慶應義塾大学学術情報リポジトリ
Keio Associated Repository of Academic resouces

Title	日本における鮭鱒の価格決定要因の分析
Sub Title	
Author	南, 啓太(Minami, Keita) 高橋, 大志(Takahashi, Hiroshi)
Publisher	慶應義塾大学大学院経営管理研究科
Publication year	2016
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2016年度経営学 第3209号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40003001-00002016-3209

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the KeiO Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

慶應義塾大学大学院経営管理研究科修士課程

学位論文（　　2016　年度）

論文題名

日本における鮭鱒の価格決定要因の分析

主　査	高橋大志
副　査	大林厚臣
副　査	市来嵜治
副　査	

氏　名	南啓太

論 文 要 旨

所属ゼミ	高橋大志研究会	氏名	南啓太

（論文題名）

日本における鮭鱒の価格決定要因の分析

（内容の要旨）

目的

社会システム、日常生活において食糧は必要不可欠であり、これら食糧の価格形成の要因の解明は重要な意義を持つ。本研究は、日本市場における鮭鱒の価格に焦点を当て、これら水産物の価格形成をシステマチックに説明にすることを目的とする。

手法

（1）マクロ的時系列分析　本分析では鮭鱒を、養殖・天然、及び、冷凍・フレッシュといった属性に基づき 4 つのカテゴリーについて分類し、カテゴリー毎に、餌、在庫量などの要因を基に価格形成の決定要因について分析を行う。

（2）ミクロ的シミュレーション分析　川上から川下まで漁場、生産者、卸売、量販店の４層の人工水産モデルを構築する。それぞれのプレーヤーがバランスシートを持ち、仕入価格、販売価格、純資産の推移を観測する。プレーヤー間の相互作用など、現実の主要な側面を取り込んだ分析を実施する。

結果

（1）期間全体でみると、冷凍及びフレッシュ全体の魚価と、餌の主成分３要素との間には、共和分の関係が確認できた。顕著な時系列の変化は確認できなかった。

（2）各種４魚種のセグメントごとに、特徴が分かれ、養殖、生鮭鱒は Fishmeal、天然、冷凍鮭鱒は Soybean とのグレンジャー因果関係が確認できた。また、冷凍鮭鱒に関しては国内鮭鱒在庫の関係が確認できた。

（3）水産市場において、債務超過に陥った業者の操業停止が、各プレーヤーの相対的力関係に与える影響を、シミュレーション分析を通じ明らかにした。

これらの結果は、水産物の価格形成について興味深い結果を示すものである。

今後の課題

天然鮭鱒についての今回示した結果は、これまで行われておらず、その意味で本研究の意義は大きいと考える。本　分析では、時系列計量分析において、天然物についての季節変動や、フレッシュ鮭鱒の流動性等は今回の分析には考慮されていない。これらの点を考慮した分析は今後の課題として挙げられる。

また、養殖鮭鱒の方が餌の主要素の影響が大きく、その中でも Fishmeal の影響が大きいという点は先行研究及び仮説と整合的な結果であったが、冷凍天然鮭鱒と Soybean との関連については、分析の余地がある。これらの点も今後の課題として挙げられる。

水産市場シミュレーション分析において、現実の側面を取り込んだ分析を実施しているが、考慮する要因を拡張した分析は、今後の課題として挙げられる。

１はじめに

社会システム、日常生活において食糧は必要不可欠であり、これら食糧の価格形成の要因の解明は重要な意義を持つ。我々は日々あまり意識していないかもしれないが、食糧の価格は日々変動しており社会生活に大きな影響を与えている。本レポートでは日本市場における鮭鱒の価格をシステマチックに説明にすることを目的とする。

私自信は KBS 進学前に水産系の専門商社で鮭鱒の担当していたのだが、相場の先読みについては経験則に頼る所が多く、定量的なアプローチをすることで相場の予測可能かどうかは以前から分析してみたいと考えていた。

鮭鱒は従来日本におい天然物の紅鮭、白鮭を中心とした消費であり、相場の変動には大きな変動が見られていたが、1990 年前半以降養殖のチリ産銀鮭が安定に輸入されるようになってからは、比較的一定のサイクルで推移するようになった。

近年水産物の日本の買い負けも囁かれるようになっているが、日本の鮭鱒類の輸入量は大きな変化が見られない、一方で、世界的に見ると中国等の国々の輸入量が大幅に増大してきており、輸入量や消費量に関して、相対的な日本のプレゼンスは低下する一方である。世界的にも水産資源は食糧源としての注目が高まっており、日本においても水産大手 3 社体制寡占の状況から、総合商社等が参入を増やしていることで競争は激化している。安直な買い付けによる相場下落による含み損を避けるためにも、水産物の相場を読む力は一層重要性を増してきているともいえるだろう。

本研究では、コモディティの一種として水産物をとらえ、実際の売買データ等を用いたマクロ的な時系列計量分析と、実際に売買が行われる市場をモデリングしたシミュレーション分析によるミクロ的な分析の 2 つのアプローチで日本における鮭鱒の価格決定要因を分析していきたい。

