.2 Quantitative Analysis

4.2.1 Performance Measure

The performance measure is called *relative deviation from target*, defined as:

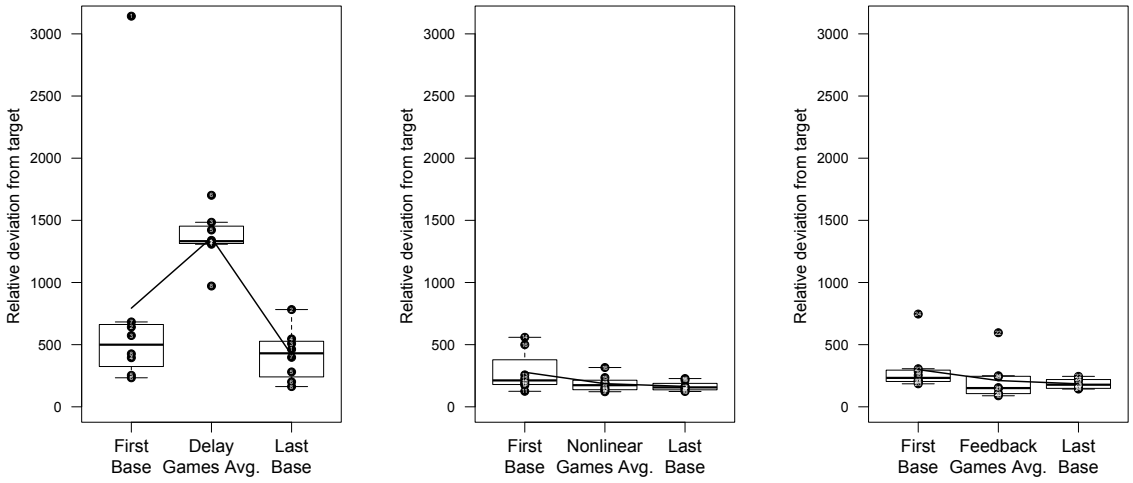
$$\textit{Cumulative deviation from target} - \textit{Benchmark's cumulative deviation from target} \quad (1)$$

Recall that the benchmark behavior is the best possible behavior. By deducing the minimum possible deviation from target from the players' total deviation, we make sure that we make a fair comparison between different game versions. Taking the difference of these two variables makes more sense than taking their ratio, because the variations between game versions that create changes in the benchmark scores do not cause a change the problem scale. They only affect the initial transient period. Since the problem scale does not change, the absolute value of any deviation from the target should be treated identically.

We also analyze subjective difficulty ratings collected from the subjects. These ratings provide valuable information because they reflect the perceived difficulties of the games that may not be reflected to objective performances. On the other hand, this measure may not be able to capture small changes in complexity because of its discrete nature and low resolution.

4.2.2 First versus Last Base Games

First, we compare the scores of two base games, which are played as first and last games (See Table 2). As Figure 11 shows, the difference is significant in *nonlinearity* and *feedback* groups (p-values: 0.0342 and 0.0464, respectively), indicating an improvement due to learning gained in the middle nine games. However, *delay* group players' last base game performances are not superior to their first base game performances, despite the experience and the fact that it is the same base game played at the beginning. Also note the disparity between *delay* group's last base scores and other groups' last base scores (Third column of Figure 11(a) versus 11(b) and 11(c)). Last base game performance in the *delay* group is worse compared to other two groups. These two observations show that, while experience with *nonlinear* and *feedback* games contributes to the performance in the base game, experience with *delay* games does not. The reason for this difference is that *delay* games are less similar to the base game, as compared to the other game versions. Indeed, as further explained below, while *nonlinear* and *feedback* does not worsen players' performances, *delay* brings a significant deterioration.



(a) *Delay* group (b) *Nonlinearity* group (c) *Feedback* group

Figure 11: Comparison of first base game, average of trials 3–10 (games involving complexity elements) and the final base game scores. The lines connect the means.

4.2.3 Base Games versus Games Involving Complexity Factors

In this part, we calculate the average scores of eight games played in trials 3–10 (each average is represented by a point in the middle columns of Figure 11) and carry out one-sided paired t-tests for the differences between these average scores and the scores of two base games.

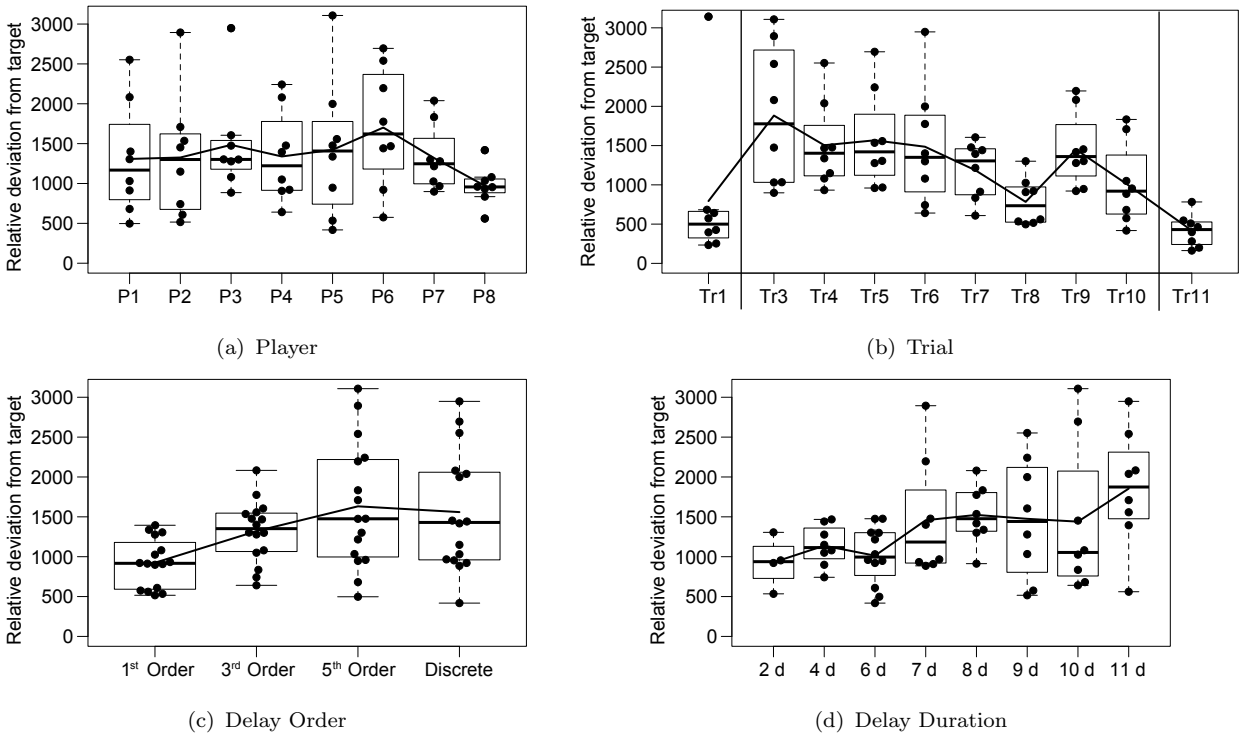
The results show that games involving *delay* yield significantly worse scores than both base games (p-values: 0.0797 for first base–delay games average difference; 0.00002 for last base–delay games average difference). However, the average scores of *nonlinear* games is not inferior to the base game scores. Furthermore, there is an improvement from the first base to *nonlinear* games (p-value: 0.0489), and from *nonlinear* games to the last base (p-value: 0.1004). This continued improvement indicates a clear learning effect. The average scores of the games involving *feedback* are not statistically different than neither of the first

and last base games (p-values: 0.1412 and 0.3029, respectively). However, Figure 11(c) indicates a slight improvement from the first base to the *feedback* games.

