Software Quality, Killer Applications, and Network Effects:

The Case of the U.S. Home Video Game Industry

Corresponding Author: Richard T. Gretz

Bradley University

1501 W. Bradley Ave., Peoria, IL 61625

Office: (309) 677-2298, FAX: (309) 677-4174, rgretz@bumail.bradley.edu

Jannett Highfill

Bradley University

Abstract

This paper uses an extension of CES preferences to empirically derive a continuous game quality measure for the home video game industry (January 1995 to October 2007). Estimates based on the quality measure are more consistent with theory than estimates based on number of available games. Hardware market share elasticity is estimated to be 13.04% for killer applications and insignificant for non-killer applications. However, the indirect network effect is 2.75 times larger for non-killer applications than killer applications. Our results suggest indirect network effects on hardware market share may be overstated if differences between software qualities are not considered. Finally, higher proportions of killer applications are associated with successful entry and generational leadership even when competitors offer a larger selection of games.

Keywords: killer applications, video games, indirect network effects, hardware and software

JEL Classification: L13; L63; D43

1. Introduction

Positive network externalities exist when consumer benefit increases with the number of adopters of a network. These externalities are indirect when an increase in consumers results in an increase of complementary products provided for the network. The video game industry is an example, more and better software is made available for a video game system as more consumers adopt that system; consumers are more likely to adopt a system that has more and better available software.[1] While network effects on primary good market shares have received a good deal of attention in both the theoretical literature,[2] and the empirical literature,[3] as far as we are aware, both literatures consider only symmetric complementary products. This assumption completely ignores the possibility of "killer applications"—complementary products so valuable their presence alone may induce a considerable amount of consumers to purchase access to a network.

As the present paper shows, the theory to accommodate asymmetric complementary products is a straightforward extension of CES preferences often used to model network effects (Chou and Shy, 1990; Church and Gandal, 1992; Park, 2002; Nair et al., 2004; Clements and Ohashi, 2005). One of the primary contributions of the paper is using this extension to empirically derive game quality from a unique data set covering a significant portion of the life of the home video game industry (monthly from January 1995 to October 2007). This allows us to estimate the demand elasticity for hardware market share with respect to a quality index for software.

[1] On the other hand, these externalities are direct when the attractiveness of a network increases directly with the number of consumers; for example, a telephone network.
[2] See Farrell and Saloner, 1986; Katz and Shapiro, 1986, 1992; Church and Gandal, 1993; Woeckener, 2000; Shy, 2001; Park, 2002, and Clements, 2005.
[3] Symmetrical complementary products is probably a reasonable assumption in Gandal, et al. (2000) which examines the compact disk industry, Nair, et al. (2004) which examines the personal digital assistant industry, and Park (2004) which examines video cassette recorder industry.

We perform two hardware estimations for comparison, one employing the quality index for software and the other using the number games, separating games into killer applications and non-killer applications, in place of the former. Regardless of the specification, hardware estimation results suggest market share is highly elastic with respect to killer applications: this elasticity is estimated at 13.04% and 8.41% in the quality index specification and number of games specification, respectively. On the other hand, non-killer applications have an insignificant effect on market share in the quality index specification and a negative and significant effect in the number of games specification. We suspect the quality index specification offers more reasonable estimates given its results are more consistent with theory (i.e. additional games should not negatively affect market share).

We examine whether network effects differ between non-killer applications and killer applications given the large effect of the latter on hardware market share. We find the indirect network effect, or the elasticity of the number of games with respect to the installed base of consumers, is 2.75 (2.2% to 0.8%) times larger for non-killer applications than killer applications. Our results suggest estimates of the indirect network effect on hardware market share may be overstated if differences between killer applications and non-killer applications are not considered.

Our results regarding the effect of killer applications on hardware market share as well as differences in indirect network effects are used to discuss industry market share dynamics in a stylized manner. The data suggests consoles can successful enter a generation by having a higher percentage of killer applications relative to available games than an incumbent. Also, an established console's leadership position can be displaced by a later entrant with a smaller number of games but a larger concentration of killer applications. Finally, generational leaders often offer a higher percentage of killer applications than generational laggards.

That killer applications are an important characteristic of the home video game industry has been recognized in the literature (Clements and Ohashi, 2005; Coughlan, 2001, Gretz, *in press*; Hillis, 2001, and Shankar and Bayus, 2003). To highlight this, consider the fact that relatively few games perform well in the marketplace. Out of all video games released in 1998, only 10% made a profit. In 2000, only about a quarter of the 1300 games produced that year were able to sell enough copies to make back their development costs (Hillis, 2001). However, due to data restrictions, the empirical work closest to that of the present paper has assumed software is symmetric when estimating the impact of indirect network effects on hardware market share (see Gretz (*in press*), Clements and Ohashi (2005), and Sankar and Bayus (2003)). The methodology used here overcomes the insufficient data problem by allowing for empirical derivation of a continuous quality measure for complementary products based on readily available price and quantity data.

The paper will proceed by setting out the model, and then discussing the data, game quality, and killer applications. The latter are generally defined in terms of revenue. The empirical estimates of hardware shares are done next. This is followed by estimates of the indirect network effect on the number of killer applications and non-killer applications. Our results are then used to discuss dynamics of the hardware industry. The final section concludes by summarizing and using the quality index to obtain marginal effects on console market share of the top revenue generating games.

2. The Model

This section develops a model where hardware firms and software firms interact in an industry characterized by indirect network effects. To focus on this, assume consumers only purchase hardware systems in order to have access to compatible software titles. A hardware system becomes more attractive to consumers when they have access to higher quality software, and it is more attractive for software firms to provide for a hardware system with a larger network of consumers. The formal model is given below.

A partial equilibrium model is used to determine market share of competing hardware systems. Software is provided competitively by third party firms and is incompatible between hardware systems. For simplicity assume a software firm produces a single title for one hardware system. Consumers, software, and hardware are indexed by $i = 1 \cdots N$, $j = 1 \cdots J$, and $k = 1 \cdots K$, respectively. Consumers are identical except that installed base consumers, N_0, own hardware while new consumers, N_1, do not. (New hardware has no installed base consumers).

A new consumer i will choose to buy hardware k if doing so maximizes i's utility

$$U_i^k = E\left(U_{i,J^k}^k\right) + \varepsilon_i^k \qquad (1)$$

where $E(U_{i,J^k}^k)$ is i's expected utility of consuming J^k software titles compatible with hardware k, net of software prices, and $\varepsilon_{i,k}$ is an individual specific shock (formal assumptions on $\varepsilon_{i,k}$ are given in Section 4.1). The goal of the present section is to describe how U_{i,J^k}^k depends on software quality and the price of hardware k, that is, to provide structure for equation (1).

Suppose the agents play a multi-stage game with the order of play as follows. First hardware firms simultaneously set system price. Then new consumers decide which hardware system to purchase. After new consumers make their purchase decisions, compatible software firms enter the market competitively and set software prices. After this, both new and installed base consumers purchase compatible software. Note that only new consumers buy hardware while both new and installed base consumers buy compatible software. Subgame perfect Nash equilibria are computed using generalized backwards induction. As such, the software consumption decision is analyzed first.

Consumer i's utility for software compatible with hardware k is modeled using CES preferences as follows:[4]

$$\max_{x_j^k} U_{i,J^k}^k = x_0 + \left(\sum_{j=1}^{J^k} \left(x_j^k \theta_j^k \right)^{\frac{1}{\beta}} \right)^{\frac{1}{\alpha}}, \ \alpha \geq 1 \text{ and } \beta > 1 \tag{2}$$

where x_0 is a numeraire good, x_j^k and θ_j^k are the quantity[5] and quality of software title j, and J^k is the number of software titles available for hardware k. This specification along with $\alpha \geq 1$ and $\beta > 1$ ensures increasing and concave preferences in J^k.[6] New consumers face the budget constraint $y - p_h^k = x_0 + \sum_{j=1}^{J^k} p_{s,j}^k x_j^k$ where y is income, p_h^k is the price of hardware k, and $p_{s,j}^k$ is the price of software title j compatible with hardware k. Installed base consumers

---

[4] CES preferences are often used to model network effects (Chou and Shy, 1990; Church and Gandal, 1992; Park, 2002; Nair et al., 2004; Clements and Ohashi, 2005).

[5] The model simplifies the analysis by considering quantity a continuous variable. This approach is used throughout the literature on network effects because it yields a tractable model capturing consumer preferences over software variety (see cites in the above footnote as well as Spence (1976) and Dixit and Stiglitz(1977)). Another approach is model quantity choices discretely. Consumers who own console k would decide whether or not to buy each unit of compatible software on an individual basis. The discrete choice demand specification would have $\sum_{r=1}^{J^k} \frac{J^k!}{r!(J^k - r)!}$ (Wackerly et al., 1996) software consumption choices, where J^k is the number of compatible units of software, as the consumer considers all possible combinations of software. Anderson et al. (1992) approaches this formulation by assuming consumers purchase a variable amount of a single variant of software.

[6] This specification is analogous to Nair et al. (2004). Clements and Ohashi (2005) also use this specification to derive econometric estimates of the elasticity of hardware adoption with respect to software variety. A similar transformation is employed here as Clements and Ohashi's estimates are used for numerical simulations below.

face the same budget constraint except p_h^k is not included. Performing the same derivations below without p_h^k in the budget constraint yields the same result. Therefore, software demand is the same for both new consumers, N_1^k, and installed base consumers, N_0^k. The total number of consumers of software title j compatible with hardware k is $N_0^k + N_1^k = N^k$.