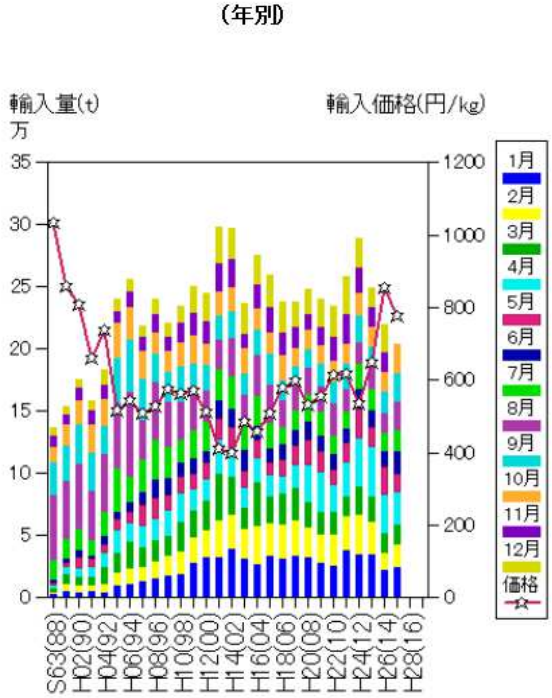

図１　日本の鮭鱒輸入量と平均価格

出典　国立研究開発法人　水産研究・教育機構　北海道区水産研究所　ホームページ

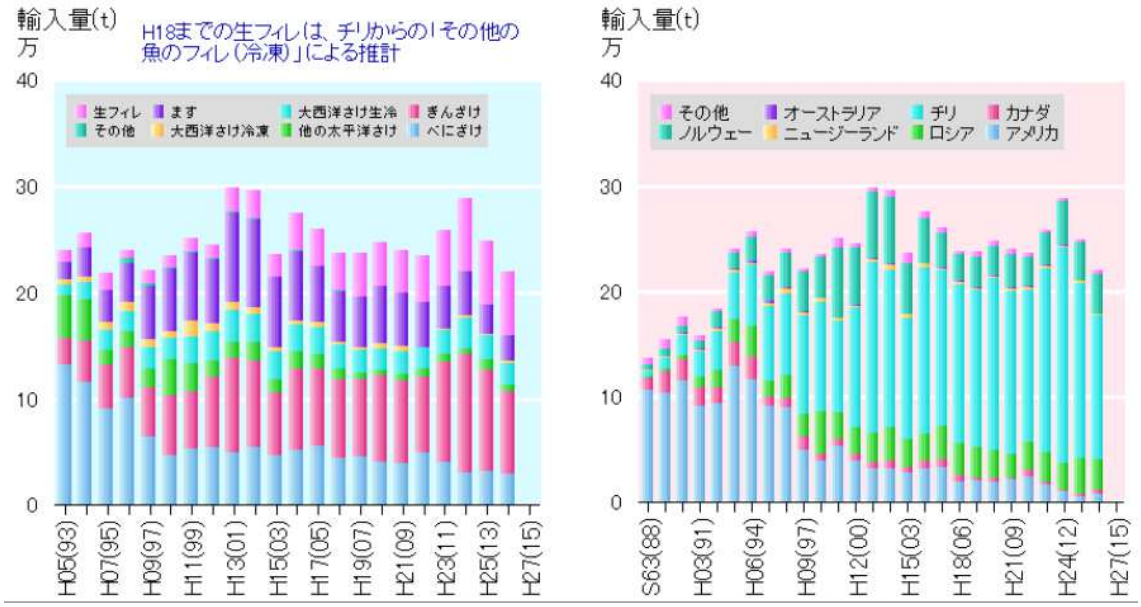

図２　日本の年代・品目別・国別さけます輸入量

出典　国立研究開発法人　水産研究・教育機構　北海道区水産研究所　ホームページ

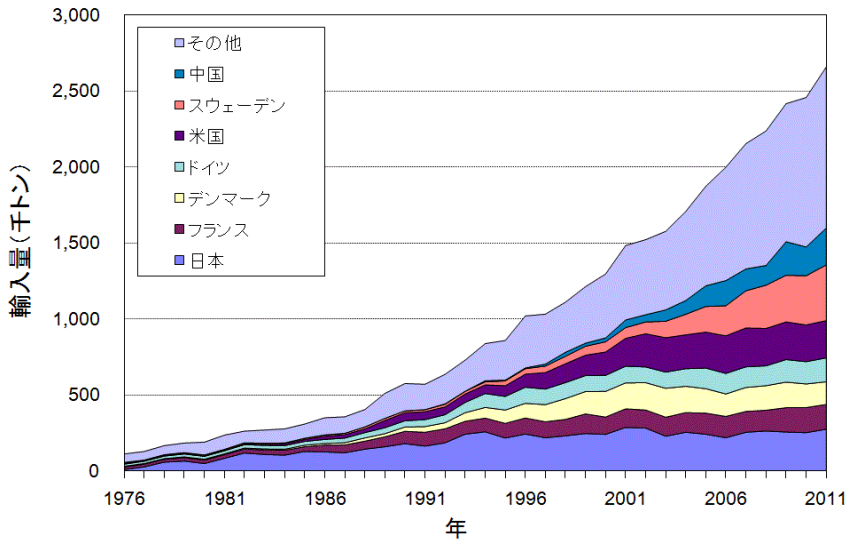

図３　世界の鮭鱒国別輸入量

出典　国立研究開発法人　水産研究・教育機構　北海道区水産研究所　ホームページ

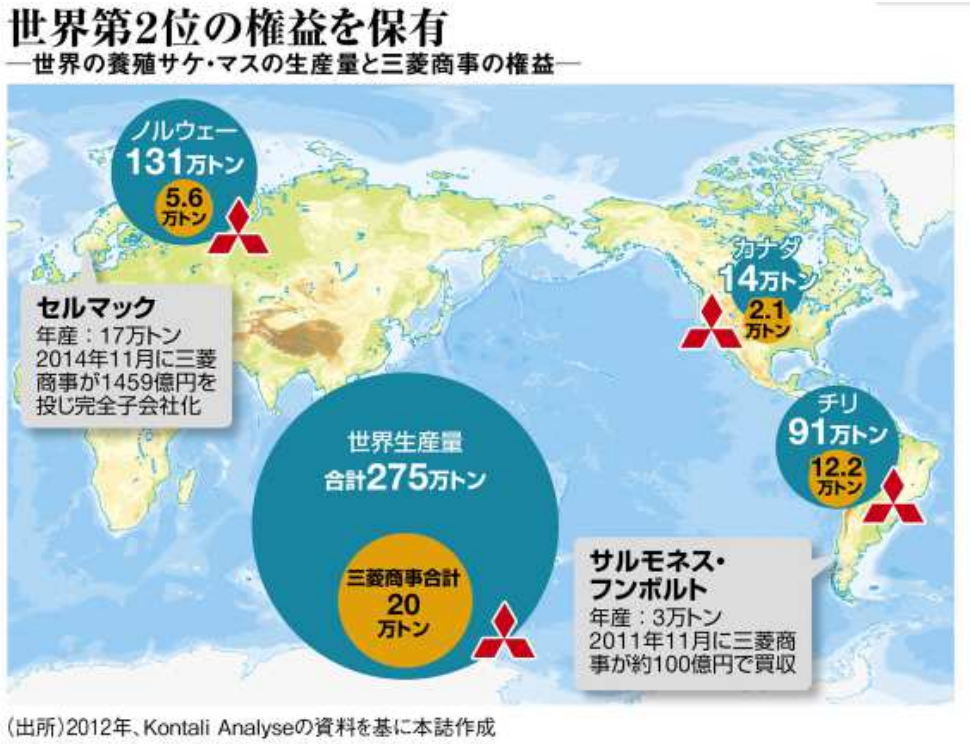

図４　世界の養殖鮭鱒市場における三菱商事の動向
出典　東洋経済オンライン 2015 年 9 月 12 日

２先行研究

主に２タイプの先行研究を紹介する、一つ目はコモディティや時系列分析等のマクロ的な分析関連であり、二つ目は市場モデルシミュレーション分析関連である。

２－１　コモディティや時系列分析等のマクロ的な分析関連

世界的にはコモディテイや水産物の相場に関する研究は多く見られるが、日本においては天然のフレッシュ流通の鮭に関する研究がわずかにあるのみであり、日本国内の鮭鱒消費割合からいえば主流とは言えない程度である。日本国内の消費のメインである養殖鮭鱒や冷凍鮭鱒関する研究は不十分といえる。

２－１－１

“Frank Asche,Atle Oglend” “The relationship between input-factor and output prices in commodity industries The case of Norwegian salmon aquaculture” Journal of commodity markets 2016

ノルウエーにおける鮭鱒の養殖業は大成功ということが出来る。1986 年度の養殖量は 46,500 トンであったが、2013 年度には 1.15 百万トンへと大幅に増加した。世界の輸出量に占めるシェアも大幅に増加した。一方で生産性の向上も限界になり、養殖コストにおける餌の価格の占める寄与度が高くなってきているという問題意識から、過去 30 年のノルウエーサーモンの売価と、投入要素の価格の関係を時系列的に

調査した。結果、2000 年～2015 年にかけて、鮭鱒価格の変動における餌の主要要素の価格変動の影響は高まってきており、近年は共和分の関係にあることが明らかになっている。

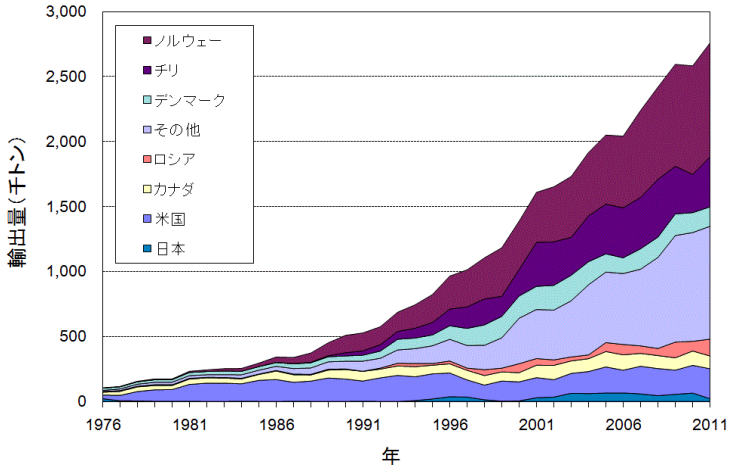

図 5　世界の鮭鱒類国別輸出量
出典　国立研究開発法人　水産研究・教育機構　北海道区水産研究所　ホームページ

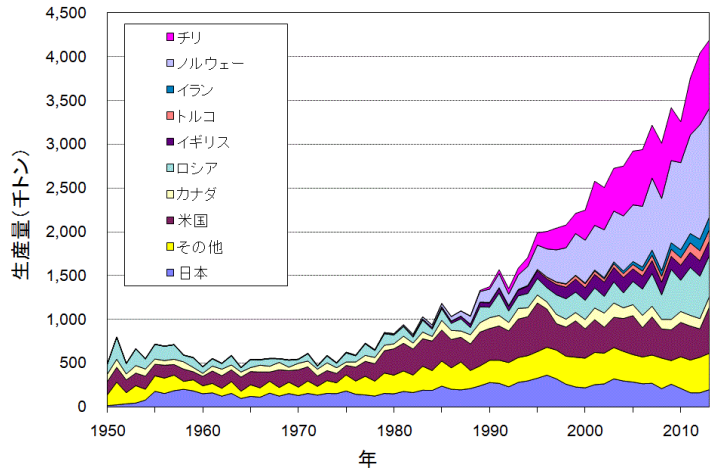

図 6　世界の鮭鱒類国別生産量
出典　国立研究開発法人　水産研究・教育機構　北海道区水産研究所　ホームページ

２－１－２
“Kazuhiko Ohashi Tatuyoshi Okimoto” “Increasing trends in the excess comovement of commodity prices” Journal of commodity markets 2016
　世界的なコモディティ間の相関関係時系列変化を分析している。長期の大きなトレンドと短期の変動について分けて、複数のモデルを用いて分析している。2000 年以降において世界的にコモディティ間の相関は上昇傾向にあるが、オフインデックスの商品に関しては必ずしもそうとは言えないようである。

２－１－３
清水幾太郎　生鮮サケ類の産地価格形成要因に関わる 輸入量と在庫量の影響さけ・ます資源管理センター 研究報告　第 6 号　1-11 頁 2004 年 3 月
　国産の生鮮サケ類は輸入冷凍鮭鱒の価格は増加の影響を受けて低下傾向にある。また、低い流動性による扱いにくさや、輸入鮭鱒が代替品としての存在感を増すことによる需要の低下、国内輸入冷凍鮭鱒の在庫の増減等の影響を受けている。

２－２　市場モデルシミュレーション分析関連
　水産物の商流をモデル化した先行研究は少なく、プレーヤーも卸と買参人の 2 層のみであり、今回試みるような、漁場、生産者、卸、量販店の 4 層まで拡張しているものはまだ前例がないと考えられる。また近年ではコンピューターの性能も向上してきているので、より現実を反映したモデリングが可能になるかもしれない。

２－２－１
“Alan P.Kirman,Nicolass J.Vriend” ”Learning to be loyal. A study of the Marseille Fish Market” Interaction and Market structure 2000
　卸売と買参人のみのモデルで、各プレーヤーの取る購買戦術ごとの有効性を分析している。互いにロイヤリティを構築することで、共進化が生まれるプロセスを明らかにしている。

３目的
３－１　マクロ的アプローチ分析：水産価格データを用いた分析
　先行研究２－１－１を参考に日本市場においても、鮭鱒の価格と養殖鮭鱒の餌の主要３要素との間に関係があることを明らかにしたい。さらに上記に加えて、先行研究では細分化されていなかったが、本研究では鮭鱒を冷凍ＯＲフレッシュ、および、養殖ＯＲ天然、２×２＝４セグメントにまで細分化して、それぞれの鮭鱒ごとの性質の差異や国内鮭鱒在庫との関係まで拡張していきたい。

３－２　ミクロ的アプローチ分析：コンピュータシミュレーション分析
　先行研究２－２－１を参考に、水産物の商流を川上から川下までモデル化し、各プレーヤーが相互に関係をしあっていることから、一部のプレーヤーにショックを与えることでその他のプレーヤーにまで影響を与えることが予測でき、どういった影響を与えるのかを明らかにしたい。問題意識としては主に２つあり、一つ目は 2016 年度にニュースを騒がせた築地市場の豊洲移転問題である、安全面を中心の議論に終始していたが、本来の市場の意義である価格形成にどういった影響があるのかを考察してみたい。二つ目は漁業資源管理による価格形成への影響の差である、日本のようなオリンピック方式と、ノルウエーのようなＩＱ（ＩＴＱ）方式の違いで、どういった差異が起こるのかを考察してみたい。