In the following sections, we further analyze the game results for trials 3–10, where the subjects play games involving complexity factors. Since every level of each complexity factor is played in every order, we can identify the effects of factors as well as progress of scores between trials.

4.2.4 Delay Group

Figure 12 shows box plots of *relative deviations from target* for the levels of *player*, *trial*, *delay order* and *delay duration* factors. Table 4 shows the ANOVA table of the same group. These results show that there is a significant variance between different players as well as trials. As players get experience, the scores show a decreasing trend, indicating a learning effect.



(a) Player

(b) Trial

(c) Delay Order

(d) Delay Duration

Figure 12: Box plots showing effects of different factors on game score (*relative deviation from target*) in the *delay* group. The lines connect the means.

As the ANOVA table and Figure 13 show, there is a strong interaction effect between *delay order* and *delay duration*. While *delay duration* is less effective on the scores when *delay order* is low, its effect becomes more noticeable in high delay orders. These results are in agreement with the qualitative observations: higher order and longer delays result in worse performances.

Table 4: ANOVA table for the *delay* group scores.

	Df	Sum Sq	F value	p-value	
Player	8	5975318	3.3505	0.003628	**
Trial	1	2454548	11.0104	0.001659	**
Delay Order	1	128073	0.5745	0.451898	
Delay Duration	1	146	0.0007	0.979674	
Delay Order × Delay Duration	1	1140347	5.1153	0.027924	*
Residuals	52	11592332	52		

Signif. codes for p-values: ***: 0–0.001, **: 0.001–0.010, *: 0.010–0.050, .: 0.050–0.100
Adjusted $R^2 = 0.90$

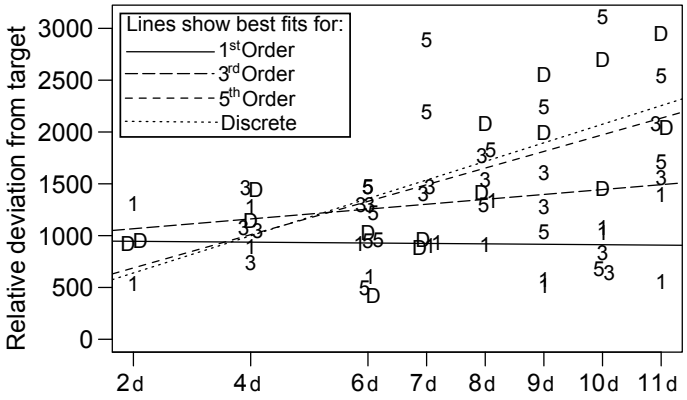


Figure 13: Interaction between *delay order* and *delay duration*.

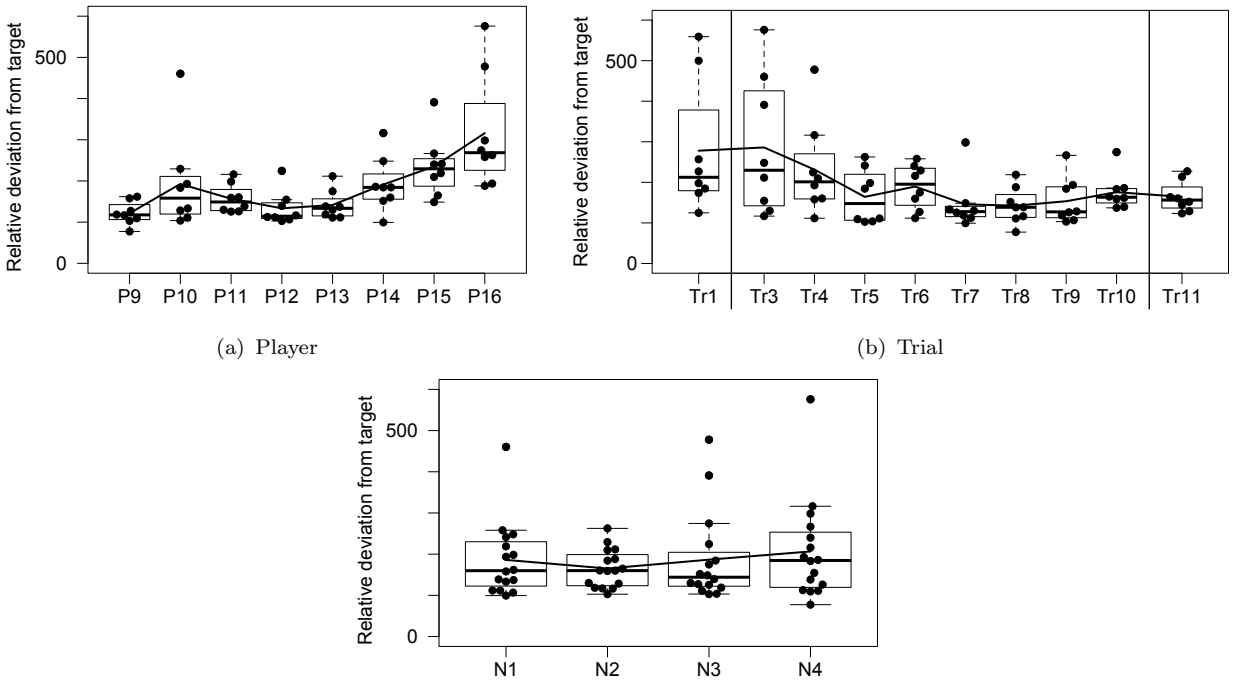
4.2.5 Nonlinearity Group

Figure 14 and Table 5 show the analysis results for the *nonlinearity* group. Note that the scores of *nonlinearity* group are much lower than the scores of the *delay* group (Notice the difference in the vertical scales of the plots). Like the *delay* group, *player* and *trial* effects are significant. Learnings reaches a saturation after a certain number trials. Also, the scores of the base games and nonlinear games follow a continuum (Figure 14(b)), indicating that learning from one version can be transferred to the others. Level of *nonlinearity* does not have a significant effect on the performance, all nonlinear functions yields similar performances. Even the scores of games involving extreme *nonlinearity* are not statistically different from the base game scores.

Table 5: ANOVA table for the *nonlinearity* group scores.

	Df	Sum Sq	F value	p-value	
Player	8	549867	14.9141	2.948e-11	***
Trial	1	76837	16.6724	0.0001479	***
Nonlinearity	1	5508	1.1953	0.2791258	
Residuals	54	248865			

Signif. codes for p-values: ***: 0–0.001
Adjusted $R^2 = 0.89$



(a) Player

(b) Trial

(c) Nonlinearity

Figure 14: Box plots showing effects of different factors on game score (*relative deviation from target*) in the *nonlinearity* group.