Demand for software title j is found using a two-stage budgeting process as follows: first consumers maximize by allocating income between the numeraire good and a quantity index for software compatible with hardware k, and then consumers choose the optimal quantity for each software title, x_j^k. Analysis begins in the second stage where the goal is to find x_j^k as a function of the quantity index.

In the second stage of the two-stage budgeting process new consumer i chooses x_j^k to solve the following problem:

$$\max_{x_j^k} U_{i,J^k}^k = x_0 + \left(\sum_{j=1}^{J^k} \left(x_j^k \theta_j^k \right)^{\frac{1}{\beta}} \right)^{\frac{1}{\alpha}} = y - p_h^k - \sum_{j=1}^{J^k} p_{s,j}^k x_j^k + \left(zed^k \right)^{\frac{1}{\alpha\beta}} \text{ s.t.}$$

$$\left(zed^k \right)^{\frac{1}{\beta}} = \left(\sum_{j=1}^{J^k} \left(x_j^k \theta_j^k \right)^{\frac{1}{\beta}} \right)$$

where the quantity index, zed^k, is given.

Define the Lagrangian

$$\mathsf{L} = y - p_h^k - \sum_{j=1}^{J^k} p_{s,j}^k x_j^k + \left(zed^k \right)^{\frac{1}{\alpha\beta}} - \lambda \left(\left(zed^k \right)^{\frac{1}{\beta}} - \left(\sum_{j=1}^{J^k} \left(x_j^k \theta_j^k \right)^{\frac{1}{\beta}} \right) \right).$$

The first order conditions $\frac{\partial L}{\partial x_j^k} = -p_{s,j}^k + \frac{1}{\beta}\lambda\left(\theta_j^k\right)^{\frac{1}{\beta}}\left(x_j^k\right)^{\frac{1}{\beta}-1} = 0$ imply $p_{s,j}^k x_j^k = \frac{1}{\beta}\lambda\left(x_j^k \theta_j^k\right)^{\frac{1}{\beta}}$.

Summing the latter and substituting for the multiplier in the first order condition yields a result that can be written in two useful ways:

$$x_j^k = \left(p_{s,j}^k\right)^{\frac{\beta}{1-\beta}} \left(\theta_j^k\right)^{\frac{1}{\beta-1}} \left(zed^k\right)^{\frac{1}{1-\beta}} \left(\sum_{j=1}^{J^k} p_{s,j}^k x_j^k\right)^{\frac{\beta}{\beta-1}} \tag{3}$$

and

$$\theta_j^k = \left(p_{s,j}^k\right)^{\beta} \left(x_j^k\right)^{\beta-1} \left(\sum_{j=1}^{J^k} p_{s,j}^k x_j^k\right)^{-\beta} zed^k . \tag{4}$$

Another valuable relationship is found by multiplying (3) by $p_{s,j}^k$ and rearranging to yield

$$\left(\sum_{j=1}^{J^k}\left(\frac{\theta_j^k}{p_{s,j}^k}\right)^{\frac{1}{\beta-1}}\right)^{\beta-1} \sum_{j=1}^{J^k} p_{s,j}^k x_j^k = zed^k . \tag{5}$$

where we define $\left(\sum_{j=1}^{J^k}\left(\frac{\theta_j^k}{p_{s,j}^k}\right)^{\frac{1}{\beta-1}}\right)^{\beta-1}$ as the quality per dollar index.[7] Equation (5) has an intuitively appealing interpretation: the quality per dollar index multiplied by expenditure on software equals (a monotonic transformation of) the benefit consumers receive from software (i.e. the quantity index).

We can analyze the first stage of the budgeting process by using (5) to express expenditure on software as a function of zed^k. The consumer's maximization problem is then

[7] Inverting the quality per dollar index and normalizing the quality of all software to unity gives the price index introduced by Dixit and Stiglitz (1977) and used extensively in the network effects literature (see, for example, Church and Gandal (1992, 1993), Park (2002), Nair et al. (2004), and Clements and Ohashi (2005)).

$$\max_{zed^k} U^k_{i,J^k} = y - p^k_h - \sum_{j=1}^{J^k} p^k_{s,j} x^k_j + \left(zed^k\right)^{\frac{1}{\alpha\beta}}$$

$$= y - p^k_h - \frac{zed^k}{\left(\sum_{j=1}^{J^k}\left(\frac{\theta^k_j}{p^k_{s,j}}\right)^{\frac{1}{\beta-1}}\right)^{\beta-1}} + \left(zed^k\right)^{\frac{1}{\alpha\beta}} \tag{6}$$

where the solution to the first order condition is

$$zed^k = \left(\alpha\beta\right)^{\frac{-\alpha\beta}{\alpha\beta-1}}\left(\sum_{j=1}^{J^k}\left(\frac{\theta^k_j}{p^k_{s,j}}\right)^{\frac{1}{\beta-1}}\right)^{(\beta-1)\frac{\alpha\beta}{\alpha\beta-1}}. \tag{7}$$

The next step is to find equilibrium software price. For the firm's maximization problem, we use (5) and (7) to express software quantity as a function of price and quality (see Appendix A for details). Assuming the quality per dollar index is not significantly affected by the change in the quality per dollar of a single software title,[8] a constant per unit marginal cost, c_s, and a per-unit licensing fee (paid to the hardware firm by the software firm), l^k, it is a straightforward exercise to show software price is

$$p^k_s = \beta\left(c_s + l^k\right). \tag{8}$$

To find new consumer i's utility for hardware system k, substitute (7) into (6), and use (8) for software price to obtain

$$U^k_{i,J^k} = y - p^k_h + \beta(\alpha\beta - 1)\left(\alpha\beta^2\right)^{\frac{-\alpha\beta}{\alpha\beta-1}}\left(c_s + l^k\right)^{\frac{-1}{\alpha\beta-1}}\left(\sum_{j=1}^{J^k}\left(\theta^k_j\right)^{\frac{1}{\beta-1}}\right)^{(\beta-1)\frac{1}{\alpha\beta-1}}. \tag{9}$$

[8] This is similar to assuming the price index for software (Dixit and Stiglitz, 1977) is not significantly affected by the change in price of a single software title. For analytical tractability, it is usually assumed the number of software titles is so large that a change in the price of a single software title will only have a negligible affect on the price index (Park, 2002; Clements and Ohashi, 2005).

Equations (9) and (1) provide the theoretical foundation for the empirical work to follow. We define the quality index for software compatible with hardware system k as

$\Theta^k = \left(\sum_{j=1}^{J^k} \left(\theta_j^k \right)^{\frac{1}{\beta-1}} \right)^{\beta-1}$. Note the far right term in (9) depends on the quality index and constants.

In Section 4 we estimate market share as a function of the quality index.

It is necessary to obtain a measure of software quality in order to calculate the quality index. As such, a major goal of the paper is to empirically determine software quality given information of software prices and sales. To this end, we use (5) to substitute for the quality per dollar index in (7) in order to obtain $zed^k = \left(\alpha\beta\right)^{\alpha\beta} \left(\sum_{j=1}^{J^k} p_{s,j}^k x_j^k \right)^{\alpha\beta}$. This can be used with (3) to find software quality as a function prices and quantities:

$$\theta_i^k = \left(\alpha\beta\right)^{\alpha\beta} \left(p_{s,i}^k\right)^{\beta} \left(x_i^k\right)^{\beta-1} \left(\sum_{j=1}^{J^k} p_{s,j}^k x_j^k \right)^{(\alpha-1)\beta}. \qquad (10)$$

The goal of the next section is to estimate (10) for each game.

3. Software Quality and Killer Applications

We will describe the data next, then use it with equation (10) to find software quality, after that we will continue with a discussion of the relationship between our calculated quality and observed revenue, and conclude this section with a discussion of killer applications.

3.1 Data

This study employs a monthly data set covering January 1995 through October 2007 provided by The NPD Group, a marketing research firm, and includes console (hardware) and game (software) point of sale data from approximately 65% of U.S. game retailers. The data set contains observations on the quantity of consoles and software sold, average console price, average software price, and introduction and exit dates (where applicable). It covers 15 consoles with 957 console/month observations and 7761 units of software with 309909 software/month observations in total. Descriptive statistics of relevant variables are displayed in Table 1.

Table 1
Monthly Industry Descriptive Statistics. Nine Hundred Fifty Seven Observations.

	Mean	Std. Dev.	Minimum	Maximum
Console Sales (in 10,000s)	14.842	26.590	0.0001	268.629
In Market Share (excluding outside option)	0.161	0.180	6.89×10^{-7}	0.641
Share of Potential Market (including outside option)	0.005	0.009	2.16×10^{-8}	0.091
Inflation Corrected Average Console Price* (in $100s)	0.722	0.548	0.021	2.955
Installed Base (in 1,000,000s)	8.856	9.928	0.002	38.983
Console Age (Months)	57.813	41.609	0.000	243.000
Average Software Age (Months)	28.838	22.196	0.000	108.000
Number of Available Games	319.598	306.327	0.000	1408.000

* Corrected using Consumer Price Index for Urban Consumers (1980 – 1982 = $100)

Notice the two measures of market share: ‘in market share’ and ‘share of potential market.’ The former is the share of consumers who buy a particular console out of all consumers who purchase consoles in a particular month. The latter is the share of consumers who buy a particular console out of the potential market including consumers who choose not to buy consoles (the outside option). Definition of the outside option is discussed in Section 4.1.