４データ

４－１築地市場統計情報　H14～H28、冷凍鮭鱒類の月次平均価格

下記の表は東京都卸売市場での鮭鱒の売買価格の時系列データである。今回は４つのセグメントに分けて性質の違いを分析していくが、各セグメントの代表的な魚種の価格を示している。また、次の表で示すような冷凍全体、フレッシュ全体の魚価についても分析を行う。横軸は魚種の名前を、縦軸は時間軸　を示している。

表１　冷凍鮭鱒の東京都卸売市場での取引価格(抜粋)

					単位.円/kg
年	月	冷凍　銀鮭	冷凍　紅鮭	生輸入鮭鱒	生国産白鮭
H14	1	308	648	565	532
	2	303	629	552	644
	3	293	671	566	887
	4	280	617	543	1495
	5	291	688	538	982
	6	331	802	587	966
	7	378	637	606	903
	8	435	638	640	571
	9	479	629	695	345
	10	540	708	731	336
	11	553	706	769	390
	12	552	661	796	534
H15	1	544	650	753	553

出典　東京都卸売市場統計情報　ホームページ

参考までに下記の表に東京都卸売市場での鮭鱒 2015 年度全体の取り扱いデータを載せておく。

表２　2015 年度の築地市場の冷凍鮭鱒の取り扱いデータ

冷凍	数量(トン)	平均単価（円）
合計	21190	761
冷しろさけ	1069.6	617
冷べにさけ	1169.3	912
冷ぎんさけ	13427.6	680
冷ますのすけ	1140.9	961
冷ます	169.2	688
冷その他冷さけ	4213.6	962
フレッシュ	数量(トン)	平均単価（円）
合計	8681.2	1121
しろさけ	1500.7	905
ぎんさけ	689.8	933
ます	63.4	2128
輸入さけ・ます	5868.1	1341
その他さけ	559.2	909

出典　東京都卸売市場統計情報　ホームページより筆者作成

４－２水産庁統計情報　H14～H27　全国消費地月次在庫量

下記の表は水産庁が統計を取っている冷凍鮭鱒の国内在庫の時系列データである。横軸は鮭鱒の種類、縦軸は時間軸である。以下は１年分のサンプルである。以下は１年分のサンプルである。

表３　全国消費地在庫（抜粋）

H15年度	鮭類(トン)	鱒類(トン)
1月	62 877	16 138
2月	65 542	15 345
3月	60 189	15 119
4月	55 409	13 301
5月	49 124	12 176
6月	43 440	11 824
7月	43 775	11 937
8月	47 559	11 326
9月	50 137	10 990
10月	48 004	8 846
11月	40 381	9 271
12月	42 591	8 533

出典　水産庁統計情報　ホームページ

４－３世界銀行　H14～H27　養殖の餌を構成する成分要素の月次価格

本研究では先行研究に習い、fishmeal,soybean,wheat の３要素を鮭鱒養殖に餌の主要素とする。先行研究２－１－１によれば上記３要素で鮭鱒養殖の餌の内訳の約50～80%を占める。サンプル画像を以下に示す

表４　ワールドバンクピンクシート（抜粋）

World Bank Commodity Price Data (The Pink Sheet)

monthly prices in nominal US dollars, 1960 to present

(monthly series are available only in nominal US dollars)

Updated on May 03, 2016

	Crude oil, average ($/bbl)	Crude oil, Brent ($/bbl)	Crude oil, Dubai ($/bbl)	Crude oil, WTI ($/bbl)	Coal, Australian ($/mt)	Coal, Colombian ($/mt)	Coal, South Afican ($/mt)	Natural gas, US ($/mmbtu)	Natural gas, Europe ($/mmbtu)	Liquefied natural gas, Japan ($/mmbtu)	Natural gas index (2010=100)	Cocoa ($/kg)	Coffee, arabica ($/kg)	Coffee, robusta ($/kg)	Tea, avg 3 auctions ($/kg)	Tea, Colombo ($/kg)	Tea, Kolkata ($/kg)	Tea, Mombasa ($/kg)	Coconut oil ($/mt)	Copra ($/mt)	Groundnuts ($/mt)	Fish meal ($/mt)	G u ($
2014M10	86.08	87.27	86.57	84.40	63.71	63.80	65.74	3.77	9.77	15.89	103.02	3.10	4.97	2.31	2.65	3.42	2.61	1.91	1144.00	769.00	1338.00	1689.00	
2014M11	76.99	78.44	76.73	75.81	62.55	63.50	65.66	4.10	8.90	15.59	102.66	2.91	4.62	2.27	2.65	3.33	2.71	1.90	1194.00	795.00	1370.00	1836.00	
2014M12	60.70	62.33	60.52	59.26	62.44	63.75	66.14	3.43	9.83	15.62	99.00	2.95	4.34	2.20	2.62	3.37	2.62	1.88	1217.00	813.00	1360.00	1852.00	
2015M01	47.11	48.07	45.98	47.27	62.10	56.65	62.18	2.97	9.25	15.12	90.34	2.92	4.19	2.16	2.53	3.21	2.26	2.13	1159.00	764.00	1350.00	1792.00	
2015M02	54.79	57.93	55.83	50.61	61.40	57.75	63.28	2.85	8.27	13.37	82.95	2.96	3.94	2.17	2.39	3.14	1.78	2.26	1187.00	794.00	1350.00	1715.00	
2015M03	52.83	55.79	54.91	47.78	60.12	57.64	60.82	2.80	8.27	14.28	83.00	2.88	3.54	2.03	2.36	3.11	1.43	2.54	1096.00	721.00	1300.00	1630.00	
2015M04	57.54	59.39	58.80	54.44	57.81	55.31	59.34	2.58	7.42	10.22	73.52	2.87	3.62	2.03	2.64	3.06	2.29	2.56	1080.00	714.00	1300.00	1539.00	
2015M05	62.51	64.56	63.69	59.27	60.40	54.34	61.88	2.84	7.27	8.72	74.94	3.10	3.49	1.93	2.81	2.96	2.65	2.83	1133.00	748.00	1290.00	1525.00	
2015M06	61.31	62.34	61.78	59.80	58.84	53.25	60.96	2.77	7.29	8.59	74.10	3.24	3.52	1.99	2.91	2.98	2.75	3.00	1131.00	748.00	1280.00	1506.00	
2015M07	54.34	55.87	56.25	50.90	59.13	52.34	57.05	2.83	6.93	8.87	73.32	3.33	3.41	1.92	3.00	2.98	2.92	3.09	1100.00	735.00	1280.00	1466.00	
2015M08	45.69	46.99	47.22	42.86	58.57	49.65	54.36	2.76	6.95	9.18	72.81	3.15	3.46	1.89	2.88	2.87	2.84	2.92	1037.00	689.00	1150.00	1469.00	
2015M09	46.28	47.24	46.15	45.45	54.75	49.13	51.44	2.65	6.71	9.64	70.56	3.28	3.22	1.80	2.69	2.63	2.59	2.85	1063.00	699.00	1150.00	1480.00	
2015M10	46.96	48.12	46.55	46.20	52.31	48.56	49.87	2.32	6.43	9.44	65.08	3.20	3.38	1.82	2.79	2.76	2.59	3.01	1108.00	736.00	1150.00	1531.00	
2015M11	43.11	44.42	42.22	42.70	52.57	50.76	53.33	2.08	6.24	8.89	60.92	3.36	3.26	1.80	2.77	2.89	2.54	2.87	1073.00	716.00	1175.00	1537.00	
2015M12	36.57	37.72	34.77	37.23	52.13	44.70	49.98	1.92	6.10	8.50	58.11	3.35	3.28	1.75	2.72	2.89	2.43	2.85	1147.00	759.00	1200.00	1503.00	
2016M01	29.78	30.80	27.00	31.54	49.82	43.01	49.93	2.27	5.35	7.85	58.37	2.95	3.20	1.65	2.50	2.84	2.07	2.58	1155.00	763.00	1175.00	1476.00	
2016M02	31.03	33.20	29.50	30.39	50.71	41.38	51.49	1.96	4.90	8.01	52.49	2.92	3.26	1.63	2.31	2.79	1.78	2.37	1216.00	813.00	1150.00	1472.00	
2016M03	37.34	39.07	35.18	37.77	52.19	43.84	53.15	1.70	4.26	7.75	46.17	3.07	3.47	1.67	2.28	2.82	1.81	2.21	1448.00	990.00	1150.00	1448.00	
2016M04	40.75	42.25	39.04	40.96	51.18	43.17	53.47	1.90	4.13	7.75	48.01	3.08	3.40	1.77	2.50	2.94	2.52	2.04	1590.00	1045.00	1158.00	1477.00	

出典　世界銀行ホームページ

４－４DATASTREAM　その他追加情報等

為替、株価等その他分析に必要な情報等、随時検索する。

５分析方法

先行研究を調査していくうちに、マクロ的な時系列分析のアプローチと、ミクロ的な市場モデリングのアプローチがあり、今回はその両方からのアプローチを取る。

５－１マクロ編：鮭鱒価格と養殖の餌の主要３要素との関係の時系列分析

先行研究と時系列分析の定石に従って、以下の手順を踏まえる。見かけの相関に注意して、十分な頑健性を得るように心がける。基本的には主にＲを使用していく。

先行研究： Frank Asche,Atle Oglend(2016)　ノルウエーの養殖産業において、価格変動は餌の変動の影響が大きい。

「The relationship between input-factor and output prices in commodity industries The case of Norwegian salmon aquaculture」 Journal of commodity markets

$$P_t = \beta_0 + \beta_1 t + \beta_2 P_{fm,t} + \beta_3 P_{soy,t} + \beta_4 P_{wheat,t} + \epsilon_t,$$