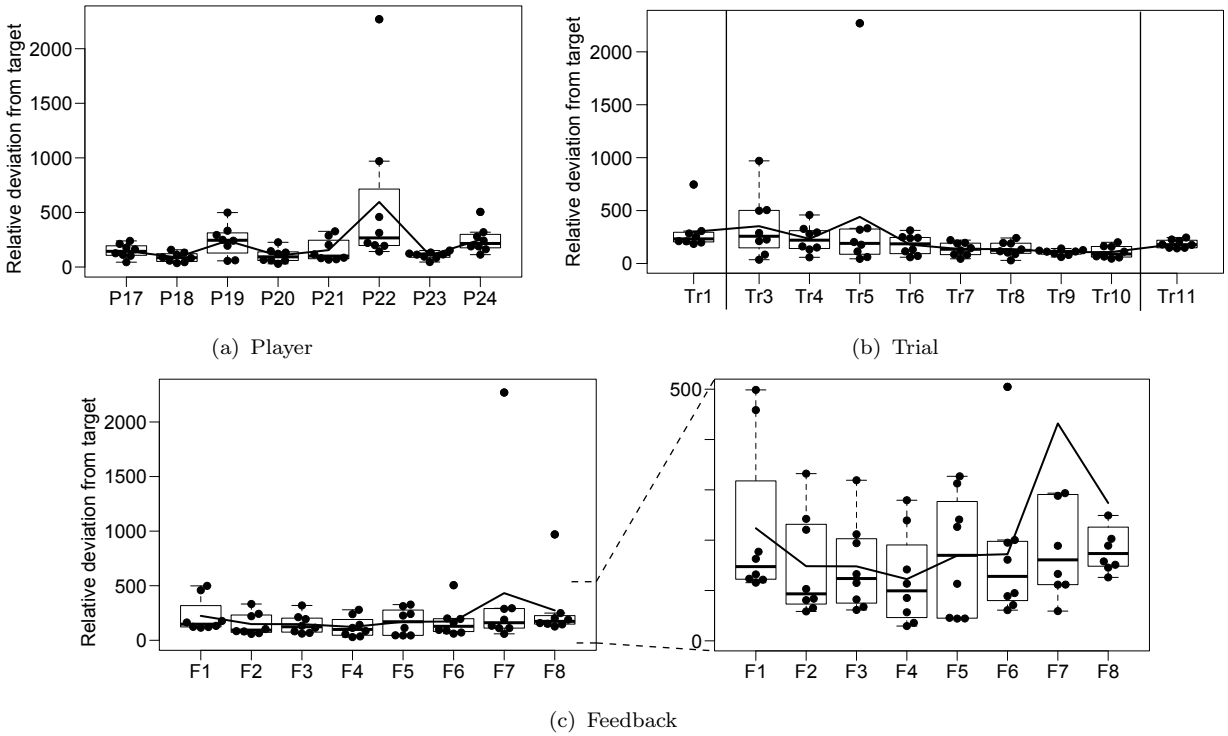
4.2.6 Feedback Group

Figure 15 and Table 6 summarize the analysis results of the *feedback* group. The *player* and *trial* effects are again significant. The strength of the *feedback* does not have a significant influence on the results. Yet, Figure 15(c) suggests that *feedback* has two types of effects depending on its level. When the *feedback* is weak, increasing feedback seems to have improve the scores. t-tests show that feedback levels F2—F4 yield significantly better scores than the base game scores. This improvement might be due to the balancing effect of the *negative* feedback loop between *inventory* and *sales* (Figure 4). As the level of *feedback* factor increases, the *positive* feedback loop starts to show some influence and deteriorate the score. However, the effect of positive feedback is not so strong to create a statistically significant difference.

Table 6: ANOVA table for the *feedback* group scores.

	Df	Sum Sq	F value	p-value	
Player	8	2217334	4.3710	0.0004078	***
Trial	1	515448	8.1288	0.0061588	**
Feedback	1	169887	2.6792	0.1074831	
Residuals	54	3424134			

Signif. codes for p-value: ***: 0–0.001, **: 0.001–0.010
Adjusted $R^2 = 0.52$



(a) Player

(b) Trial

(c) Feedback

Figure 15: Box plots showing effects of different factors on game score (*relative deviation from target*) in the *feedback* group.

4.2.7 Subjective Difficulty Assessments

We repeated the statistical analysis reported above using our other measure of complexity: subjective difficulty assessments. Unlike the game score, subjective difficulty ratings can take only integer values from 1 to 9.

Figure 16 presents a summary of the results. *Delay duration* and *delay order* influence the subjective difficulty ratings as they affect the game scores. Although their interaction effect is not statistically significant, the effects of both *delay duration* and *delay order* are significant (p-values < 0.001).

Unlike the game scores, *nonlinearity* has a weak influence on perceived difficulty (p-value: 0.094). The games involving extreme nonlinearity (N4) are perceived to be slightly difficult.

As Figure 16 indicates, *feedback* seems to have a varying effect on subjective difficulty ratings, as it has on the game scores. First, the perceived difficulty drops until F5. After F5, it shows a rise. Statistical tests show that subjective difficulty ratings of F4 and F5 games are significantly lower than that of F7 and F8, also F1.

In general, subjective difficulty assessments are in parallel to *relative deviations from target* (See Figure 17). Overall, there is a +0.58 correlation between two performance measures. This indicates that subjects perceived the game as easier when they performed better. This observation is in accordance with the result in the growth management game (Özgün and Barlas, 2011).