3.2 Software Quality

Finding game qualities from (10) requires estimates of α and β. Fortunately, Clements and Ohashi (2005) provide guidance for obtaining these parameter values. Assuming symmetric games, they find, on average, a 1% increase in network size yields a 4.52% increase in the number of games. In the case of symmetric software (see the Appendix A) the elasticity of the number of games to network size is given by

$$\varepsilon_{J^k,N^k} = \frac{\partial J^k}{\partial N^k}\frac{N^k}{J^k} = \frac{\alpha\beta-1}{\beta(\alpha-1)} \qquad (11).$$

Since α and β can not be identified by (7) alone, we set $\alpha = 1.1$ and solve $\frac{1.1\beta-1}{\beta(1.1-1)} = 4.52$ for $\beta \approx 1.54$. It should be noted regression results presented below are robust to various specifications α and β using the same elasticity.

Using the software data we obtain quality estimates for 7761 software titles over 154 periods (months). The result is 309869 valid quality measures for every month a game received positive sales in the data set. Descriptive statistics are displayed in Table 2.

Table 2
Descriptive Statistics for Game Quality Estimates by Console.

Console	Mean	Std. Dev.	Min	Max	N
3DO	42106.21	92855.63	0.80	1027172.58	4714
Sega Dreamcast	89544.13	286853.55	0.13	5429072.31	11858
Nintendo Gamecube	142000.11	379652.76	11.21	10278821.45	23955
Sega Genesis	86778.25	293185.21	0.69	12566494.65	31246
Atari Jaguar	30539.84	56277.50	0.50	421991.27	1350
Super Nintendo	130093.97	393092.43	1.68	15336598.02	25203
Nintendo 64	395679.68	1093363.45	0.86	26659564.33	14384
Nintendo Entertainment System	14614.30	24725.60	1.98	293326.33	4259
PlayStation	138395.63	401655.77	0.04	11615541.47	79145
PlayStation 2	216382.93	635232.45	2.02	19046214.72	63662
PlayStation 3	807956.34	786715.43	99.01	6527343.29	468
Sega Saturn	80621.27	187380.39	1.08	4139630.25	10074
Nintendo Wii	717903.49	891127.70	137.01	10476644.17	740
Microsoft Xbox	144415.21	426606.96	1.93	16529942.38	36675
Microsoft Xbox 360	898008.55	1515110.86	6151.27	26407274.38	2136
All Consoles	161719.60	516444.09	0.04	26659564.33	309869

Note we allow software quality to vary month to month as new prices and quantities are used to compute software quality each period. Another approach would be to estimate software quality given average price and total quantity for all periods in the data set. This would produce constant software quality. However, we felt allowing software quality to change over time more accurately reflects consumer perception (i.e. consumer perception of software quality likely changes over time as older titles become less attractive). It should be noted, however, regression results below are robust to constant software quality.

3.3 Quality, Observed Revenue, and a Definition for Killer Applications

Before defining killer applications per se, it is useful to look at the distribution of software quality and software revenue. Figure 1 and Figure 2 show software revenue and software quality histograms, respectively, for each game/month observation in the data set.

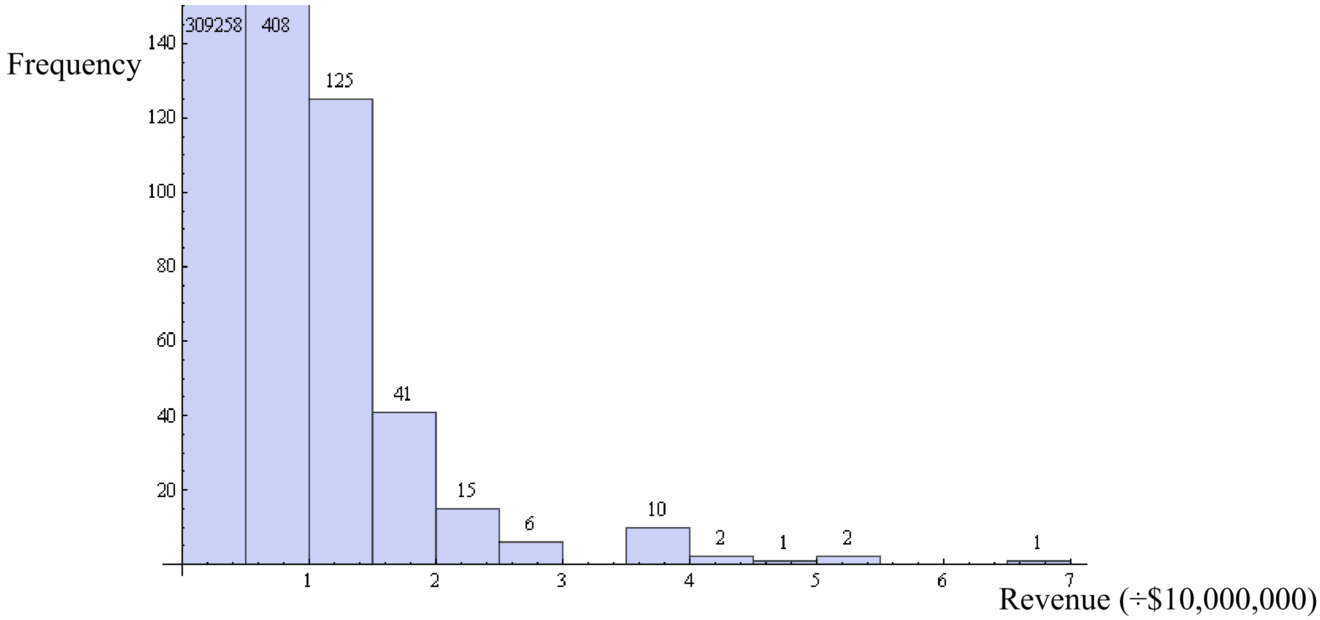

Figure 1. Software Revenue Histogram

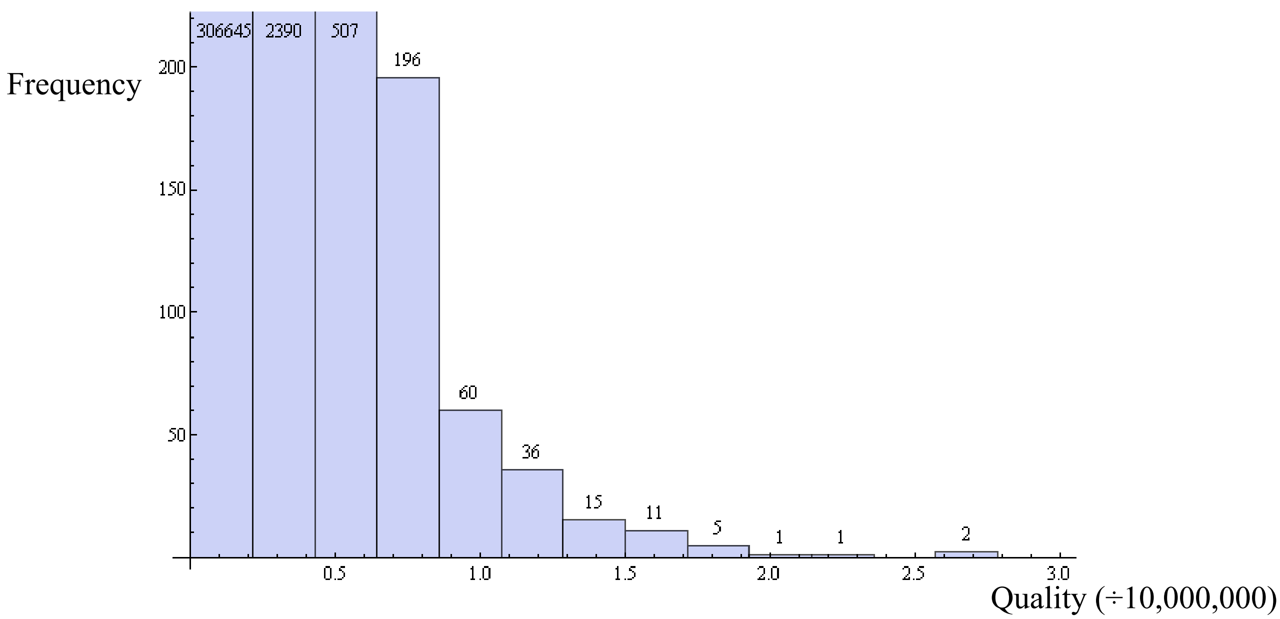

Figure 2. Software Quality Histogram

Frequency numbers are displayed on top (or inside but close to the top) of bars in the histogram figures. It is important to note we do not display all of the left most bars (the most frequently occurring observations) because they would dominate the picture and not reveal much distributional information. In fact, 99.8% (309258 out of 309869) and 99.0% (306654 out of

309869) of the revenue and quality observations, respectively, fall into the lowest range. Clearly, many games produce a small amount of revenue or are of low quality, while very few games generate high revenues or are high quality.

But the correlation between our quality measure and revenue is 0.864 for the data set. Given the high correlation between quality and revenue, intuitively one would expect a killer application to be of high quality and generate large amounts of revenue. We have chosen to define killer applications by revenue earned, both for its intuitive simplicity and because it does not require making the number or proportion of games which are killer applications predetermined. To define killer applications, we rank each game by the amount of revenue generated over the lifetime of the data set. We let killer applications be the collection of top games whose combined revenue account for 50% of total revenue generated by all games. Using this criteria, we find the top 6.2% of games (483 out of 7761) are killer applications and account for $13,309,818,259.47 of $26,617,987,346.69 total game revenue (approximately 50%).[9]

Table 3 displays a console's generational market share, number of killer applications, number of games, and the percentage of killer applications available out of all games for a console. Consoles are grouped into product generations where consoles of similar technological capabilities compete.

---

[9] Game revenue corrected for inflation using Consumer Price Index for Urban Consumers (1980 – 1982 = $100). It should be noted our results presented below our robust to narrower definitions of killer applications (e.g. the top 5% of revenue generating games, the top 2.5% of revenue generating games).