Model C: sale price on raw materials

	Coef.	t-Statistic[a]	p-Value[b]
Const.	3.218	9.76	–
Trend	−0.160	−7.68	–
Fishmeal	3.289	5.53	0.0158
Soybean	0.199	0.067	0.4790
Wheat	1.168	0.294	0.4445
R^2	0.729		
F-test	24.504	0.0133	

統計的手法による分析

（１）　重回帰分析　上記先行研究をベースに、被説明変数を冷凍ＯＲフレッシュ、及び、養殖ＯＲ天然に細分化、説明変数を国内在庫にまで拡張する。モデル：P（魚価）＝α×P（fishmeal）＋β×（soybean）＋γ×（wheat）＋δ×鮭（salmon）在庫＋ε×鱒（trout）在庫とする。仮説としては養殖鮭鱒の方が天然鮭鱒より R2 が大きくなること、及び、冷凍鮭鱒は在庫の影響を受けるということを考えている。

（２）　単位根検定　ADF unit-root test を用いて、各変数が単位根過程か否かを検定する。

（３）　時系列主成分分析、時系列相関係数分析　主成分分析の第一主成分、及び第２主成分で説明できる割合が時系列で変化しているのか否かを調べる。

（４）　共和分検定 Johansen-trace-tests　魚価と餌の主要３要素との間に一定の関

係があるのか否かを検定する。

ＶＡＲ解析

（５）　グレンジャー因果検定　ＶＡＲモデルにてＡＩＣが最少となるようなラグを推定する。餌の主要３要素の魚価へのグレンジャー因果性を調べる。

（６）　インパルス応答関数　餌の主要３要素から魚価へのインパルス応答関数を描き、比較してみる。

（７）　予測誤差分散分解　各魚価を餌の主要３要素で分散分解して、各魚種により違いがあるのか否かを調べる。

５－２ミクロ編：エージェントベースモデルにおける、水産市場についての考察

川上から川下まで漁場、生産者、卸売、量販店の４層の人工水産業市場モデルを構築する。モデルにおいては、それぞれのプレーヤーがバランスシートを持ち、計算機シミュレーションを通じ、仕入価格、販売価格、純資産の推移を分析する。プレーヤー間の相互作用など、現実の主要な側面を取り込んだ分析を実施する。本研究のシミュレーションでは、㈱ＮＴＴ数理システムのＳ４を用いた。

モデル概要　ニューラルネットワーク状な商流の人工水産業市場モデルとする。

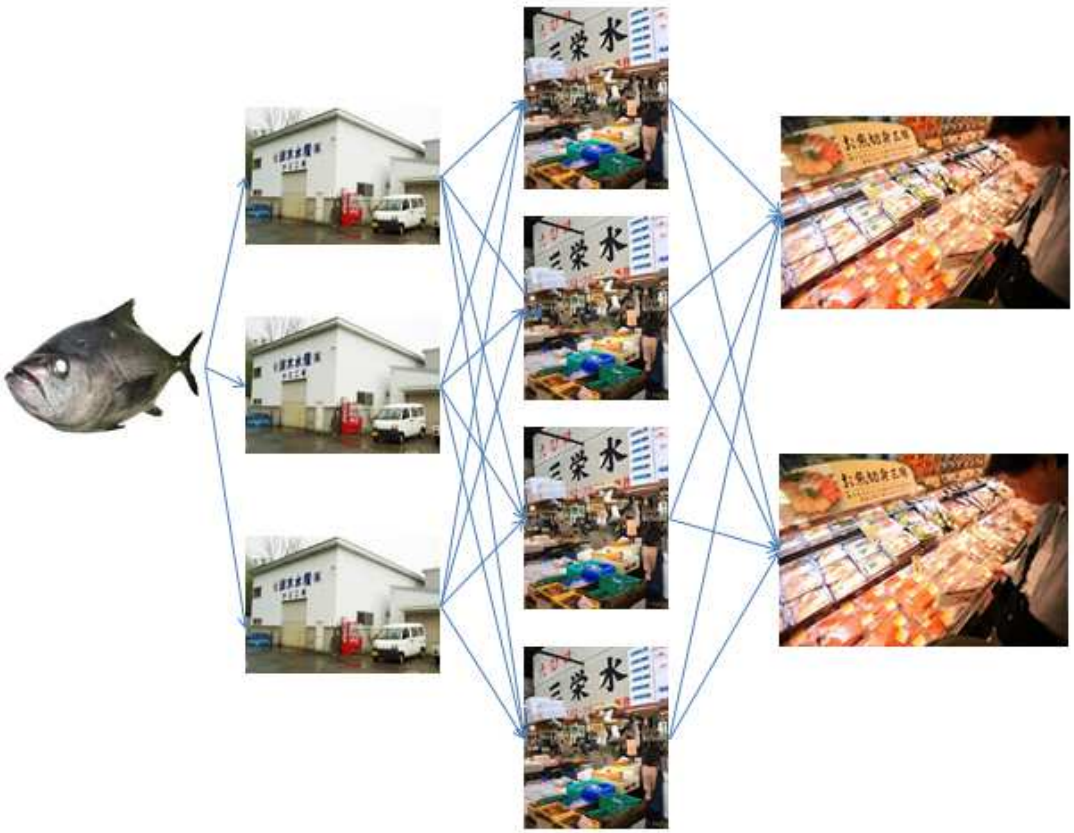

図８　人工水産業市場モデルイメージ図

漁場（正規分布）→生産者（1,2,3,i,,n）→ 卸売(1,2,3,j,,m)→量販店(1,2,3,z,,l)

漁場
水産物は一種類と仮定する
漁獲高正規分布　S_t～（μ , σ^2）に従う

生産者（Producer）
Pr1,Pr2,Pr3,,,,,Pr,i,,,,Pr,n 計 n 社
漁場→生産者
各生産者の仕入れ量（相対）
$W_{i,,t} = W^0_i$（ベース値）＋ $\varepsilon_{i,t}$（誤差項）
各生産者 i の仕入量
$V_{i,,t} =$（$W_{i,,t} / \sum_{i=1}^{n} Wi,,t$）$\times S_t$
生産者 i の仕入価格
$P_{i,t} = -a \times V_{i,,t}$（購入規模）$-b \times S_t$（漁獲高）$-c \times Pr_n$（生産者規模）$+d$

卸売(Wholesale)
Wh1,Wh2,Wh3,,,,,,Wh,j,,,,,Wh,m 計 m 社
生産者→卸売
卸売の仕入量（相対）
$W_{j,i,t} = W^0_{j,i}$（ベース値）＋ $\varepsilon_{j,i,t}$(誤差項)
卸売 j の生産者 i からの仕入量
$VW_{j,i,t} = (w_{j,i,t} / \sum_{k=1}^{m} wk,i,t) \times V_{i,t}$
卸売 j の生産者 i からの仕入価格
$P_{j,i,t} = -\alpha \times VW_{j,i,t}$（購入規模）$-\beta \times S_t$（漁獲高）$-\gamma \times Wh_m$（卸売の規模）$+\zeta \times Pr_n$（生産者の規模）$+\lambda$
卸売 j の仕入量
$VW_{j,t} = \sum_{s=1}^{n} VWj,s,t$
＃βについては過去の時系列の漁獲高の影響を受ける（詳細は後述のコードを参照）

量販店(Supermarket)
SU1,SU2,SU3,,,,,SU,z,,,,,SU,l 計 l 社
卸売→量販店
量販店の仕入量（相対）
$W^s_{z,j,t} = W^{s0}_{z,j}$（ベース値）＋ $\varepsilon_{z,j,t}$（誤差項）
量販店 z の卸売 j からの仕入量
$VS_{z,j,t} = (w_{z,j,t} / \sum_{v=1}^{l} wz,v,t) \times VW_{j,t}$
量販店 z の卸売 j からの仕入価格

$P_{z,j,t}$= －A×$VS_{z,j,t}$ （購入規模）$_t$－B×S_t（漁獲高）－C×SU_z（量販店の規模）＋D×Wh_j（卸売の規模）＋E

各卸売の損益について
仕入費用
卸売 j の生産者 i からの仕入費用
$VW_{j,i,t}$×$P_{j,i,t}$
卸売 j の仕入費用
$WC_{j,t}=\sum_{f=1}^{n} VWj,f,t \times P_{j,f,t}$
売上
卸売 j の量販店 z への売り上げ
$VS_{z,i,t}$×$PS_{z,i,t}$
卸売 j の量販店への売り上げ
$WS_{j,t}=\sum_{g=1}^{l} VSg,j,t \times PS_{g,j,t}$
卸売 j の損益
$WP_{j,t}$= $WS_{j,t}$－$WC_{j,t}$＋F
卸売 j のバランスシート
$A_{j,t}$（資産）= $E_{j,t}$（資本）＋$D_{j,t}$（負債）
$E_{j,t+1}$= $E_{j,t}$＋$WP_{j,t}$

各生産者、各量販店の損益についても、上記卸売の損益と同様に設定する

６分析結果
マクロ編：水産価格データを用いた分析
鮭鱒価格と養殖の餌の主要３要素との関係の時系列分析

統計的解析
（１） 重回帰分析
「魚価＆餌＆トレンド」
　４つのセグメントの魚種ごとにどういった違いが出てくるかを考察するために、魚価を被説明変数として、餌の主要３要素とトレンドを説明変数として重回帰分析を行った。
　全体的に説明力は養殖＞天然およびフレッシュ＞冷凍の傾向にあるといえる。養殖かつフレッシュの鮭鱒においては非常に高いＲ２となっているし、餌の３要素すべてにおいて有意となっている。一方で天然かつ冷凍鮭鱒においては、餌の３要素すべてにおいて有意でなく、天然かつフレッシュ鮭鱒においてはＲ２が他の魚種に比べてかなり低い値となっている。トレンドは４魚種すべてにおいて有意となっている。
　参考までに冷凍鮭鱒全体、フレッシュ鮭鱒全体でも同様のことを行った。フレッシ