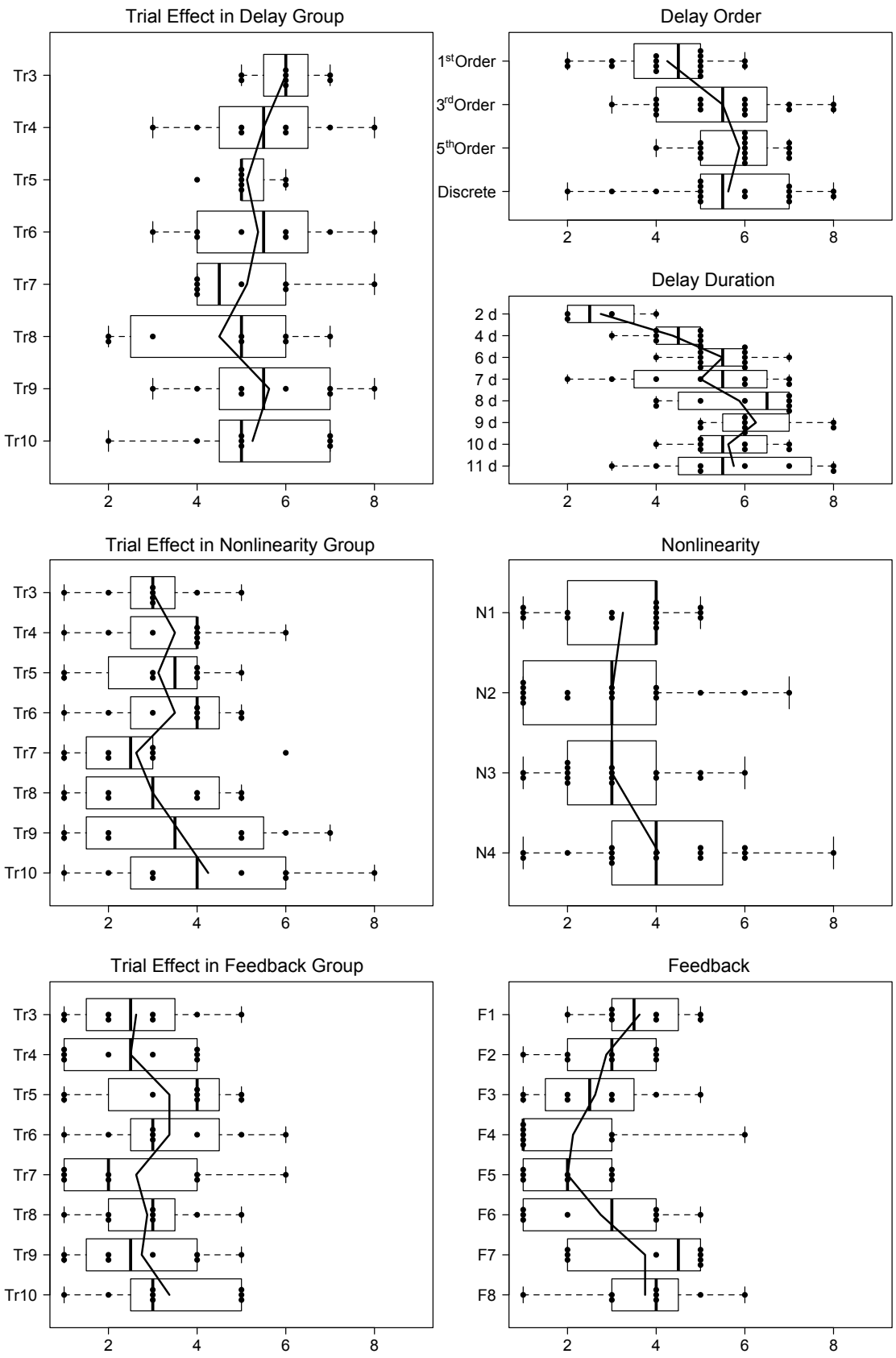


Figure 16: Summary of results for subjective difficulty ratings.

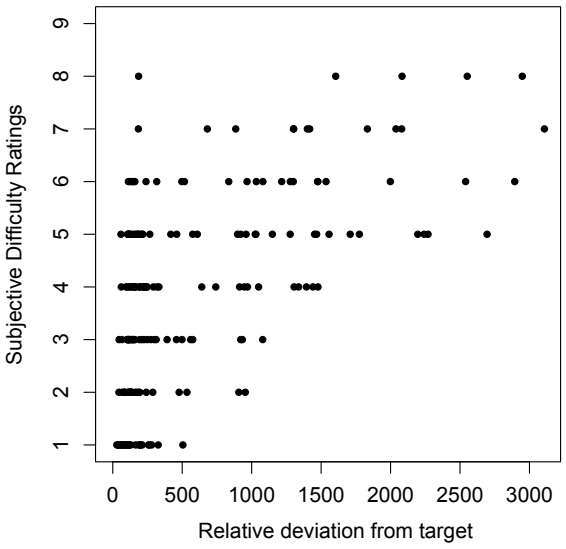


Figure 17: Subjective difficulty ratings versus game performance scores.

5 Discussion

This paper presents a simulation experiment in which effects of three systemic complexity factors on the complexity of the simulation game are tested. We designed a simple stock management game in which players need to bring *inventory* level to equilibrium. We, then, modified the game to obtain game versions involving *delay*, *nonlinearity* and *feedback*.

The factor levels are changed in the experiments by changing the delay order, delay duration, shape of nonlinear functions and gain of feedback loop in the simulator. Each complexity factor brings different difficulties to the game. *Delay* makes the difficult-to-control because players cannot see the results of their actions immediately, giving rise to oscillations around the target. *Nonlinearity* distorts the proportionality between actions and their consequences, causing the game to seem unresponsive in one place and over-responsive in another place. *Feedback* reinforces players' decisions as their inventory move away from the target, making it even more difficult to reach the target. We measure the game performance in terms of *relative deviation from target*, which is the difference between player's inventory deviation from target and minimum possible deviation.

5.1 On the effects of systemic complexity factors

Among three game groups involving complexity factors, only *delay* results in worsening in the game performance. The impact of *delay* gets more effective as *delay order* and *delay duration* increase. This results is verified by subjective complexity assessments. The growth management task experiments also indicated a strong *delay* effect (Özgün and Barlas, 2011). In that sense, two experiments agree that *delay* is the most effective systemic complexity factor.

Even in their extreme levels, *nonlinearity* and *feedback* do not have any significant negative influence on the performance. Only *nonlinearity* has a slight effect on subjective complexity ratings. *Feedback* even has a positive effect when it is at low levels, due to the fact that

we add two antagonist feedback loops: a negative feedback loop that helps stabilizing *inventory*, and a positive feedback loop that opposes stabilization. While the strength of the negative feedback is kept constant, the strength of the positive feedback loop increases with increasing level of *feedback* factor. Therefore, when *feedback* is at low levels, it helps the players to reach the target. As level of *feedback* increases, this favorable effect is removed.

5.2 On learning

All subject groups exhibit performance improvement with repeated trials. In contrast, in the growth management game, learning through trials was only observed in the *nonlinearity* group, the only game version not including a stock. Several factors contribute to this learning difference between two games. First, the growth management game had two decision variables instead of one. Having two decision variables considerably enlarges the feasible space. This complicates the process of finding a good strategy and limits learning. Another obstacle of the growth management game is the side effects of decisions that are effective in the presence of delays. Given that people have problems taking the side effects into account (Brehmer, 1992; Sterman, 1989a), it is not surprising that it further complicates learning. The stock management game used in this study creates a more salient environment, which enables learning. First, the decision space is one-dimensional. Second, the movement direction of the outcome (*inventory*) depending on the movement direction of the input (*desired allocated machinery*) is already apparent. Third, the actions do not have any unintended side effects.

Although all groups exhibit learning by trials within the games involving the same complexity factor, different complexity factors have different influences on the performance on the last base game. Subjects' performances are superior in their last base game with respect to their first base game, if they play *nonlinearity* or *feedback* games in-between. However, the performance does not improve if they play *delay* games in between. Such an effect was also observed in the growth management game experiments, and in a stronger way. (Some subjects' performances in the last base game even deteriorated after playing *delay* games.) These results indicate the importance of complexity in transfer of learning between games.

5.3 Further Research

A possible extension of this study is testing the interactions between the complexity factors. In this study, we observed that *nonlinearity* and *feedback* did not give rise to deteriorating performance. But in interaction with other complexity factors, they may have significant effects on the results. Also, the influence of the factors may vary in interaction. In a different experimental design, we are testing the interaction effects of these complexity factors.

In further experiments, we are planning to test players' conceptual learning of the underlying system. The conceptual learning will be measured by the scores from questionnaires about the underlying system structure, and by testing subjects' performances in different games with similar structures, but embedded in different contexts.

References

Arango, Santiago Aramburo. 2006. Essays on commodity cycles based on expanded cobweb experiments of electricity markets. PhD diss., University of Bergen.

Atkins, Paul W. B., Robert E. Wood, and Philip J. Rutgers. 2002. The effects of feedback format on dynamic decision making. *Organizational Behavior and Human Decision Processes* 88(2):587–604.

Bakken, Bent Erik. 1993. Learning and transfer of understanding in dynamic decision environments. PhD diss., Massachusetts Institute of Technology.