Table 3
Number of Killer Applications Per Console. Consoles Separated By Generation.*

Console	Year Introduced	Generational Market Share	Number of Killer Applications	Number of Games	Percentage of Killer Applications
Generation 1					
Sega Genesis	1989	48.62%	8	186	4.30%
Super Nintendo	1991	51.38%	10	182	5.49%
Generation 2					
3DO	1993	0.66%	0	77	0.00%
Atari Jaguar	1993	0.26%	0	30	0.00%
PlayStation	1995	60.34%	108	1353	7.98%
Sega Saturn	1995	2.77%	0	251	0.00%
Nintendo 64	1996	35.98%	68	290	23.45%
Generation 3					
Sega Dreamcast	1999	5.82%	8	250	3.20%
PlayStation 2	2000	56.59%	171	1638	10.44%
Nintendo Gamecube	2001	16.83%	30	574	5.23%
Microsoft Xbox	2001	20.76%	39	911	4.28%
Generation 4					
Microsoft Xbox 360	2005	50.31%	28	211	13.27%
Nintendo Wii	2006	35.63%	5	131	3.82%
PlayStation 3	2006	14.06%	1	80	1.25%

* Includes games released in January 1995 or later.

Table 3 shows the most successful systems (those with greater generational market share) generally have a larger absolute number of killer applications as well as a greater percentage of killer applications available out of all games for a console. Notice the relationship between the percentage of killer applications available out of all games for a console and market leadership. Super Nintendo, PlayStation 2, and Microsoft Xbox 360 all have the highest percentage of killer applications available and are the dominant consoles in their respective generations. (The exception to this, Nintendo 64, is discussed in Section 6.)

On the other end of the spectrum, 3DO, Atari Jaguar, Sega Saturn, and Sega Dreamcast all have a low percentage of killer applications along with very low generational market shares.

Highlighting the importance of killer applications, it is interesting to note Sega Saturn and Sega Dreamcast each have a significant number of games available (250 and 251, respectively) but are still not among the successful consoles within their generation.

While this analysis is cursory, it supports the argument that the number of games may not be the best predictor of console market shares. In the next section we estimate console market share based on the quality index for software which gives greater weight to higher quality games (those that might be defined as killer applications).

4. Hardware Market Share Estimations

The goal of this section is to estimate the market share for consoles, paying particular attention to quality index, killer application, and non-killer application (i.e. all other games) elasticities. Our main empirical model involves the quality index for software and is discussed in detail below. We estimate the same basic model but include the number of games, separated into killer applications and non-killer applications, in place of the quality index for software as a comparison. We refer to the former as the quality index specification and the latter as the number of games specification.

4.1 Empirical model

Employing the technique for estimating differentiated product discrete-choice demand models developed by Berry (1994) and Berry et al. (1995), we define the benefit consumer i receives from console k out of $k+1$ options at time t as

$$U_{i,t}^{k} = \delta_z z_t^k - \delta_p p_{h,t}^k + \delta_\Theta \Theta_t^k + \xi_t^k + \varepsilon_{i,t}^k \qquad (12)$$

where z_t^k is a vector of console specific attributes at time t, δ_z is a vector of consumer i's preference parameters for console attributes, $p_{h,t}^k$ is the price of console (hardware) k at time t, δ_p is consumer i's marginal (dis-) utility of price, Θ_t^k is the quality index for software compatible with console k at time t [10], δ_Θ is consumer i's marginal utility from the quality index for software, ξ_t^k are unobserved characteristics of hardware k at time t, and $\varepsilon_{i,t}^k$ is an individual specific mean-zero shock.

As is standard in these types of models (Berry, 1994; Berry et al., 1995; Nevo, 2000), we let consumers consider $k+1$ options because we include the outside option of not purchasing a console. Following Clements and Ohashi (2005), the market for video game consoles is defined as U.S. households with a television set; data is given by The Nielson Company (http://en-us.nielsen.com/).[11] To calculate the potential market in each period (month) we first find each console's installed base — the total number system purchases before a particular period. The potential market in each period is then the sum of each console's installed base subtracted from the number of U.S. households with a television set. Given this, we define a console's market share of new consumers as the number of new adopters relative to the potential market; the number of consumers who choose the outside option is the sum of all consoles' new adopters subtracted from the potential market.

[10] Initially we included two quality indexes for software in the estimations; one for software designed for the console and another for software designed for a previous generation console. Sony PlayStation 2, Sony PlayStation 3, Microsoft Xbox 360, and Nintendo Wii are 'backward compatible' in that they can play games designed for a previous generation system. We separated the indexes because we suspected an increase in the quality index for software designed for a previous generation console will have a different (smaller) effect. However, empirical evidence suggested this was not the case (the null hypothesis of equal betas was never rejected). As such, we combined the quality indexes. Also, we took the same approach for the number of games specification.

[11] Dubé et al. (*in press*) use the total number of U.S. households as the potential market in their empirical study of the 32/64-bit generation of the U.S. home video game industry; our empirical results hold when we use this definition.

It should be noted we accommodate the problem of multiple purchases by the same household (or upgrading to a newer console) by letting each console's installed base depreciate at an annual rate of 90% per year.[12] Essentially, the assumption is a single household will not purchase multiple consoles in the same period, but some fraction will purchase an additional console in a latter period.

The differences between (12) and the theoretically derived equivalent in (9) should be discussed. First, (12) is independent of income; this is an innocuous assumption since the income effect would enter each alternative for the consumer (hence, canceling out). Second, $\delta_z z_t^k$ is a console specific effect independent of the quality index for software. Inclusion of this term deviates from the mathematical model in that consumers may receive benefit from a console independent of software. This is to rationalize the inclusion of a console specific fixed effect in the estimation (see Nevo (2000) for several arguments in favor of including fixed effects in these types of estimations). Including fixed effects will capture the effect of many observable console characteristics since console performance specifications do not change over a consoles life. Third, $\delta_p p_{h,t}^k$ is simply the price effect without the normalization that the marginal effect of price is 1 (as in equation (9)). Fourth, for $\delta_\Theta \Theta_t^k$, notice the constants $\beta(\alpha\beta-1)\left(\alpha\beta^2\right)^{\frac{-\alpha\beta}{\alpha\beta-1}}\left(c_s+l^k\right)^{\frac{-1}{\alpha\beta-1}}$ are captured in δ_Θ. Another assumption we make is the quality index for software enters consumer utility in a linear fashion. This differs from equation (9) however empirical evidence suggests we cannot reject the hypothesis of a linear relationship.[13]

---

[12] The econometric results below are robust to various depreciation rates.

[13] Performing the P_E test suggested by Davidson and MacKinnon (1981) yields a *t*-stat = 1.286 which fails to reject the null hypothesis of a linear relationship.

Including unobserved console characteristics, ξ_t^k, is consistent with other empirical studies employing differentiated product discrete-choice demand models (see Berry(1994), Berry et al. (1995), Nevo (2000), and Clements and Ohashi (2005)). We assume there are some product attributes observed by consumers and producers but not observed by the econometrician. While including the fixed effect, $\delta_z z_t^k$, will capture the effect of unobserved characteristics that are constant over time, other effects such as advertising or brand image may change over a console's life.

We impose standard assumptions on $\varepsilon_{i,t}^k$ to generate a nested logit estimation. The nests are as follows: first consumers decide between purchasing a console and not purchasing (choosing the outside option), second consumers decide which console to purchase (if they choose to purchase in the first stage). Following Berry (1994) and Clements and Ohashi (2005), the regression model given these assumptions is

$$\ln\left(s_t^k\right) - \ln\left(s_t^0\right) = \delta_0 + \delta_z z_t^k - \delta_p p_{h,t}^k + \delta_\Theta \Theta_t^k + \sigma \ln\left(\frac{s_t^k}{\sum_k s_t^k}\right) + \xi_t^k \tag{13}$$

where s_t^k is console k's market share of new consumers in period t, s_t^0 is the outside option share in period t, and $\ln\left(\frac{s_t^k}{\sum_k s_t^k}\right)$ is console k's in market share (as defined in Section 3.1).

Finally, though not displayed in equation (13), we include month and year dummies in the estimation. The year dummies can capture the effect of any industry wide dynamics while the month dummies will control for any systematic trends (e.g. the holiday season).

Next we discuss potential endogeneity problems and instruments. Then we conclude this section by reviewing the estimation results.

4.2 Instruments

This paper employs instrumental variables to correct for possible endogeneity in the main explanatory variables: average console price, natural log of in market share, and the quality index for software. Average console price is likely correlated with the regression error term, ξ_t^k, because unobserved (to the econometrician) console characteristics maybe be taken into account when the hardware firm sets profit maximizing price. Correlation between the error term and natural log of in market share is likely because the latter contains part of the dependent variable. Finally, the quality index for software will increase with the number of games available since the quality of each game is, among other things, summed to determine the value of the index. The number of games maybe influenced by the market share of new consumers; Clements and Ohashi (2005) suggest this is very likely if error terms are autocorrelated.[14]

The challenge is finding instruments that are correlated with the endogenous regressors and uncorrelated with the residuals. Nevo (2000) suggests using cost side instruments when they are available. As such, following Dube et al. (*in press*) we use the producer price indexes for Electronic Computer Manufacturing, Computer Storage Device Manufacturing, and Audio and Video Equipment Manufacturing to control for endogeneity in average console price.[15] Unfortunately, the producer price indexes do not vary over consoles; they only identify changes in hardware demand (not specific console demand). As a remedy, also include console age (number of months since console introduction) as an instrument. The key assumption is manufacturing costs will be decline over the product lifecycle; we find this likely given evidence presented in Coughlan (2001). Finally, we interact the four instruments with the number of

[14] We expect endogeneity to arise in the number of games specification for similar reasons.
[15] Data on producer price indexes are obtained from the Bureau of Labor Statistics (http://www.bls.gov/ppi/)

competitors a console has in any period (i.e., the number of other consoles in the market). A greater number of competitors should be correlated with lower console prices. Interacting the variables is done to prevent the instruments from capturing a simple time trend.