ュにおいては養殖と天然中間くらいの説明力となったが、冷凍においては大きく説明力が低下してしまった。

表５　重回帰分析結果　魚価と餌の主要素とトレンド

冷凍養殖チリ銀鮭				冷凍天然ロシア紅鮭			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	408.26	12.70	2e-16 ***	Const	515.42	10.83	2e-16 ***
Trend	0.85	2.61	0.009740 **	Trend	1.27	2.62	0.00946 **
Fishmeal	1.17	3.58	0.000443 ***	Fishmeal	0.43	0.90	0.37
Soybean	0.60	0.45	0.66	Soybean	3.20	1.62	0.11
Wheat	-4.47	-2.66	0.008510 **	Wheat	-2.71	-1.09	0.28
R2	0.45			R2	0.38		
F-test	37.39			F-test	27.87		
フレッシュ養殖輸入鮭鱒				フレッシュ国産白鮭			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	440.56	12.08	2e-16 ***	Const	670.73	6.99	5.69e-11 ***
Trend	1.11	3.01	0.00303 **	Trend	-0.27	-0.28	0.78
Fishmeal	2.50	6.75	2.11e-10 ***	Fishmeal	2.60	2.67	0.00838 **
Soybean	7.44	4.90	2.15e-06 ***	Soybean	3.88	0.97	0.33
Wheat	-6.38	-3.35	0.00101 **	Wheat	-13.90	-2.77	0.00621 **
R2	0.80			R2	0.11		
F-test	183.70			F-test	6.52		
冷凍鮭鱒全体				フレッシュ鮭鱒全体			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	536.69	11.89	2e-16 ***	Const	438.80	9.35	2e-16 ***
Trend	-0.07	-0.16	0.87	Trend	1.28	2.68	0.00801 **
Fishmeal	1.40	3.05	0.00264 **	Fishmeal	2.24	4.71	5.06e-06 ***
Soybean	1.87	0.98	0.33	Soybean	5.58	2.80	0.00564 **
Wheat	-3.09	-2.55	0.01165 *	Wheat	-6.05	-2.41	0.01680 *
R2	0.15			R2	0.67		
F-test	8.84			F-test	89.20		

「魚価＆餌＆トレンド＆国内鮭鱒在庫」

前節から４つのセグメントごとの魚種のだいたいの傾向が見えてきたので、続いて、説明変数に国内在庫を追加して同様の分析を行い、どのような変化があるのかを確認した。

冷凍鮭鱒においては鮭鱒国内在庫を説明変数に加えることでＲ２が増加する傾向にあるし、有意水準も満たしていることが確認できた。一方でフレッシュ鮭鱒においてはＲ２がほとんど上昇しないと同時に有意水準を満たしていない傾向が確認できた。

冷凍鮭鱒全体においては在庫を加えることでＲ２が大幅に増加した、一方でフレッシュ鮭鱒全体にはほとんど影響がないと同時に、有意水準に至っていないことが確認できた。

表６　重回帰分析結果　魚価と餌の主要素とトレンド、国内鮭鱒在庫

冷凍養殖チリ銀鮭				冷凍天然ロシア紅鮭			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	626.80	12.77	2e-16 ***	Const	767.40	10.23	2e-16 ***
Trend	1.35	3.84	0.000178 ***	Trend	2.56	4.77	4.05e-06 ***
Fishmeal	1.40	4.73	4.92e-06 ***	Fishmeal	0.41	0.90	0.37
Soybean	-1.30	-0.99	0.32	Soybean	0.20	0.10	0.92
Wheat	-2.60	1.53	0.13	Wheat	2.40	0.92	0.36
stock1	0.00	-3.39	0.000875 ***	stock1	0.00	-4.83	3.23e-06 ***
stock2	-0.01	-3.64	0.000370 ***	stock2	-0.01	-2.18	0.031 *
R2	0.49			R2	0.49		
F-test	27.39			F-test	27.75		
フレッシュ養殖輸入鮭鱒				フレッシュ国産白鮭			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	756.10	14.10	2e-16 ***	Const	673.47	4.18	4.86e-05 ***
Trend	2.04	5.31	3.58e-07 ***	Trend	-0.21	-0.18	0.86
Fishmeal	2.73	8.46	1.63e-14 ***	Fishmeal	0.86	2.50	0.0134 *
Soybean	4.67	3.26	0.00137 **	Soybean	2.83	0.66	0.51
Wheat	-4.55	-2.46	0.01517 *	Wheat	-9.32	-1.67	0.0967 .
stock1	0.00	1.86	0.06497 .	stock1	0.00	-1.66	0.0992 .
stock2	-0.02	6.48	1.08e-09 ***	stock2	0.01	0.91	0.37
R2	0.83			R2	0.1411,		
F-test	134.00			F-test	4.38		
冷凍鮭鱒全体				フレッシュ鮭鱒全体			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	670.30	11.90	2e-16 ***	Const	537.00	6.61	5.39e-10 ***
Trend	1.78	4.42	1.82e-05 ***	Trend	1.44	2.48	0.0141 *
Fishmeal	1.17	3.46	0.000704 ***	Fishmeal	2.32	4.75	4.55e-06 ***
Soybean	-1.27	-0.84	0.40	Soybean	4.98	00　2.295	0.0230 *
Wheat	-2.12	-1.09	0.28	Wheat	-6.16	2.19	0.0297 *
stock1	0.00	-2.50	0.013562 *	stock1	0.00	0.35	0.72
stock2	-0.01	-3.66	0.000347 ***	stock2	-0.01	-1.61	0.11
R2	0.4928,			R2	0.6319,		
F-test	25.91			F-test	45.78		

冷凍鮭鱒とフレッシュ鮭鱒で、影響の受け方にラグが生じる可能性もあるので、その点に関しては Appendix にて本節のような月次データではなく 12 か月移動平均での結果をもとに考察をした。

（２）　単位根検定

前節では４つの魚種のセグメントごとの性質の違いと、在庫の効果が比較的わかりやすい傾向にて確認できたが、時系列分析においては見かけの相関に注意するべきであるので、複数の異なる手法でも分析を行い、結果の頑健性を担保するようにしたい。

各魚種単体では養殖の魚種は単位根過程であったが、冷凍及びフレッシュの全体になると単位根過程ではなかった。一方で天然の鮭鱒は単位根過程ではなかった。餌の主な構成要素３つはすべて単位根過程であった。国内在庫は単位根過程ではなかった。

表 7　ADF unit-root test（単位根検定）

PP.test	Dickey-Fuller	p-value	単位根
冷凍　養殖	-2.6618	0.2997	○
冷凍　天然	-7.7315	0.01	×
フレッシュ　養殖	-2.4506	0.3879	○
フレッシュ　天然	-6.8748	0.01	×
冷凍全体	-6.8267	0.01	×
フレッシュ全体	-4.4141	0.01	×
Fishmeal	-2.598	0.3263	○
Soybean	-3.1043	0.1147	○
Wheat	-2.263	0.4663	○
Salmon stock	-3.472	0.04716	×
Trout stock	-3.6888	0.02718	×

上記結果を踏まえ、単位根過程と非単位根過程が混じるので、前節の回帰分析のみでは危険であり、また、非定常状態のデータの時系列分析は慎重に行う必要がある。

以降の分析では必要に応じて差分データを使用して分析を行う。

（３）時系列主成分分析、相関関係分析

「主成分分析、単年度時系列及び、2002～2016 年通年」

変数に単位根過程が混じるため、仮説検証の頑健性を出すためにも別の手法でも分析をしていく。まず、主成分分析（魚価４種及び餌の主要３要素）の第一主成分及び第二主成分まで説明できる割合の変化を見ていったが、第一主成分で約５０～６０％ほど、第二主成分までで約８０％前後という非常に高い割合が説明出来ていることが確認できたが、時系列の顕著な変化は確認できなかった（先行研究に習う形で実施、各成分の係数は考慮していない）。また、変数に在庫を加えることで累積寄与率は増加した、こちらも時系列の変化も見てみたが、主だった変化は確認できなかった。