Barlas, Yaman, and Mehmet Günhan Özevin. 2004. Analysis of stock management gaming experiments and alternative ordering formulations. *Systems Research and Behavioral Science* 21:439–470.

Brehmer, Berndt. 1992. Dynamic decision making: Human control of complex systems. *Acta Psychologica* 81(3):211–241.

———. 1995. Feedback delays in complex dynamic decision tasks. In *Complex problem solving: The european perspective*, ed. P. Frensch and J. Funke, 103–130. Lawrence Erlbaum, Hillsdale, NJ, U.S.A.

Broadbent, Donald E., and Ben Aston. 1978. Human control of a simulated economic system. *Ergonomics* 21:1035–1043.

Diehl, Ernst W. 1989. A study on human control in stock adjustment tasks. In *Proceedings of the 7th international conference of the system dynamics society*, 205–212. Stuttgart, Germany.

Diehl, Ernst W., and John D. Sterman. 1995. Effects of feedback complexity in dynamic decision making. *Organizational Behavior and Human Decision Processes* 62(2):198–215.

Kampmann, Christian Peter Erik. 1992. Feedback complexity and market adjustment: An experimental approach. PhD diss., Massachusetts Institute of Technology.

Langley, Paul A., Mark Paich, and John D. Sterman. 1998. Explaining capacity overshoot and price war: misperceptions of feedback in competitive growth markets. In *Proceedings of the 16th international conference of the system dynamics society*. Québec City, Canada.

Özgün, Onur, and Yaman Barlas. 2011. Analysis of the effects of different complexity factors on the complexity of a simulation game. In *Proceedings of the 29th international conference of the system dynamics society*. Seoul, Republic of Korea.

Paich, Mark, and John D. Sterman. 1993. Boom, bust, and failures to learn in experimental markets. *Management Science* 39(12):1439–1458.

Sterman, John D. 1989a. Misperceptions of feedback in dynamic decision making. *Organizational Behavior and Human Decision Processes* 43(3):301–335.

———. 1989b. Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Management Science* 35(3):321–339.

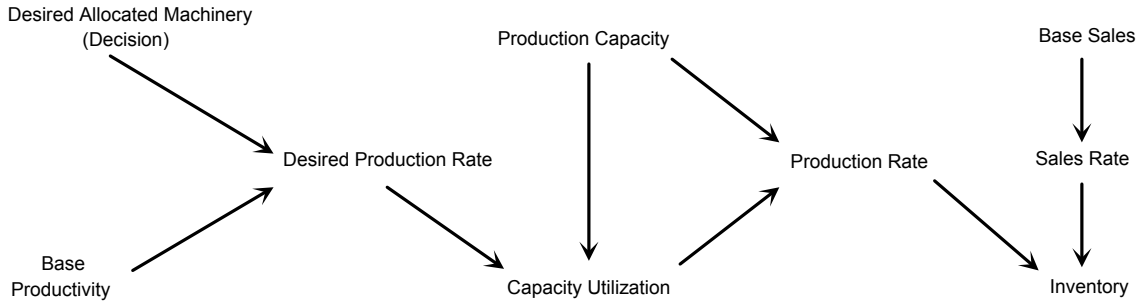
Tawarmalani, Mohit, and Nikolaos V. Sahinidis. 2005. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming* 103:225–249.

Young, Showing H., Chia Ping Chen, Sy-Weng Wang, and Chi Hung Chen. 1997. The landmine structure and the degrees of freedom of decision-making. In *Proceedings of the 15th international conference of the system dynamics society*, 15–20. Istanbul, Turkey.

Appendices

A Game Instructions

This interactive simulator is about a company that produces textile products. The company operates in a hypothetical world and all the rules of economy may not work as they do in the real world. You are the production manager who is only responsible from the production of t-shirts and your aim is to eliminate the oscillations in the t-shirts inventory. Your inventory level increases with production and decreases with sales. Backlogs are allowed. The production rate is the number of boxes of t-shirts produced per day. As the production manager responsible from t-shirts, you can determine the desired number of allocated machines to t-shirt production (*Desired Allocated Machinery*). The figure below gives a broad representation of the causal relationships between key variables. Note that this is a general overview and there may be some other causal links in particular game versions you play.



By setting *Desired Allocated Machinery*, you will determine your *Desired Production Rate* (boxes/day) which is the product of your *Desired Allocated Machinery* (machines) and *Productivity* of the machines (boxes/machine/day). The base value of *Productivity* is 1 box/machine/day. The actual *Production Rate* is the product of *Production Capacity* (boxes/day) and *Capacity Utilization*. *Capacity Utilization* is a variable that shows the percentage of the *Production Capacity* used in t-shirt production. The *Production Capacity* is constant at 50 boxes/day. *Capacity Utilization* function yields a *Production Rate* which is equal to your *Desired Production Rate*, in the base game. The *Capacity Utilization* is not so rigid and may be somewhat above 1 as well, which will indicate overtime production. The *Sales Rate* is the number of units sold per day. It is dependent on unknown

random *Base Sales*. *Sales Rate* immediately decreases the *Inventory* level and there is no delay or distortion in observing the *Inventory* level.

As the production manager, you have one decision to control the *Inventory*: **desired allocated machinery**. You have no direct information regarding the *Sales Rate* or the *Capacity Utilization*. Depending on the game you play, there may or may not be delays in adjusting the allocated machinery. The basic challenge in the game is to control and stabilize the *Inventory* without a direct information regarding the *Sales Rate*.

You will decide on desired allocated machinery for 40 days and **your objective is to stabilize the inventory around the target level of 200 boxes as quickly as possible**. Your performance will be assessed by the total deviation from the Target Inventory. You will start from an off-equilibrium condition and seek the target level.

You will play 11 different games. Each game will be independent from each other. The first game will be the base game and will have the simple underlying structure explained above. The second game will be a much difficult game and will be a modified version of the first game. The remaining eight games will be different from the base game **only** in one aspect: there will be a time delay between your decision and its effect on production rate, and the duration and order of this delay will be varied from game to game. To be more specific, the *Production Rate* will be a delayed function of *Desired Production Rate*. Since there is delay, your decisions will not be immediately effective on *Inventory*. The games may have different initial conditions and parameter values, so a specific strategy that works in one game may not automatically work in another game. Finally, at the very end of the experiments you will play the simplest base game once again.

You will be asked to assess the difficulty of achieving success in each game on a scale from 1 to 9 as shown below.

EXTREMELY EASY		EASY		AVERAGE		HARD		EXTREMELY HARD
1	2	3	4	5	6	7	8	9

The base game (the first and the last game) is already assigned a difficulty of 1. The second game with a much higher difficulty has a pre-assigned difficulty measure of 7 as a reference. You have to rate the difficulty of the remaining eight games. There is no "correct" answer in the difficulty assessments. You may assign the same difficulty to two or more games, if you think they are at the same difficulty level. You do not have to utilize the entire scale up to 9. If you think the games are not that hard, you may assign all of them difficulties lower than 9. There will be no systematic playing order with respect to difficulty. After each game, circle your difficulty assessment of the game on the sheet provided. You can revise your previous assessments after playing and observing the difficulties of the succeeding games. At the end of the 11th game, you will be asked to finalize your rating list and return it to the facilitator.

(Figure 2 is shown here.)