Average software age is used as an instrument for the natural log of in market share. Following Clements and Ohashi (2005), we expect consoles with older games on average to have a larger in market share. The logic is as follows: popular games will attract more consumers to a console (yielding a greater in market share), and games are likely to survive longer only if they are popular. We interact this variable with console age because we suspect popular games on newer consoles will have a different effect than popular games on older consoles.

Finally, we use producer price indexes for Game Software Publishing and Magnetic and Optical Recording Media Manufacturing as cost side instruments for the quality indexes for software.[16] Again, because these instruments do not vary over consoles and only capture industry wide cost shocks, we interact them with average software age and the number of competitors.[17]

4.3 Results

Descriptive statistics and estimations are presented in Table 4 and Table 5, respectively. In Table 5, estimations (1), (3), and (5) employ the quality index specification while estimations (2), (4), and (6) use the number of games specification. Please see Table B in Appendix B for summary statistics on console, year, and month dummies.

[16] It should be noted the producer price index for Game Software Publishing begins in December, 1997. Year dummies account for dates when the index was not available; we let the value of the variable be zero in periods it was not observed.

[17] We use the same instruments in the number of games specification.

Table 4
Descriptive Statistics for Hardware Regressions. Nine Hundred Fifty Seven Observations.

	Mean	Std. Dev.	Minimum	Maximum
Dependent Variable:				
$\ln\left(s_t^k\right) - \ln\left(s_t^0\right)$	-7.773	3.503	-17.638	-2.225
Endogenous Regressors:				
Inflation Corrected Average Console Price	0.722	0.548	0.021	2.955
Quality Index for Software	0.761	0.965	0.000	6.942
Number of Non-Killer Applications*	3.718	3.683	0.000	15.730
Number of Killer Applications*	0.400	0.558	0.000	2.400
$\ln\left(\text{In Market Share}\right)$	-3.902	3.450	-14.188	-0.444
Instruments:*				
Average Software Age × Magnetic and Optical Recording Media Manufacturing Producer Price Index	21.624	16.683	0.000	75.000
Average Software Age × Game Software Publishing Producer Price Index	14.246	17.560	0.000	94.594
Number of Competitors × Magnetic and Optical Recording Media Manufacturing Producer Price Index	4.078	0.925	2.055	5.642
Number of Competitors × Game Software Publishing Producer Price Index	23.014	18.495	0.000	70.000
Number of Competitors × Electronic Computer Manufacturing Producer Price Index	12.623	8.558	2.667	29.040
Number of Competitors × Computer Storage Device Manufacturing Producer Price Index	10.323	5.510	2.871	20.112
Number of Competitors × Audio and Video Equipment Manufacturing Producer Price Index	4.092	0.948	2.043	5.761
Number of Competitors × Console Age	3.098	2.390	0.000	14.520

* Values divided by 100 for presentation purposes.

Table 5
Hardware Adoption Estimations: Natural Log of Share of the Potential Market Subtracted from Natural Log of Outside Market Share on Console Price, Quality Index, Non-Killer Applications, Killer Applications, and In Market Share.

	OLS Estimations		2SLS Estimations		GMM Estimations	
	(1)	(2)	(3)	(4)	(5)	(6)
Inflation Corrected Average Console Price	-0.287*** (0.064)	-0.404*** (0.073)	-0.463 (0.404)	-2.036** (0.993)	-0.478* (0.359)	-2.595*** (0.755)
Quality Index for Software	0.285*** (0.027)		1.811*** (0.259)		1.715*** (0.241)	
Number of Non-Killer Applications		-0.060** (0.017)		-0.429 (0.366)		-0.465* (0.291)
Number of Killer Applications		0.796*** (0.172)		12.915*** (4.518)		11.848*** (3.405)
$\ln(\text{In Market Share})$	0.954*** (0.008)	0.935*** (0.010)	0.863*** (0.039)	0.526*** (0.150)	0.866*** (0.037)	0.569*** (0.121)
Number of Observations	957	957	957	957	957	957
R^2	0.986	0.985				
J Statistic			3.732	4.700	2.980	3.103
1-stg F-Stats:						
Inflation Corrected Average Console Price			10.810***	10.810***		
Quality Index for Software			7.145***			
Number of Non-Killer Applications				11.331***		
Number of Killer Applications				7.024***		
$\ln(\text{In Market Share})$			31.280***	31.280***		

*, **, and *** indicates significance at the 10%, 5%, and 1% level, respectively.
(Heteroskedastic consistent standard errors are presented in parenthesis.)
The dependent variable is the natural log of share of potential market subtracted from the natural log of outside market share.
A constant as well as console, year, and month dummies are included in estimations but not displayed for brevity.
Instruments for 2SLS and GMM estimation are listed in Table 4.

Table 5 displays OLS (Ordinary Least Squares), 2SLS (2-Stage Least Squares), and GMM (Generalized Method of Moments) estimates of the parameters. As expressed above, OLS will be biased if our regressors are endogenous. A cursory look at the parameter estimates suggests this is indeed the case. A test of endogeneity confirms our suspicions; the Hausman (1978) statstic is significant in both specifications.

Diagnostics from Table 5 show the instruments are not weak; the first stage F-Statistics are all significant at the 99% level. Further, the J Statistic fails to reject the overidentifying restrictions; the instruments are orthogonal to the errors. As such, we will use GMM estimates (estimations (5) and (6) in Table 5) for our analysis below. We prefer GMM to 2SLS because it is more efficient in the presence of heteroskedasticity. It should be noted that GMM may perform worse than 2SLS if errors are homoskedastic and the sample size is small. However, this is not the case here.

A number of parameter estimates conform to our priors. Price has a negative and significant effect while the quality index for software, from (5), and the number of killer applications, from (6), have a positive and significant effect. However, the number of non-killer applications, from (6), is significant and has the wrong sign. Our expectation is the number non-killer applications should have a smaller effect than the number of killer applications, but it should not have a negative impact on market share. While estimations (5) and (6) are statistically similar,[18] use of the quality index ensures additional games (either killer application or not) positively affect market share.

[18] We fail to reject the null hypothesis of asymptotically equivalent lack-of-fit criteria (Rivers and Vuong, 2002).

The elasticity of market share with respect to the quality index for software and number of killer applications is 8.41% and 9.95%, respectively, at average variable values.[19] However, comparison is not straight forward since the quality index is not directly related to the number of games. In order to facilitate direct comparison, we estimate the elasticity of the quality index for software to the number of killer applications and number of non-killer applications in Appendix C. Our results suggest a 1% increase in killer applications will increase the quality index for software by 1.55%. Therefore, a similar increase in killer applications will, on average, yield a 13.04% (= 1.55 × 8.41) increase in market share according to the quality index specification. Comparing estimations (5) and (6), the quality index specification suggests a roughly 31% (13.04 to 9.95) greater impact from killer applications.

Using the estimation of the elasticity of the quality index for software to the number of non-killer applications from Appendix C, a 1% increase in non-killer applications will cause a 0.40% (= 0.047 × 8.41) increase in market share. However, it should be noted results from Appendix C show the number of non-killer applications does not significantly effect on the quality index. We expect either a small positive or insignificant effect is more reasonable than the -2.22% (significant) elasticity of market share to non-killer applications (calculated at the average values) given in the number of games specification.

In either specification, the influence of killer applications on market share is paramount. Given this, it is important to determine if the network effect is different for killer applications and non-killer applications.

---

[19] Quality index elasticity is given by $\left(1-\sigma\frac{s^k}{\sum_k s^k}-(1-\sigma)s^k\right)\frac{\delta_\Theta \Theta_t^k}{1-\sigma}$. Killer applications elasticity and non-killer application elasticity is calculated in a similar fashion.

5. Killer Applications and the Network Effect

The goal of this section is to determine whether the network effect for killer applications is stronger or weaker than the network effect for non-killer applications. Since killer applications have a much larger influence on market share than non-killer applications, previous network effect estimates based on number of available games may be overstated (if the network effect is less for killer applications) or understated (if the opposite is true).

We perform two reduced form estimations where the dependent variables are (1) the natural log of the number of killer applications and (2) the natural log of the number of non-killer applications. We use KA and NKA, respectively, to distinguish between the estimations. In each instance, the independent variable is the natural log of the installed base. The installed base is the total number of console adopters prior to the current period. We account for backward compatibility in the installed base by summing where applicable. For example, PlayStation 3 can play PlayStation 2 games: the installed base for PlayStation 2 games is found by summing the installed base for PlayStation 3 and PlayStation 2.

As Clements and Ohashi (2005) note, an unobserved shock in the software market in the previous period can produce an increase in the current periods installed base. As such, some cost side instruments described in Section 4.2 are employed to control for potential endogeneity. The instruments used are given in Table 6 along with the descriptive statistics. The estimation results are displayed in Table 7. Note the number of observations is different in the KA and NKA estimations because the log-log specifications force us to eliminate observations where the number of killer applications or the number of non-killer applications is zero.

Table 6
Descriptive Statistics for Network Regressions.