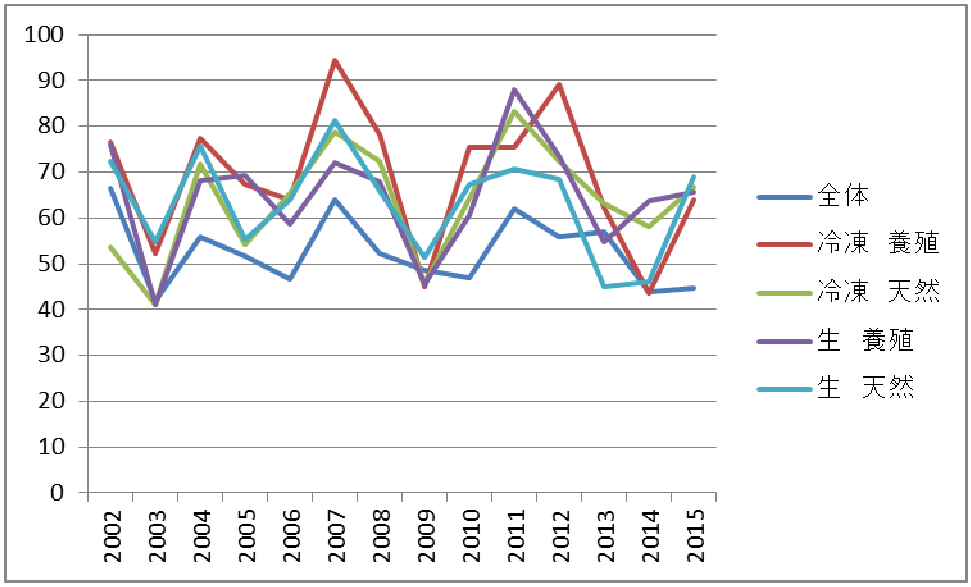

図 9　主成分分析においての時系列累積寄与率（魚価４種及び餌の主要３要素、第１主成分まで）横軸は年度（年）、縦軸は主成分の累積比率（%）である。

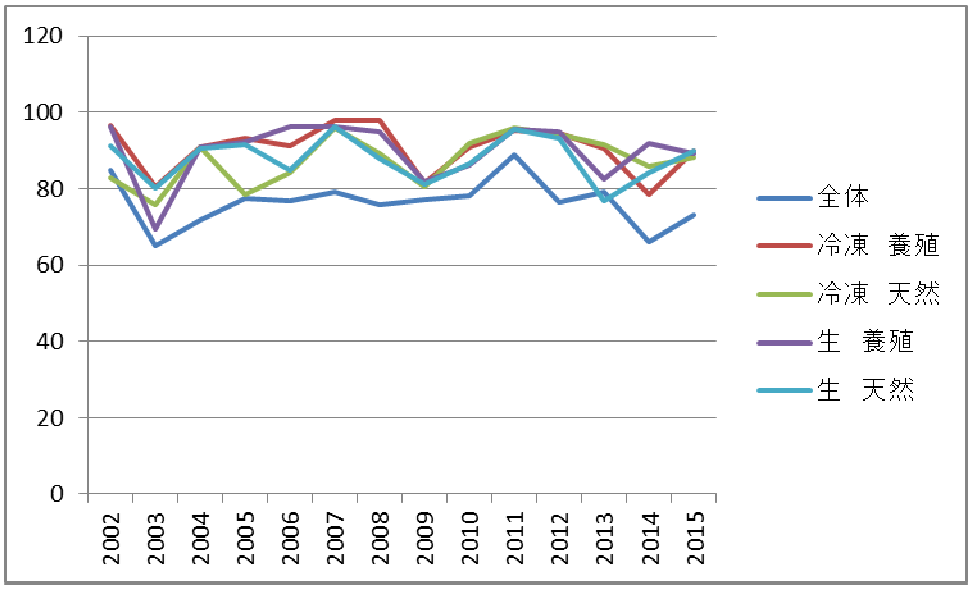

図 10　主成分分析においての時系列累積寄与率（魚価４種及び餌の主要３要素、第２主成分まで）横軸は年度（年）、縦軸は主成分の累積比率（%）である。

表 8　主成分分析においての累積寄与率（2002～2016 年全体）

累積比率	第一主成分	第二主成分
在庫なし	0.5915	0.8983
在庫有り	9.78E-01	1.00E+00

「相関係数 2002 年～2016 年」

基礎統計量の一種ではあるが、分析結果として確認した。フレッシュ養殖輸入鮭鱒は餌の主要な構成３要素とまんべんなく高い相関を示している、一方でフレッシュ天然国産白鮭においてはまんべんなく低い相関を示している。期間を分けて時系列の変化を見てみたが大きな変化は確認できなかった。

餌の主要な３要素同士の相関も大きく、餌の構成を変更することでコスト削減をすることが難しいことが推測される。

表 9　相関係数（2002～2016 年全体）

	冷凍　養殖	冷凍　天然	生　養殖	生　天然	fishmeal	soybean	wheat
冷凍　養殖	1.00	0.52	0.77	0.36	0.62	0.44	0.26
冷凍　天然		1.00	0.54	0.28	0.52	0.55	0.32
生　養殖			1.00	0.39	0.87	0.79	0.59
生　天然				1.00	0.39	0.39	0.39
fishmeal					1.00	0.74	0.72
soybean						1.00	0.72
wheat							1.00

ＶＡＲ解析

（４）　グレンジャー因果関係分析（差分で実行）

各変数に単位根過程が混じるので、回帰分析以外のアプローチでも餌の影響を見たいので、魚価との間にグレンジャー因果関係があるのか否かを検定した。下準備として統計ソフトのＲを用いてＶＡＲモデルをＡＩＣが最少となるように推定した。グレンジャー因果関係が優位であったのは、①フレッシュ養殖輸入サーモンへの fishmeal と②冷凍天然ロシア産紅鮭への soybean の２関係のみであった。念のため冷凍鮭鱒全体とフレッシュ鮭鱒全体でも同様の検定を行ったが３種全ての餌の要素において優位ではなかった。

表 10　グレンジャー因果検定の結果

冷凍　養殖	p-value	G因果	冷凍　天然	p-value	G因果
fishmeal	0.7912	×	fishmeal	0.7398	×
soybean	0.7974	×	soybean	0.0201	○
wheat	0.7974	×	wheat	0.3261	×
生　養殖	**p-value**	**G因果**	**生　天然**	**p-value**	**G因果**
fishmeal	0.0020	○	fishmeal	0.7754	×
soybean	0.7439	×	soybean	0.6966	×
wheat	0.4404	×	wheat	0.4070	×
冷凍全体	**p-value**	**G因果**	**生全体**	**p-value**	**G因果**
fishmeal	0.8170	×	fishmeal	0.2402	×
soybean	0.7144	×	soybean	0.5566	×
wheat	0.4988	×	wheat	0.4276	×

（５） インパルス応答関数分析（差分で実行）

前節で一部グレンジャー因果関係が優位な関係が確認できたが、その影響の強さは図ることができないためインパルス応答関数も作成した。前節でグレンジャー因果関係のあった、フレッシュ養殖輸入サーモンへの fishmeal と冷凍天然ロシア産紅鮭への soybean の２つ以外は９５％区間が非常に広がってしまっている。冷凍天然ロシア産紅鮭への soybean のインパクトはプラスとマイナスの両方に触れているため、グレンジャー因果関係がありかつ、プラスのみへのインパクトを持つのは、フレッシュ養殖輸入サーモンへの fishmeal のみである。養殖鮭鱒においてはコストアップの影響がある一方で、天然鮭鱒においては魚価をかく乱させるような影響を与えるのかもしれないことが予測される。