Do not open or play the games before you are told to do so. You will play the games in a specific order as indicated by numbers. For opening a game, double click the file. The game screen is as shown above. When you open the game file click the Start button once

to start the game. This will initialize the game and advance you to the first day. The *Inventory* will start above or below the Target Inventory level. You cannot change the first days *Inventory* so do **not** move the sliders **before** clicking the Start button. Each day, you must set a *Desired Allocated Machinery* value using the slider and click the Advance button once. You will observe the *Inventory* behavior on the graph in blue and see its numerical value in a blue box above the graph. You will also see the constant *Target Inventory* on the same graph in red. When you complete 40 days, a warning box will appear. When you finish the game you should (1) write down your *Total Deviation* and your difficulty assessment on the sheet provided, (2) click the Exit button and (3) **save the game** when you are asked. Do not play any game more than once, pass to the next game. If you did something by error that you did not intend to do, please stop immediately and inform the facilitator. You will have a trial game at the beginning for you to familiarize with the game interface. Please take your time to experiment with the controls and understand how they work.

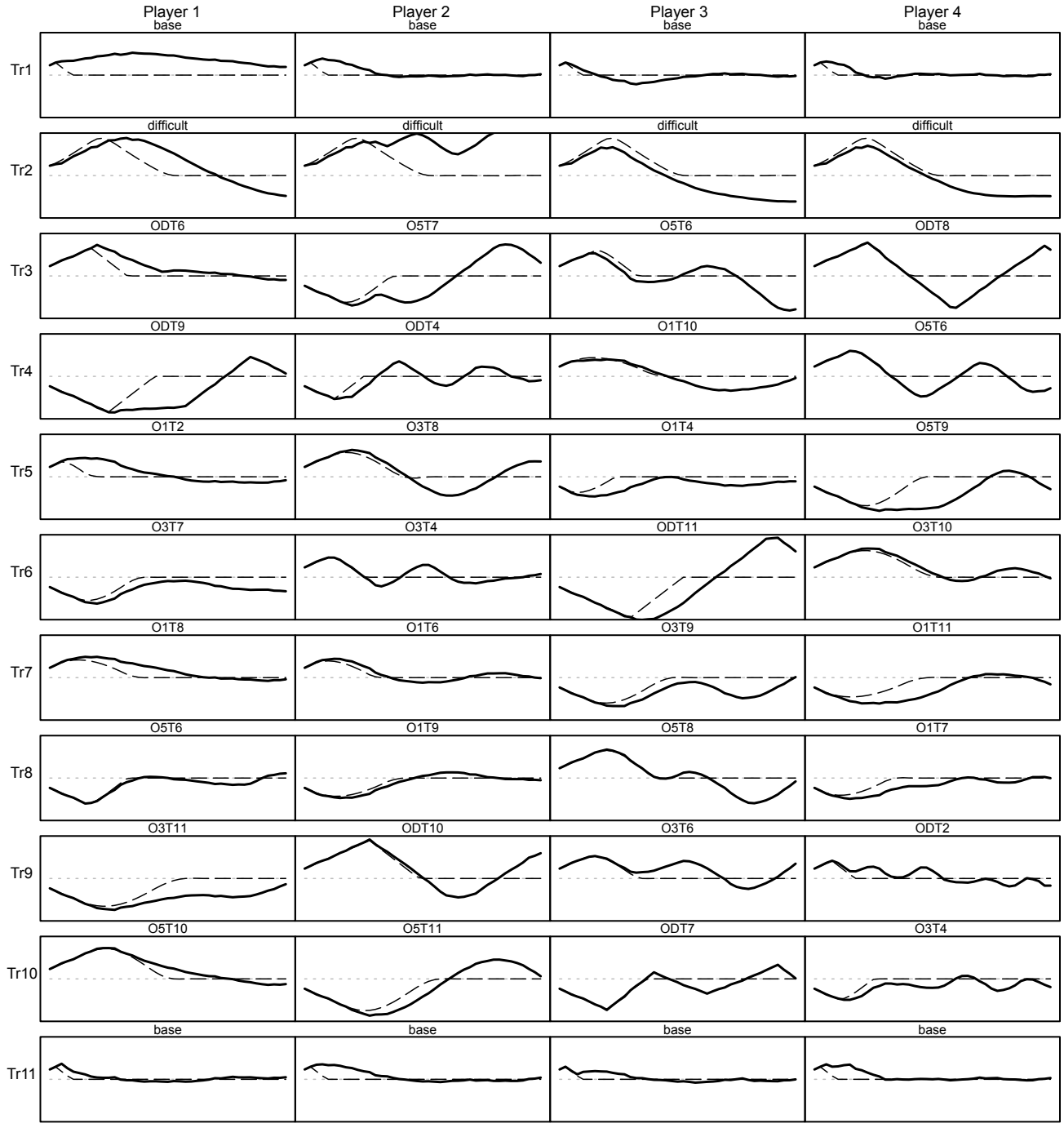
Make sure that you understand the instructions completely before you start the experiments. If there is **anything you do not** understand, please ask your questions before you start playing. It is important that you know what you have to do in the experiments. For the validity of the results, it is necessary that the experiment be carried out as intended. Work on your own and do not talk to the other subjects.

You must save the game files and fill out the game sheet for the proper completion of the experiment. If you complete the experiment properly, you will earn a reward depending on your performance in the games that you play. If you show the best performance among four players playing identical games with you, you will earn a reward of 18 TL. Your reward will be 10 TL, 7 TL and 5 TL if you get the second, third or fourth place, respectively. Only performance scores will be used in determining reward amounts. Your difficulty assessments will **not** have any effect on your payment. Thank you for your participation.