	N	Mean	Std. Dev.	Minimum	Maximum
Dependent Variables:					
ln (Number of Killer Applications)	744	3.062	1.119	0.000	5.130
ln (Number of Non-Killer Applications)	943	5.128	1.220	0.000	7.124
Endogenous Regressor:					
ln (Installed Base)	947	15.831	1.695	10.418	18.076
Instruments:*					
Number of Competitors × Magnetic and Optical Recording Media Manufacturing Producer Price Index	947	4.076	0.927	2.055	5.642
Number of Competitors × Audio and Video Equipment Manufacturing Producer Price Index	947	4.090	0.950	2.043	5.761
Number of Competitors × Console Age	744	2.951	2.127	0.000	10.150
Average Software Age × Magnetic and Optical Recording Media Manufacturing Producer Price Index	947	21.852	16.622	0.000	75.000

* Values divided by 100 for presentation purposes.
Both specifications use first two instruments listed.
Number of Competitors × Console Age is only used as an instrument when ln (Number of Killer Applications) is the dependent variable.
Average Software Age × Magnetic and Optical Recording Media Manufacturing Producer Price Index is only used as an instrument when ln (Number of Non-Killer Applications) is the dependent variable.

Table 7
Network Effect Estimations: Natural Log of the Number of Killer Applications and Natural Log of the Number of Non-Killer Applications Regressed on the Natural Log of Installed Base.

	OLS Estimations		2SLS Estimations		GMM Estimations	
	KA	NKA	KA	NKA	KA	NKA
ln (Installed Base)	0.879* (0.046)	1.498* (0.064)	0.658* (0.187)	2.156* (0.149)	0.800* (0.159)	2.206* (0.146)
Number of Observations	744	943	744	943	744	943
R^2	0.870	0.802				
J Statistic			4.403	4.230	2.084	3.379
1-stg F-Stats:						
ln (Installed Base)			15.124*	59.716*		

* indicates significance at the 1% level.
(Heteroskedastic consistent standard errors are presented in parenthesis.)
KA indicates the dependent variable is the natural log of the number of killer applications.
NKA indicates the dependent variable is the natural log of the number of non-killer applications.
A constant as well as console, year, and month dummies are included in estimations but not displayed for brevity.
Instruments for 2SLS and GMM estimation are listed in Table 6.

The GMM estimations displayed in Table 7 will be used for the analysis below. We avoid the OLS estimations because our endogeneity concerns seem well founded; the Hausman (1978) statistic is significant in every case. Also, the instruments are not weak given the J Statistics and the first stage F-Statistics.

Results from the KA and NKA estimations in Table 7 suggest the network effect is significantly (with 99% confidence) larger for non-killer applications than killer applications. Specifically, a 1% increase in the installed base will increase the number of non-killer applications by 2.2% compared to a 0.8% increase in killer applications.[20] In other words, the network effect is 2.75 (= 2.2 / 0.8) times greater for non-killer applications. We consider this

[20] It should be noted the network effect is also captured in the quality index specification. Using similar instruments and explanatory variables with the natural log of the quality index for software as the dependent variable produces a statistically significant quality index for software elasticity to installed base of 3.719%.

result, and the importance of killer applications highlighted in Section 4, while discussing industry market share dynamics below.

6. Network Effects and Killer Applications

Recall Table 3 in Section 3.3 showed the most successful systems (those with greater generational market share) generally have a larger absolute number of killer applications as well as a greater percentage of killer applications relative to the total number of games available for the console. The objective of this section is to examine the dynamics of generational market share over the life of a console and relate it to the total number of games and percentage of killer applications available. We do this in a stylized manner by looking at several descriptive tables.

Tables 8a – 8c display the average number of games available, the average percentage of killer applications (out of games available for the console), and the generational total market share (percent of installed base), respectively, for consoles of interest in each year.

Table 8a
Yearly Average of Total Number of Games Available for Each Console.*

	Year												
Console	95	96	97	98	99	00	01	02	03	04	05	06	07
Generation 1													
Sega Genesis	610	617	534	362	218	129	85	34	11	-	-	-	-
Super Nintendo	523	508	392	273	171	121	79	-	2	-	-	-	-
Generation 2													
3DO	98	140	125	28	-	-	-	-	-	-	-	-	-
Atari Jaguar	24	40	38	10	-	-	-	-	-	-	-	-	-
PlayStation	10	122	298	461	607	763	916	891	830	723	580	292	123
Sega Saturn	12	100	203	234	184	79	21	-	-	-	-	-	-
Nintendo 64	-	3	22	76	158	238	270	218	114	68	29	-	1
Generation 3													
Sega Dreamcast	-	-	-	-	11	108	224	247	202	135	-	-	-
PlayStation 2*	-	-	-	-	-	772	1034	1214	1400	1540	1640	1553	1498
Nintendo Gamecube	-	-	-	-	-	-	3	75	227	350	445	498	478
Microsoft Xbox	-	-	-	-	-	-	6	102	278	477	676	821	837
Generation 4													
Microsoft Xbox 360*	-	-	-	-	-	-	-	-	-	-	342	448	544
Nintendo Wii*	-	-	-	-	-	-	-	-	-	-	-	503	547
PlayStation 3*	-	-	-	-	-	-	-	-	-	-	-	1555	1542

* Backward compatible games included.

Table 8b
Yearly Average of Percent of Killer Applications Available out of Total Number of Games Available for Each Console.

	Year												
Console	95	96	97	98	99	00	01	02	03	04	05	06	07
Generation 1													
Sega Genesis	0.01	0.01	0.02	0.03	0.05	0.08	0.10	0.17	0.00	-	-	-	-
Super Nintendo	0.01	0.03	0.04	0.06	0.08	0.10	0.09	-	0.00	-	-	-	-
Generation 2													
3DO	0.00	0.00	0.00	0.00	-	-	-	-	-	-	-	-	-
Atari Jaguar	0.00	0.00	0.00	0.00	-	-	-	-	-	-	-	-	-
PlayStation	0.08	0.06	0.08	0.10	0.12	0.12	0.11	0.12	0.12	0.12	0.14	0.21	0.29
Sega Saturn	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-	-	-	-	-	-
Nintendo 64	-	0.97	0.68	0.36	0.27	0.24	0.24	0.27	0.35	0.40	0.49	-	1.00
Generation 3													
Sega Dreamcast	-	-	-	-	0.15	0.07	0.04	0.03	0.04	0.06	-	-	-
PlayStation 2*	-	-	-	-	-	0.12	0.11	0.12	0.12	0.13	0.13	0.14	0.13
Nintendo Gamecube	-	-	-	-	-	-	0.17	0.12	0.07	0.06	0.06	0.06	0.06
Microsoft Xbox	-	-	-	-	-	-	0.14	0.08	0.05	0.05	0.05	0.05	0.05
Generation 4													
Microsoft Xbox 360*	-	-	-	-	-	-	-	-	-	-	0.07	0.08	0.09
Nintendo Wii*	-	-	-	-	-	-	-	-	-	-	-	0.06	0.06
PlayStation 3*	-	-	-	-	-	-	-	-	-	-	-	0.14	0.13

* Backward compatible games included.

Table 8c
Percent of Installed Base in a Generation Per Console.*

	Year												
Console	95	96	97	98	99	00	01	02	03	04	05	06	07
Generation 1													
Sega Genesis	0.50	0.50	0.49	0.49	0.49	0.49	1.00	-	-	-	-	-	-
Super Nintendo	0.50	0.50	0.51	0.51	0.51	0.51	-	-	-	-	-	-	-
Generation 2													
3DO	0.32	0.07	0.03	-	-	-	-	-	-	-	-	-	-
Atari Jaguar	0.14	0.03	0.01	-	-	-	-	-	-	-	-	-	-
PlayStation	0.35	0.46	0.45	0.54	0.57	0.59	0.59	0.61	0.62	0.63	0.63	1.00	0.63
Sega Saturn	0.19	0.15	0.11	0.06	0.04	-	-	-	-	-	-	-	-
Nintendo 64	-	0.29	0.40	0.40	0.39	0.41	0.41	0.39	0.38	0.37	0.37	-	0.37
Generation 3													
Sega Dreamcast	-	-	-	-	1.00	0.80	0.37	0.17	0.11	0.08	-	-	-
PlayStation 2*	-	-	-	-	-	0.20	0.50	0.55	0.55	0.53	0.56	0.58	0.60
Nintendo Gamecube	-	-	-	-	-	-	0.06	0.12	0.15	0.17	0.19	0.18	0.18
Microsoft Xbox	-	-	-	-	-	-	0.07	0.15	0.18	0.22	0.25	0.24	0.22
Generation 4													
Microsoft Xbox 360*	-	-	-	-	-	-	-	-	-	-	1.00	0.84	0.51
Nintendo Wii*	-	-	-	-	-	-	-	-	-	-	-	0.12	0.34
PlayStation 3*	-	-	-	-	-	-	-	-	-	-	-	0.05	0.14

* December values are reported; only total adopters of consoles observed in December are used to calculate values.

Consider Generation 1 first. Though not displayed in the table, Super Nintendo entered the market in 1991, two years after Sega Genesis. Since its introduction, Super Nintendo has had fewer games available than Sega Genesis in every year. However, early on Super Nintendo had a larger percentage of killer applications. Our evidence suggests Sega Genesis' network advantage was weak because they had a larger portion of non-killer applications available than their main competitor. This can explain the slight advantage in generational market share Super Nintendo enjoyed from 1997 onward. In other words, an established console's leadership position was displayed by a late second mover with a smaller number of available games but a larger concentration on killer applications.

Evidence from Generation 2 suggests similar dynamics. 3DO and Atari Jaguar entered the market two years before PlayStation and Saturn did in 1995. Saturn did not fare well due to its lack of killer applications. PlayStation, however, was able to displace market leaders by offering a larger fraction of killer applications. 3DO and Atari Jaguar had no killer applications available though they did have a large selection games, the former offering 140 in 1996.