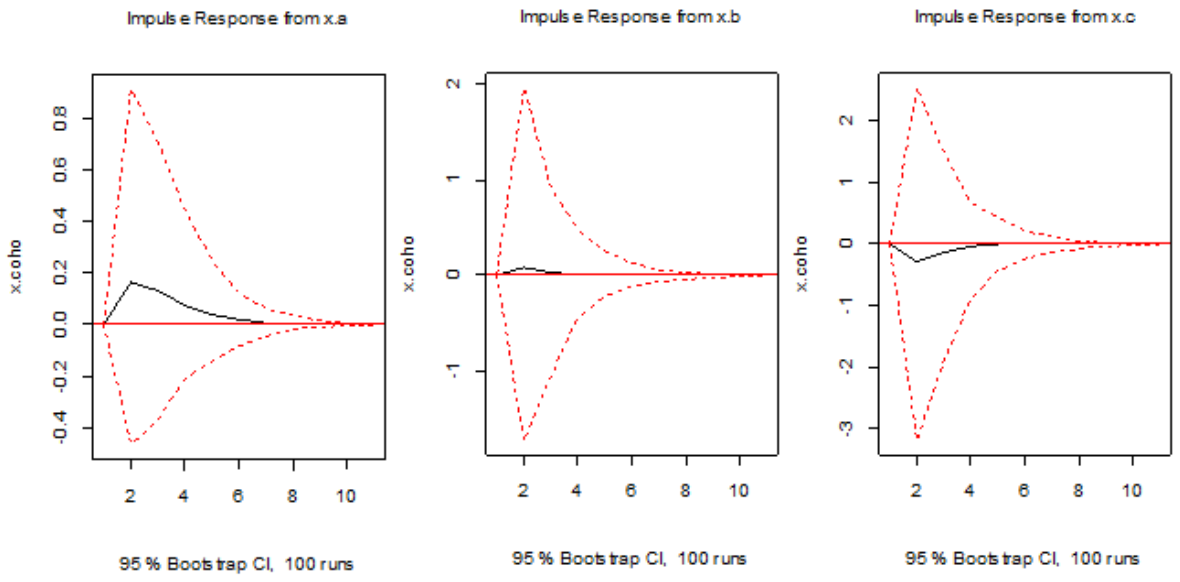

図 11　冷凍養殖チリ銀鮭へのインパルス応答関数 a=fishmeal b=soybean c=wheat

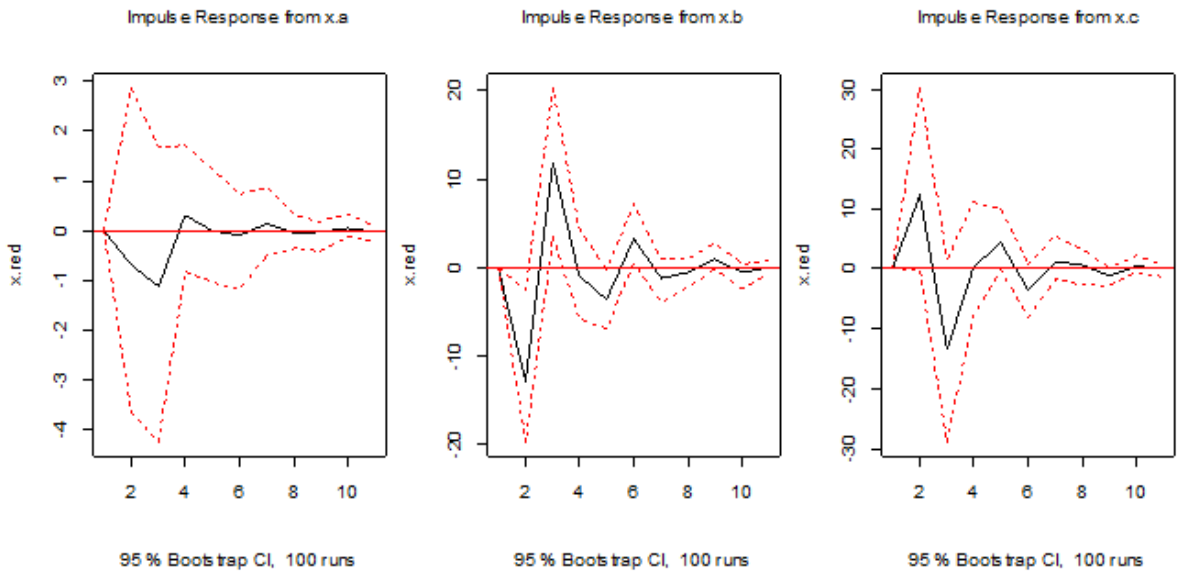

図 12 冷凍天然ロシア産紅鮭へのインパルス応答関数 a=fishmeal b=soybean c=wheat

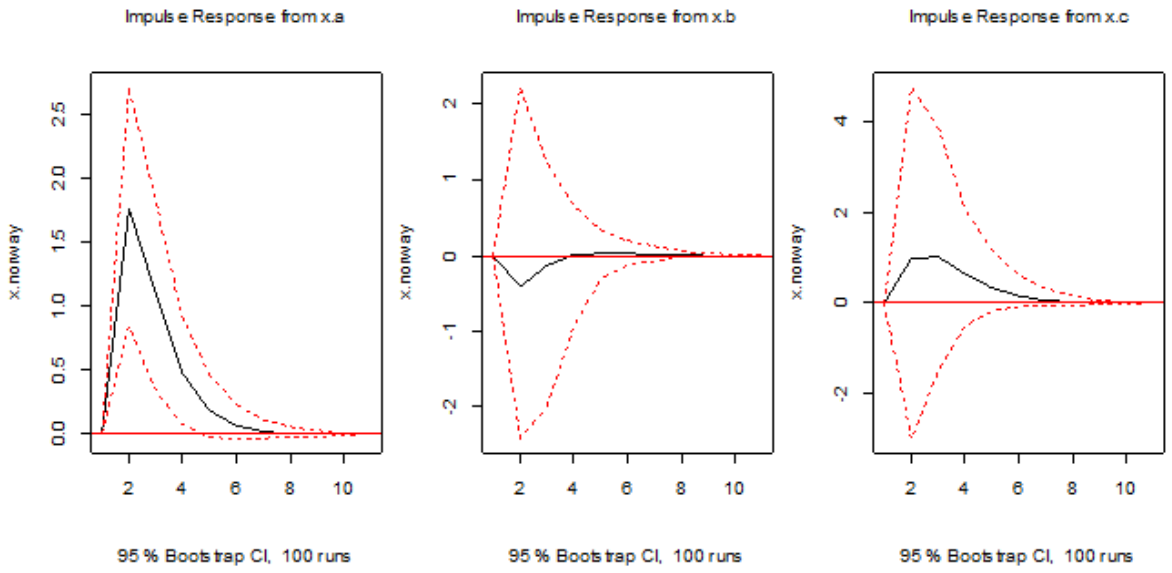

図 13 フレッシュ養殖輸入鮭鱒へのインパルス応答関数 a=fishmeal b=soybean c=wheat

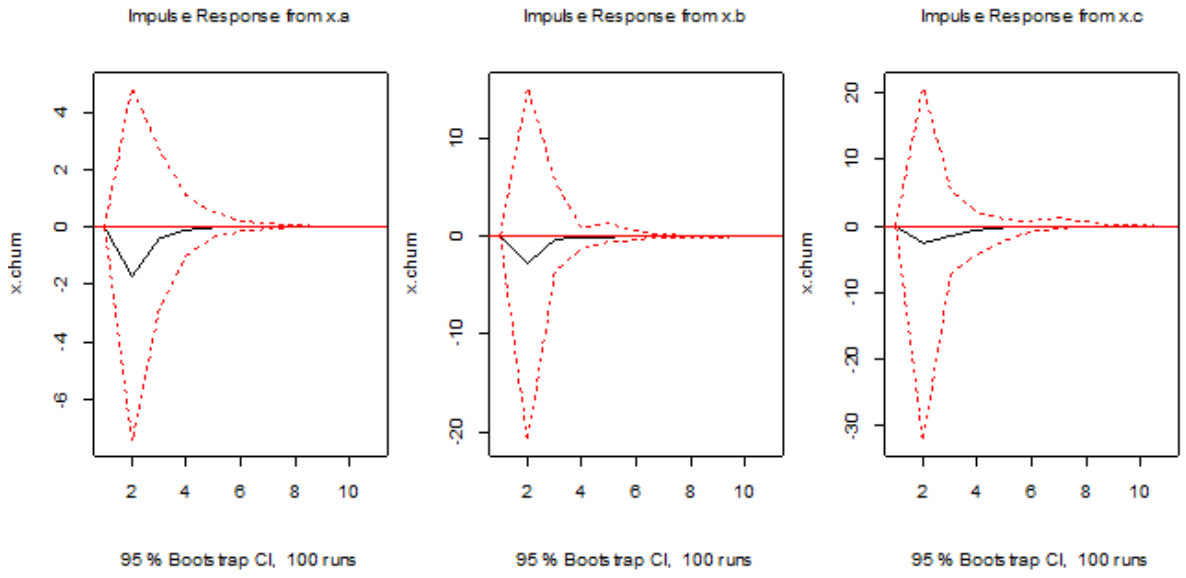

図 14 フレッシュ天然国産白へのインパルス応答関数 a=fishmeal b=soybean c=wheat

（６）　共和分検定分析

魚価と餌の主要素の間に一定の関係で推移するのか否かを調べるために、共和分検定を行った。

４種類の鮭鱒と餌の主要な３つの成分との間に共和分の関係があるか否かを単年度データで時系列に検定していったところ、それぞれ共和分の関係がある時期と無い時期が混ざり合って、特に傾向や関係は確認できなかった。

続いて、続いては冷凍鮭鱒全体、フレッシュ鮭鱒全体、鮭鱒全体にて共和分の関係を検定した。

表 11　共和分検定結果（単年度、単魚種）

共和分検定（単年度）	チリ産銀鮭（冷凍、養殖）	ロシア産紅鮭（冷凍、天然）	輸入サーモン（養殖、生）	国産白鮭（天然、生）
2002 年	○	○	○	○
2003 年	×	○	○	○
2004 年	○	×	×	○
2005 年	○	○	×	×
2006 年	○	○	×	○
2007 年	○	○	×	×
2008 年	×	○	×	×
2009 年	×	○	○	○
2010 年	×	○	○	○
2011 年	○	○	○	○
2012 年	○	○	×	○
2013 年	○	×	×	○
2014 年	○	○	×	○
2015 年	○	○	○	○
2016 年	○	○	○	○

冷凍鮭鱒全体と餌の間には、共和分の関係がないという対立仮説は棄却されなかったため、共和分関係ではないことが確認できた。一方で、フレッシュ鮭鱒全体においては共和分関係がないという対立仮説が棄却されたので共和分関係があり、共和分ランクは１であることが確認できた。

上記に加えて、冷凍及びフレッシュ鮭鱒全体と餌の間には共和分の関係がないという対立仮説は棄却されたので、共和分の関係があるということになる、共和分ランクは１であることが確認できた。

表 12 Johansen trace-tests 共和分検定結果（2002～2016 年全体）

冷凍鮭鱒全体					生鮭鱒全体				
Johansen trace-tests					Johansen trace-tests				
	test	10pct	5pct	1pct		test	10pct	5pct	1pct
r <= 3	1.91	6.5	8.18	11.65	r <= 3	1.62	6.5	8.18	11.65
r <= 2	6.25	12.91	14.9	19.19	r <= 2	7.62	12.91	14.9	19.19
r <= 1	12.67	18.9	21.07	25.75	r <= 1	12.2	18.9	21.07	25.75
r = 0	14.96	24.78	27.14	32.14	r = 0	37.87	24.78	27.14	32.14
鮭鱒全体									
Johansen trace-tests									
	test	10pct	5pct	1pct					
r <= 4	1.86	6.5	8.18	11.65					
r <= 3	6.3	12.91	14.9	19.19					
r <= 2	11.81	18.9	21.07	25.75					
r <= 1	15.46	24.78	27.14	32.14					
r = 0	45.4	30.84	33.32	38.78					

Johansen trace-tests	冷凍　鮭鱒全体	フレッシュ 鮭鱒全体	冷凍、フレッシュ 鮭鱒全体
2002～2016 年	×	○ ランク 1	○ ランク 1

上記分析から期間を長くとっていくと日本の鮭鱒全体と餌の主要３要素の間には一定の関係が見られることから、ある一定の関係を持って推移してきたことが推測される。一方でごく短期の期間にての試行や、各魚種を個別にみていくと、必ずしもそうとも言えないことが確認できた。

（７）　予測誤差分散分解分析

回帰分析や相関関係において養殖、フレッシュ鮭鱒において餌の主要３要素との関係が高いという傾向が見とれたが、予測誤差分散分解分析でも同様の傾向が見て取れるのかを確認するために比較を行った。

各種鮭鱒の餌の成分３種類（fishmeal,soybean,wheat）における予測誤差分散分解の結果を以下に示す。

各魚価の将来の不確実性における、餌の主要３要素の寄与度を比較した結果としては、養殖かつフレッシュ鮭鱒において他の魚種と相対的に分散における餌の寄与度が大きく、一方で天然かつフレッシュ鮭鱒においては逆に寄与度が低いことが確認できた。