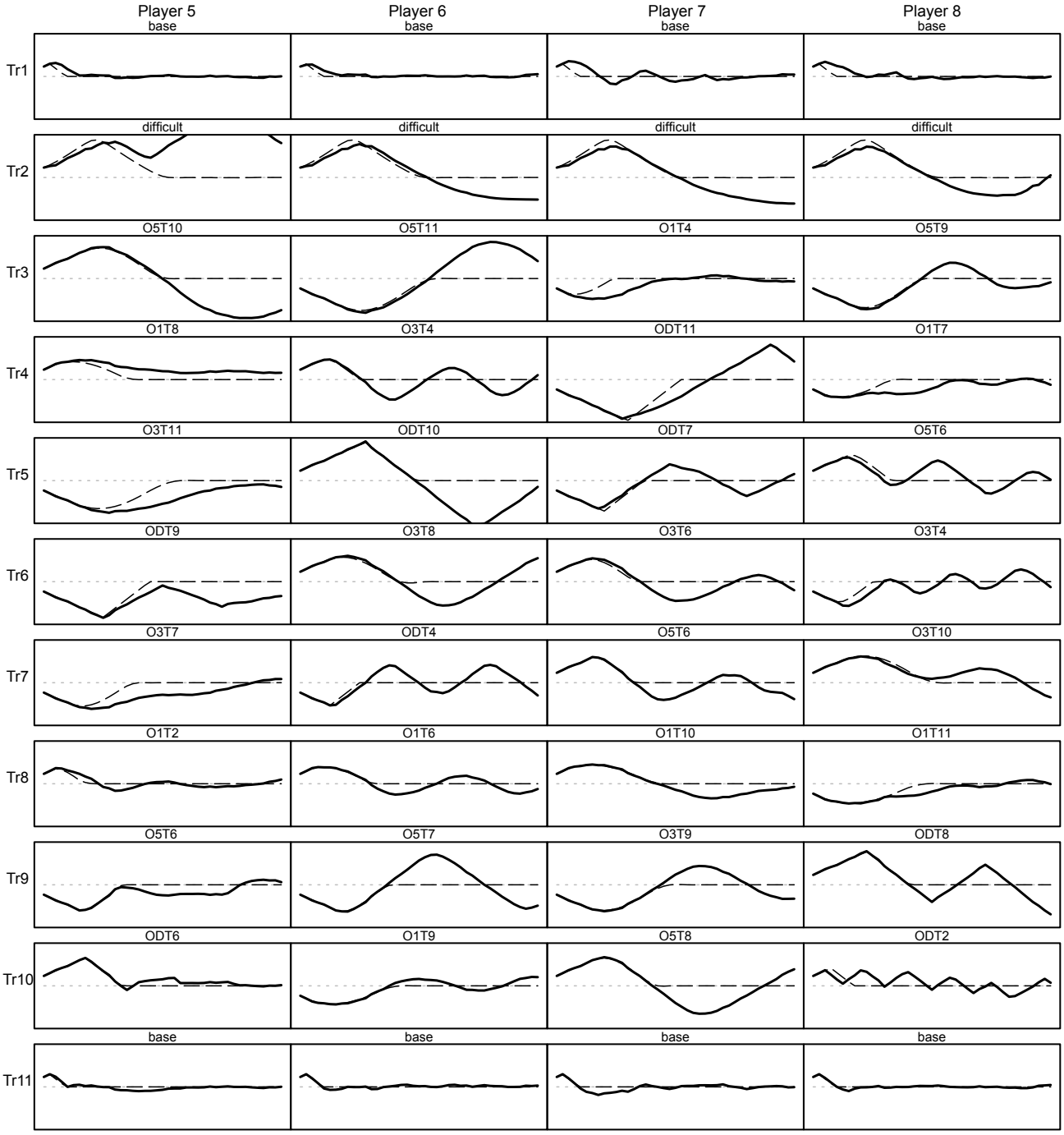
B Game Behaviors

The following figures show the behaviors of *inventory* for the subjects (solid line) and for the benchmark (dashed line). The columns show players and the rows show playing order. The game versions are indicated above each plot. The vertical scale is 0–400.