To emphasize the importance of killer applications to successful entry again, it is interesting to compare Nintendo 64 and PlayStation. It is important to note Nintendo 64 was the last console to enter in the generation, introduced a year after PlayStation which had quickly become the market leader. Nintendo 64 was at a severe network disadvantage if only looking at total available games. To the point, in 1997 PlayStation had 298 games available compared to Nintendo 64's 22. However, 68% of games for Nintendo 64 were killer applications; PlayStation only boasted 8%. While PlayStation remained the dominant console in the generation, Nintendo 64 was able to capture 37% of generational market share even though it was the last mover.

Evidence from Generations 3 and 4 reinforce findings from Generations 1 and 2. Consoles successfully enter a generation when they offer a higher percentage of killer applications than the incumbent. And generational leaders often have a higher percentage of killer applications than second and third place consoles.[21]

6. Conclusions

We find the availability of killer applications has a much larger effect on console adoption than non-killer applications. However, our results suggest network effects may be smaller for killer applications. As such, it seems possible that some of what has been attributed to networks in

[21] The exception to this is PlayStation 3 in generation 4. However, the majority of killer applications are designed for either PlayStation or PlayStation 2. That is, they are backward compatible games. Only 1.25% (1 out of 80) games designed for PlayStation 3 are killer applications.

previous studies might be attributable to software asymmetry, in particular, the highest revenue producing games.

One of the primary conclusions of the present paper is that software quality is an important determinant of the dynamics of hardware shares in the video game industry. A key contribution is the use of a quality index for software in estimating market share. Our formulation allows for better quality software to be given larger weight in the quality index. The use of a quality index approach has the added advantage of allowing different values for every game. As a result, we are able to estimate the different marginal effect of each game. Essentially, the marginal effect is the change in the predicted number of consumers for a console when a given game is removed. (Appendix D describes how to obtain the marginal effect of software on hardware in detail). Table 9 displays the marginal effects of the top 10 revenue generating games in the data set relative to the marginal effect of a median game released on the same console.

Table 9
Marginal Effect of Top 10 Revenue Generating Games (January 1995 – October 2007) Relative to the Marginal Effect of the Median Game for the Console.

Game	Total Revenue*	Intro Date	Console	Marginal Effect Relative to Median Game for the Console
Super Mario 64	$189,837,318.75	Sep-96	N64	62.530
Grand Theft Auto: San Andreas	$164,001,549.46	Oct-04	PS2	3.988
Grand Theft Auto: Vice City	$162,680,115.70	Oct-02	PS2	8.6253
Goldeneye 007	$152,989,440.10	Aug-97	N64	41.544
Mario Kart 64	$145,596,215.39	Feb-97	N64	32.944
Grand Theft Auto 3	$137,003,083.16	Oct-01	PS2	18.462
Zelda: Ocarina of Time	$117,465,388.51	Nov-98	N64	84.427
Halo 2	$95,813,337.55	Nov-04	Xbox	265.172
Guitar Hero 2 W/Guitar	$94,204,882.73	Nov-06	PS2	5.069
Halo	$93,102,199.88	Nov-01	Xbox	92.571
Average				61.533

* Corrected for inflation using Consumer Price Index for Urban Consumers (1980 – 1982 = $100)

Results from Table 9 show the "best games" have, on average, a roughly 62 times larger effect on console adoption than the median game. Although the ten best games are only a fraction of the killer applications, even this subset is clearly important in determining console market share and leadership. In fact, these games account for 5.08% ($1,352,693,531.23 out of $26,617,987,346.69) of game revenue in the sample. Coughlin (2001) finds the top 10 best-selling games in each year accounted for 1/3rd of yearly global revenue each by the late 1990's. The difference suggests turnover in the best selling games from year to year.

Another thing to notice from Table 9 is the differences in marginal effects. These differences would not be captured with a dummy variable approach where games are defined as killer applications according to subjective criteria. Further, the dummy variable approach may yield perverse results because differences within subjective categories are not captured. To the point, the criteria for killer applications used in this paper led to an unrealistic result in our hardware estimations: non-killer applications have a negative and significant effect on console market share. Use of the quality index avoids this problem since an additional game necessarily increases the index.

But it goes without saying that the video game industry is not the only one with both network effects and killer applications. Killer applications have been an important factor in many industries; Microsoft Office on computers and laptops, NFL Sunday Ticket on DirecTV, The Howard Stern Show on Sirius Satellite Radio, and the Apple iPhone on the AT&T network are just some examples. Perhaps future work might apply the methodology of the present paper to other industries.

Appendix A

A.1 The Software Firm's Problem

Software firm j 's profit function is given by

$$\Pi_s^j = \left(p_{s,j}^k - c_s - l^k\right) x_j^k N^k - f_s \text{ where}$$

$$x_j^k = \left(p_{s,j}^k\right)^{\frac{\beta}{1-\beta}} \left(\theta_j^k\right)^{\frac{1}{\beta-1}} \left(\alpha\beta\right)^{\frac{-\alpha\beta}{\alpha\beta-1}} \left(\sum_{j=1}^{J^k} \left(\frac{\theta_j^k}{p_{s,j}^k}\right)^{\frac{1}{\beta-1}}\right)^{-\frac{\beta(\alpha-1)}{\alpha\beta-1}} \quad \text{(A.1)}$$

is found by substituting for $\sum_{j=1}^{J^k} p_{s,j}^k x_j^k$ and zed^k from (5) and (7) into (3). The marginal and fixed costs of software are c_s and f_s, respectively, and l^k is a per-unit licensing fee paid to hardware firm k. Inclusion of the latter is for consistency with the industry; third party software developers typically pay a licensing fee, or royalty, to the hardware firm for each game sold. However, for simplicity we assume the licensing fee is constant.

Software price is found by maximizing Π_s^j with respect to $p_{s,j}^k$ and solving. Note, as stated in footnote 8, we assume $\frac{\partial}{\partial p_{s,j}^k}\left(\sum_{j=1}^{J^k}\left(\frac{\theta_j^k}{p_{s,j}^k}\right)^{\frac{1}{\beta-1}}\right) = 0$. The result is given by (8) in the text.

A.2 Elasticity of the Number of Games to Network Size

In the special case of symmetric games (A.1) simplifies to

$$x_j^k = \left(\alpha\beta\right)^{\frac{-\alpha\beta}{\alpha\beta-1}} \left(p_{s,j}^k\right)^{\frac{-\alpha\beta}{\alpha\beta-1}} \left(J^k\right)^{\frac{(1-\alpha)\beta}{\alpha\beta-1}} \left(\theta_j^k\right)^{\frac{1}{\alpha\beta-1}}$$

and the zero profit condition implies

$$\left(\beta-1\right)\left(c_s+l^k\right)\left(\left(\alpha\beta\right)^{\frac{-\alpha\beta}{\alpha\beta-1}}\left(\beta\left(c_s+l^k\right)\right)^{\frac{-\alpha\beta}{\alpha\beta-1}}\left(J^k\right)^{\frac{(1-\alpha)\beta}{\alpha\beta-1}}\left(\theta_j^k\right)^{\frac{1}{\alpha\beta-1}}\right)N^k-f_s=0$$

from which the elasticity reported in (11) is derived.

Appendix B

Table B.1
Descriptive Statistics for Dummy Variables in Hardware Regressions. Nine Hundred Fifty Seven Observations.

	Mean	Std. Dev.	Minimum	Maximum
Console Dummies:				
3DO	0.046	0.210	0.000	1.000
Sega Dreamcast	0.060	0.237	0.000	1.000
Nintendo Gamecube	0.075	0.264	0.000	1.000
Sega Genesis	0.093	0.291	0.000	1.000
Atari Jaguar	0.041	0.198	0.000	1.000
Super Nintendo	0.079	0.271	0.000	1.000
Nintendo 64	0.113	0.317	0.000	1.000
Nintendo Entertainment System	0.053	0.225	0.000	1.000
PlayStation 2	0.089	0.285	0.000	1.000
PlayStation 3	0.013	0.111	0.000	1.000
Sega Saturn	0.073	0.261	0.000	1.000
Nintendo Wii	0.013	0.111	0.000	1.000
Microsoft Xbox	0.075	0.264	0.000	1.000
Microsoft Xbox 360	0.025	0.156	0.000	1.000
Year Dummies:				
1996	0.092	0.289	0.000	1.000
1997	0.100	0.301	0.000	1.000
1998	0.086	0.280	0.000	1.000
1999	0.069	0.254	0.000	1.000
2000	0.076	0.266	0.000	1.000
2001	0.073	0.261	0.000	1.000
2002	0.080	0.272	0.000	1.000
2003	0.074	0.262	0.000	1.000
2004	0.071	0.257	0.000	1.000
2005	0.061	0.239	0.000	1.000
2006	0.068	0.252	0.000	1.000
2007	0.074	0.262	0.000	1.000
Month Dummies:				
Feb	0.086	0.280	0.000	1.000
Mar	0.085	0.278	0.000	1.000
Apr	0.083	0.275	0.000	1.000
May	0.082	0.274	0.000	1.000
Jun	0.084	0.277	0.000	1.000
Jul	0.080	0.272	0.000	1.000
Aug	0.080	0.272	0.000	1.000
Sept	0.084	0.277	0.000	1.000
Oct	0.084	0.277	0.000	1.000
Nov	0.083	0.275	0.000	1.000

Table B.2
Descriptive Statistics for Dummy Variables in Generalized Linear Model. Seven Hundred Fifty Three Observations.

	Mean	Std. Dev.	Minimum	Maximum
Console Dummies:				
Sega Dreamcast	0.076	0.265	0.000	1.000
Nintendo Gamecube	0.096	0.294	0.000	1.000
Sega Genesis	0.118	0.323	0.000	1.000
Super Nintendo	0.101	0.301	0.000	1.000
Nintendo 64	0.143	0.351	0.000	1.000
PlayStation 2	0.113	0.317	0.000	1.000
PlayStation 3	0.016	0.125	0.000	1.000
Nintendo Wii	0.016	0.125	0.000	1.000
Microsoft Xbox	0.096	0.294	0.000	1.000
Microsoft Xbox 360	0.032	0.176	0.000	1.000
Year Dummies:				
1996	0.053	0.224	0.000	1.000
1997	0.064	0.244	0.000	1.000
1998	0.064	0.244	0.000	1.000
1999	0.069	0.254	0.000	1.000
2000	0.084	0.277	0.000	1.000
2001	0.088	0.283	0.000	1.000
2002	0.102	0.303	0.000	1.000
2003	0.094	0.292	0.000	1.000
2004	0.090	0.287	0.000	1.000
2005	0.076	0.265	0.000	1.000
2006	0.085	0.279	0.000	1.000
2007	0.094	0.292	0.000	1.000
Month Dummies:				
Feb	0.084	0.277	0.000	1.000
Mar	0.084	0.277	0.000	1.000
Apr	0.082	0.275	0.000	1.000
May	0.080	0.271	0.000	1.000
Jun	0.084	0.277	0.000	1.000
Jul	0.081	0.273	0.000	1.000
Aug	0.082	0.275	0.000	1.000
Sept	0.085	0.279	0.000	1.000
Oct	0.084	0.277	0.000	1.000
Nov	0.084	0.277	0.000	1.000

Appendix C

The goal of this appendix is to obtain estimates of the elasticity of the quality index for software with respect to the number of killer applications and non-killer applications. We use a log-log because the relationship between the number of games and the quality index is non-linear by construction. Because of the log-log structure, we do not include observations with either zero killer or non-killer applications. However, including all observations and estimating the relationship linearly with squared terms for both killer and non-killer applications yields similar results.

Finally, we do not adjust the quality index for software, the number of killer applications, or the number of non-killer applications to take into account backward compatible games. The quality index for software is constructed using only games designed for the current system in the hardware estimations. A separate index is constructed for backward compatible games; both indexes are added together for those systems with backward compatible capability. This adds to the non-linearity of the problem. Using the quality index incorporating backward compatible games and including separate backward compatible variables for both killer and non-killer applications yields similar results in the linear estimation. However, too many degrees of freedom are lost in the log-log estimation; observations with zero backward compatible killer or non-killer applications have to be removed.

Descriptive statistics and OLS results are displayed in Table C.1 and Table C.2, respectively.

Table C.1
Descriptive Statistics for Quality Index for Software Regression. Seven Hundred Forty Six Observations.

	Mean	Std. Dev.	Minimum	Maximum
Dependent Variable:				
ln (Quality Index for Software)	14.838	2.237	5.208	17.896
Regressors:				
ln (Number of Killer Applications)	0.239	0.616	0.000	2.996
ln (Number of Non-Killer Applications)	0.260	0.696	0.000	3.497

Table C.2
Quality Index for Software Estimation: Natural Log of Quality Index for Software Regressed on Natural Log of Number of Killer Applications and Natural Log of Number of Non-Killer Applications.

	OLS Estimation	
	Coefficient	Standard Error
ln (Number of Killer Applications)	1.552*	0.097
ln (Number of Non-Killer Applications)	0.047	0.064
Number of Observations	746	
R^2	0.929	

* indicates significance at the 1% level.
Heteroskedastic consistent standard errors are presented.
The dependent variable is the natural log of the quality index for software.
A constant as well as console, year, and month dummies are included in estimations but not displayed for brevity.

Appendix D

The goal of this appendix is to demonstrate how to obtain the marginal effect of each game for each console.

First we construct the marginal effect for each game in each period in the data set, the 'period marginal effect.' This is done by finding the 'base' predicted market share: predicted market share for all consoles given all games are available.[22] Using this, we calculate the 'base' predicted number of new consumers for each console in each period by multiplying the base predicted market share by the total number of new console adopters in each period. Next we find the 'alternative' market share for each game in the data set. The alternative market share is the predicted market share for a console when the quality index for software for that console is calculated without the specific game. The 'alternative' number of new consumers for a specific game in a period is calculated by multiplying the alternative predicted market share by the total number of new console adopters in that period. Essentially, the alternative number of new consumers is the predicted number of new consumers for a console if a particular game was not available. Finally, the period marginal effect for a game is computed as the difference between the base number of new consumers and the alternative number of new consumers.

To use an example to illustrate, assume the base predicted market share for some console in a particular period is 10%. If there are 100,000 total new console adopters in that period then the base predicted number of new consumers is 10,000 (= 100,000 × 0.1). Assume removing a particular game from the data set results in an alternative predicted market share of 9%. The alternative number of new consumers is then 9,000 (= 100,000 × 0.09). Therefore, the period marginal effect of the game removed from the data set is 1,000 (= 10,000 – 9,000) consumers.

[22] All predictions are obtained using the GMM Estimation for hardware adoption in Table 5.

We calculate the marginal effect of a game by summing the period marginal effects. Continuing the example above, assume the game is available for two periods where the period marginal effects are 9,000 and 6,000 in periods 1 and 2, respectively. The marginal effect of the game is then 15,000 (= 9,000 + 6,000) consumers.

References

Anderson, S. P., de Palma, A., Thisse, J. F., 1992. Discrete Choice Theory of Product Differentiation. Cambridge, Massachusetts: MIT Press.

Berry, S., 1994. Estimating discrete-choice models of product differentiation. RAND Journal of Economics 25 (2), 242 – 262.

Berry, S., Levinsohn, J., Pakes, A., 1995. Automobile prices in market equilibrium. Econometrica 63 (4), 841 – 890.

Chou, C., Shy, O., 1990. Network effects without network externalities. International Journal of Industrial Organization 8, 259 – 270.

Church, J., Gandal, N., 1992. Network effects, software provision, and standardization. The Journal of Industrial Economics 40 (1), 85 – 104.

Church, J., Gandal, N., 1993. Complementary network externalities and technological adoption. International Journal of Industrial Organization 11, 239 – 260.

Clements, M. T., 2005. Inefficient standard adoption: inertia and momentum revisited. Economic Inquiry 43 (3), 507 – 518.

Clements, M. T., Ohashi, H., 2005. Indirect network effects and the product cycle: video games in the U.S., 1994 – 2002. Journal of Industrial Economics 53 (4), 515 – 542.

Coughlan, P. J., 2001. Note on home video game technology and industry structure. Harvard Business School Case 9 – 700 – 107.

Davidson, R., MacKinnon, J., 1981. Several tests for model specification in the presence of alternative hypotheses. Econometrica 49 (3), 781 – 793.

Dixit, A., Stiglitz, J., 1977. Monopolistic competition and optimum product diversity. American Economic Review 67, 297 – 308.

Dubé, J., Hitsch, G., Chintagunta, P., *in press*. Tipping and concentration in markets with indirect network effects. Marketing Science.

Farrell, J., Saloner, G., 1986. Installed base and compatibility: innovation, product preannouncements, and predation. American Economic Review 76 (5), 940 – 955.

Gandal, N., Kende, M., Rob, R., 2000. The dynamics of technological adoption in hardware/software systems: the case of compact disk players. RAND Journal of Economics 31 (1), 43 – 61.

Gretz, R. T., *in press*. Console price and software availability in the home video game industry. Atlantic Economic Journal.

Hausman, J., 1978. Specification tests in econometrics. Econometrica 46 (3), 1251 – 1271.

Hillis, S., 18 May 2001. Video game industry seen poised for fast growth. Xtra. At http://www.xtra.co.nz/technology/0,,827-478466,00.html.

Katz, M. L., Shapiro, C., 1986. Technology adoption in the presence of network externalities. The Journal of Political Economy 94 (4), 822 – 841.

Katz, M. L., Shapiro, C., 1992. Product introduction with network externalities. The Journal of Industrial Economics 40 (1), 55 – 83.

McCullagh, P., Nelder, J.A., 1989. Generalized Linear Models. London: Chapman and Hall.

Nair, H., Chintagunta, P., Dube, J., 2004. Empirical analysis of indirect network effects in the market for personal digital assistants. Quantitative Marketing and Economics 2 (1), 23 – 58.

Nevo, A., 2000. A practitioner's guide to estimation of random-coefficients logit models of demand. Journal of Economics and Management Strategy 9 (4), 513 – 548.

Park, S., 2002. Linear network benefit function in the presence of indirect network externalities. Working Paper, State University of New York at Stony Brook.

Park, S., 2004. Quantitative analysis of network externalities in competing technologies: the VCR case. Review of Economics and Statistics 86 (4), 937 – 945.

Rivers, D., Vuong Q., 2002. Model selection tests for nonlinear dynamic models. Econometrics Journal 5, 1 – 39.

Shankar, V., Bayus, B., 2003. Network effects and competition: an empirical analysis of the home video game industry. Strategic Management Journal 24, 375 – 384.

Shy, O., 2001. The Economics of Network Industries. Cambridge: Cambridge University Press.

Spence, M., 1976. Product selection, fixed costs and monopolistic competition. Review of Economic Studies 43, 217 – 235.

Wackerly, D. D., Mendenhall III, W., Scheaffer, R. L., 1996. Mathematical Statistics with Applications, 5th ed. Belmont, California: Duxbury Press.

Woeckener, B., 2000. The competition of user networks: ergodicity, lock-ins, and metastability. Journal of Economic Behavior and Organization 41 (2), 85 – 99.