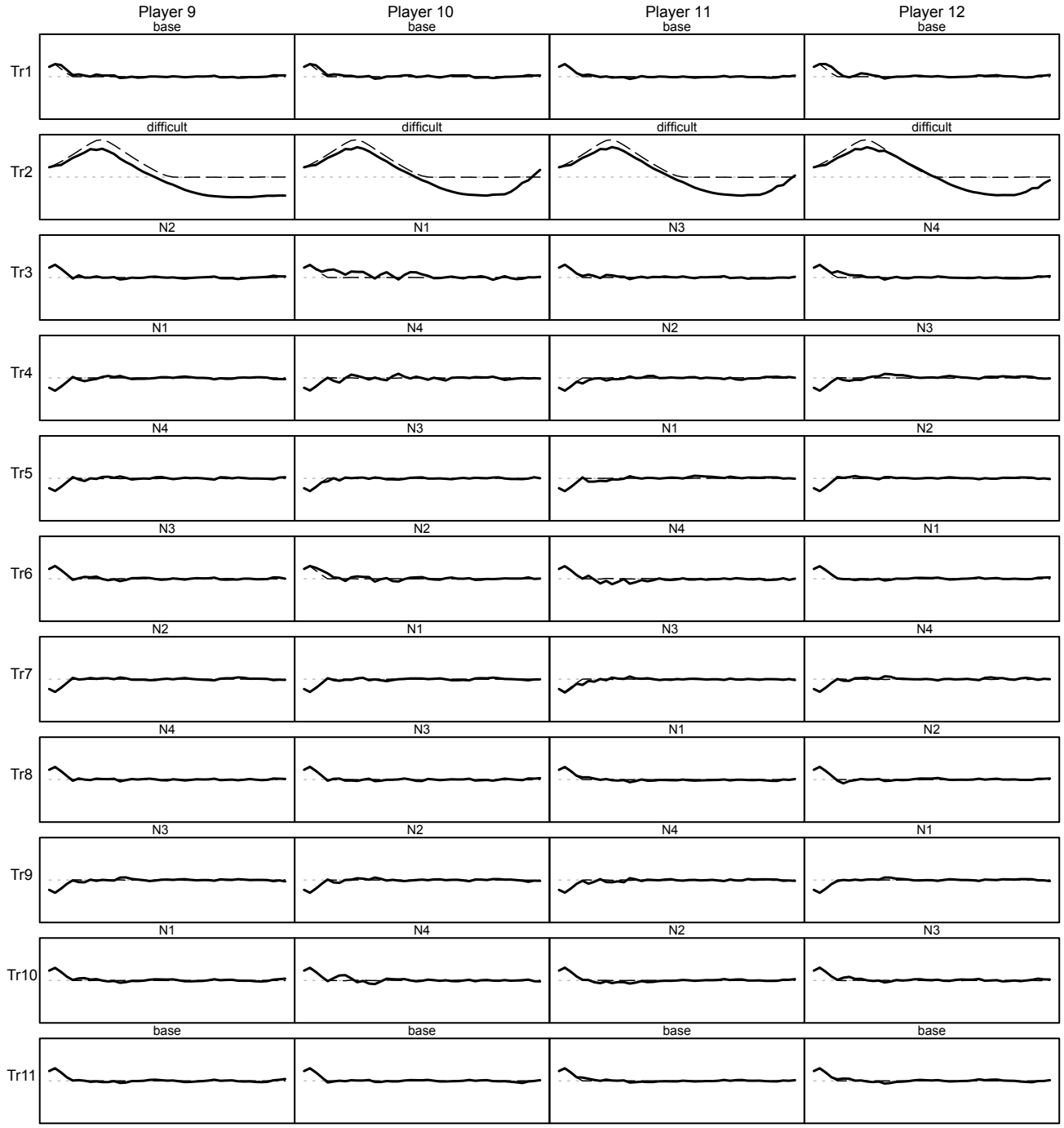
Delay Group



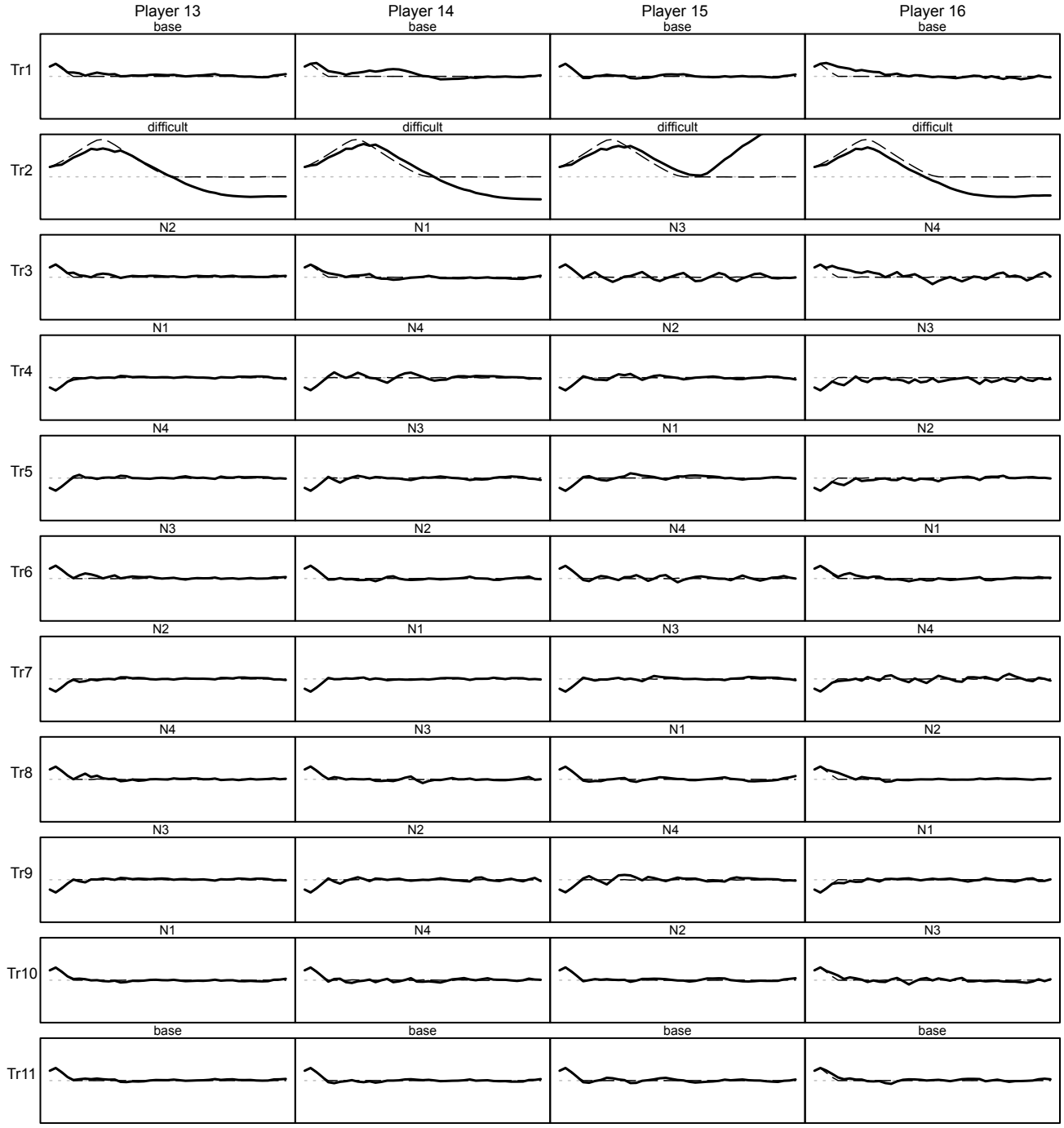
Delay Group


Player 5
Player 6
Player 7
Player 8
Tr1
base
Tr2
difficult
Tr3
O5T10
O5T11
O1T4
O5T9
Tr4
O1T8
O3T4
ODT11
O1T7
Tr5
O3T11
ODT10
ODT7
O5T6
Tr6
ODT9
O3T8
O3T6
O3T4
Tr7
O3T7
ODT4
O5T6
O3T10
Tr8
O1T2
O1T6
O1T10
O1T11
Tr9
O5T6
O5T7
O3T9
ODT8
Tr10
ODT6
O1T9
O5T8
ODT2
Tr11

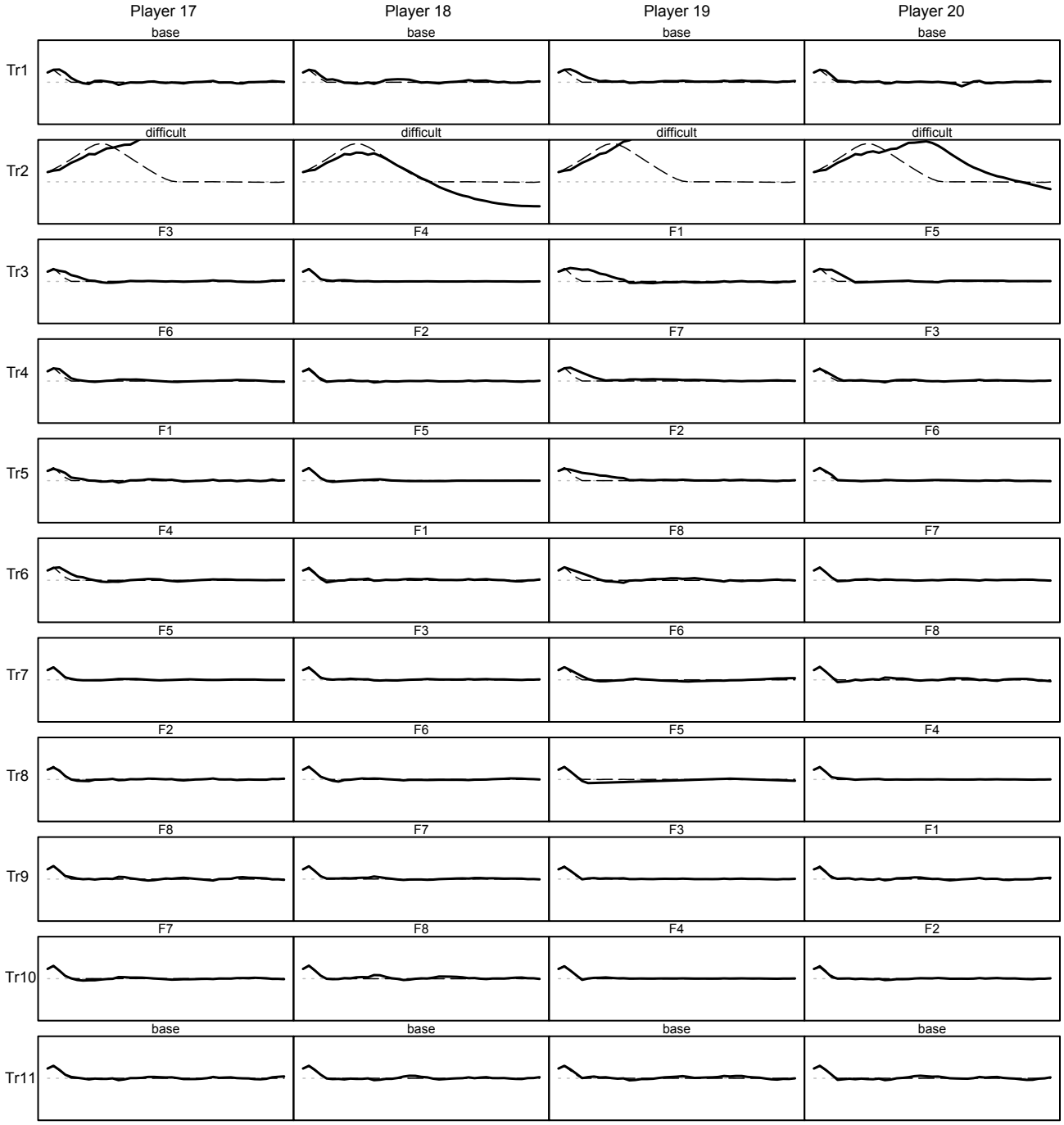
Nonlinearity Group



Nonlinearity Group



Feedback Group



Feedback Group