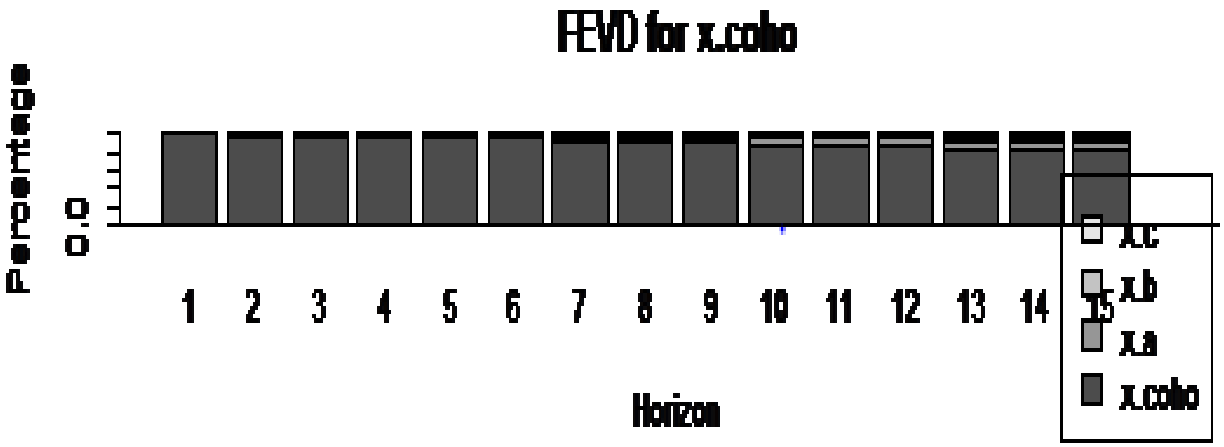

図 15　冷凍養殖チリ産銀鮭への分散分解　a=fishmeal b=soybean c=wheat

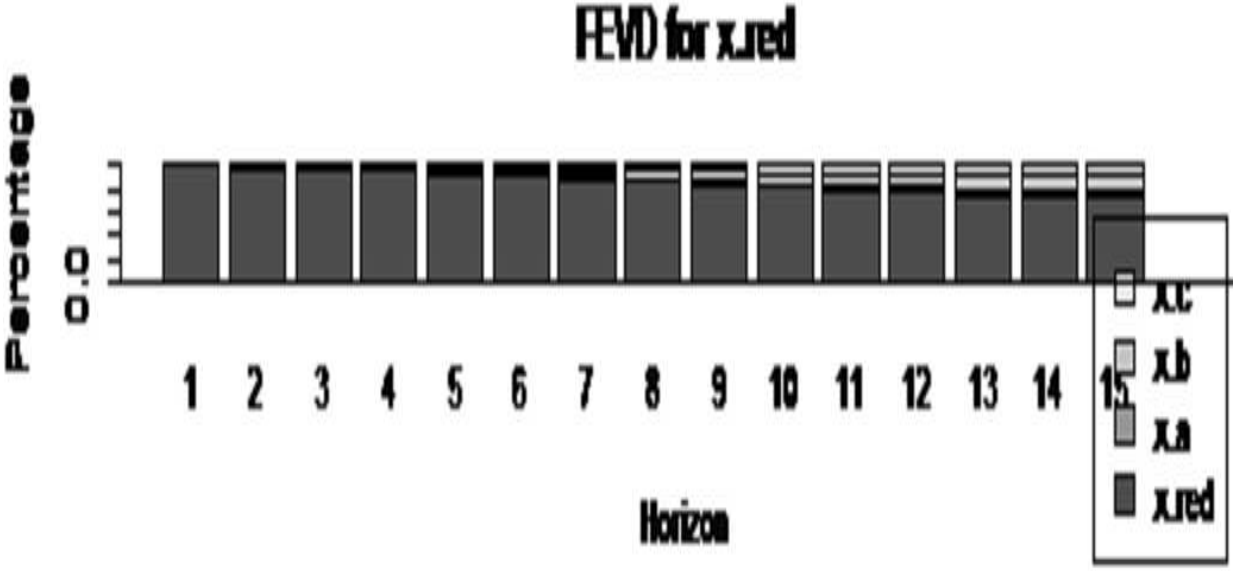

図 16　冷凍天然ロシア産紅鮭への分散分解　a=fishmeal b=soybean c=wheat

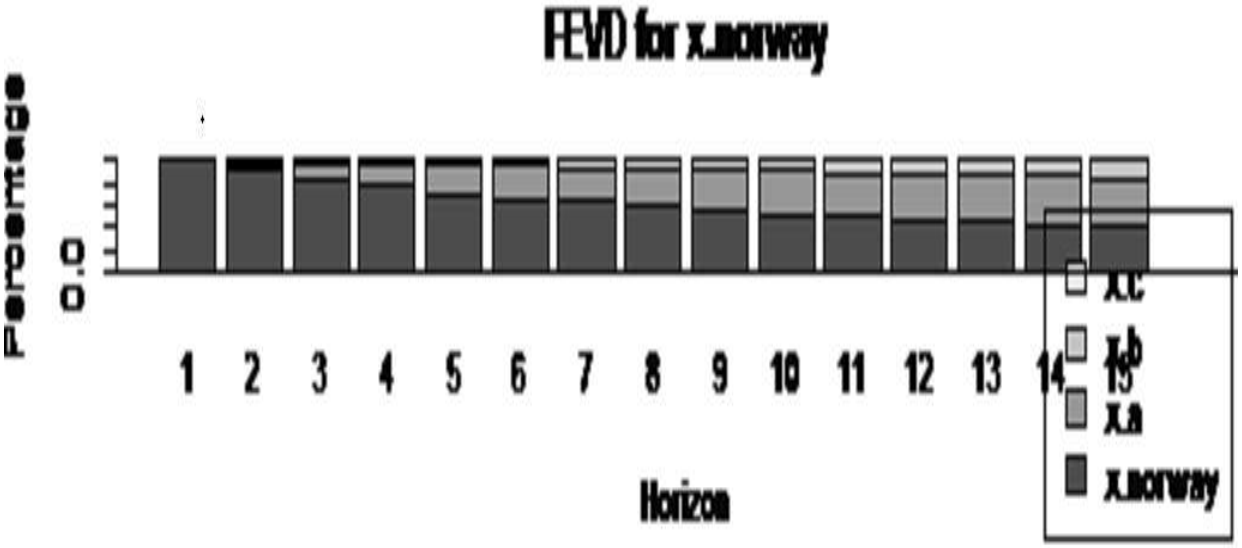

図 17　フレッシュ養殖輸入鮭鱒への分散分解　a=fishmeal b=soybean c=wheat

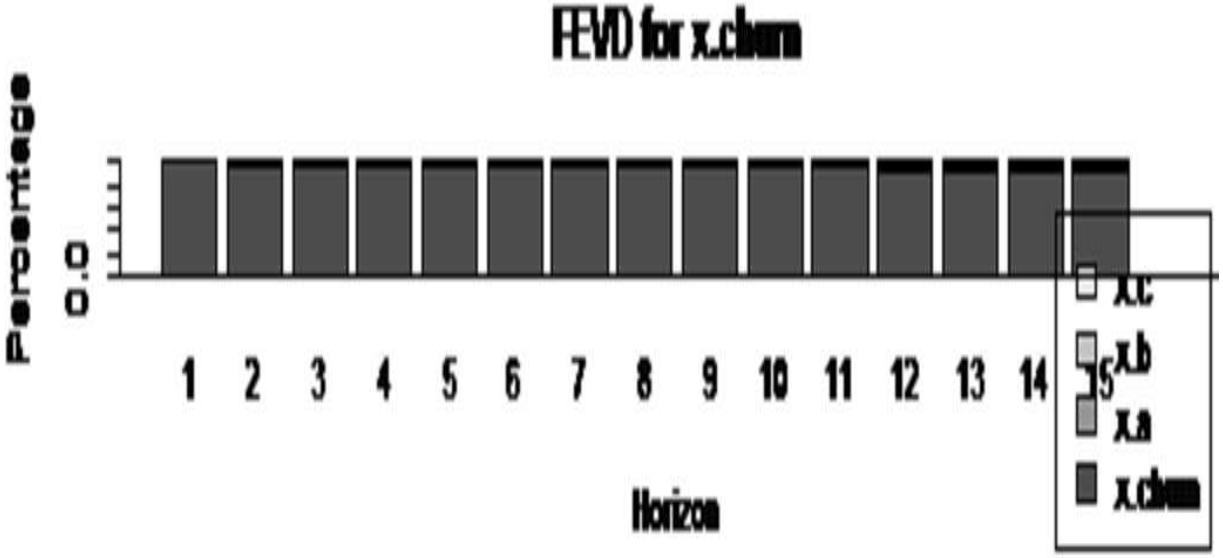

図 18　フレッシュ天然国産白鮭への分散分解　a=fishmeal b=soybean c=wheat

６－２ミクロ編：コンピュータシミュレーション分析
エージェントベースモデルにおける、人口水産市場モデルについての考察

まず５節の手法にあるように、各プレーヤーが操業しているモデルを作成する。各プレーヤーは互いに相互関係を持ち影響しあっていることと、漁獲高等にランダムの要素を盛り込んでいるので、各時系列時々において赤字を出す業者もいれば、利益を出す業者もいる。このような過程をもとに、いくつかの外部環境の変化をプログラムして、どういった差異が見られるかを検証する。

（１） 卸内のバランスシートにマイナスを与える場合（築地卸売業者の豊洲移転イベントをイメージ）

シミュレーションの途中のある一点において、卸売にキャッシュアウトのショックを与えた場合のモデル全体へ波及する影響を考察したい。

債務超過になっても卸売を操業停止させなかった場合においては各プレーヤーの間に相互作用は機能せずに、主だった変化は確認できなかった。一方で債務超過に陥った卸売を操業停止させた場合においては、相互作用のバランスが崩れショックを与えた前後で変化が確認できた。

「債務超過操業停止無し」

まず、各卸売業者が通常通り営業活動をして、純資産がたまっていく様子が確認できる。赤字になっている業者もいれば、黒字になっている業者もいることが確認できる。

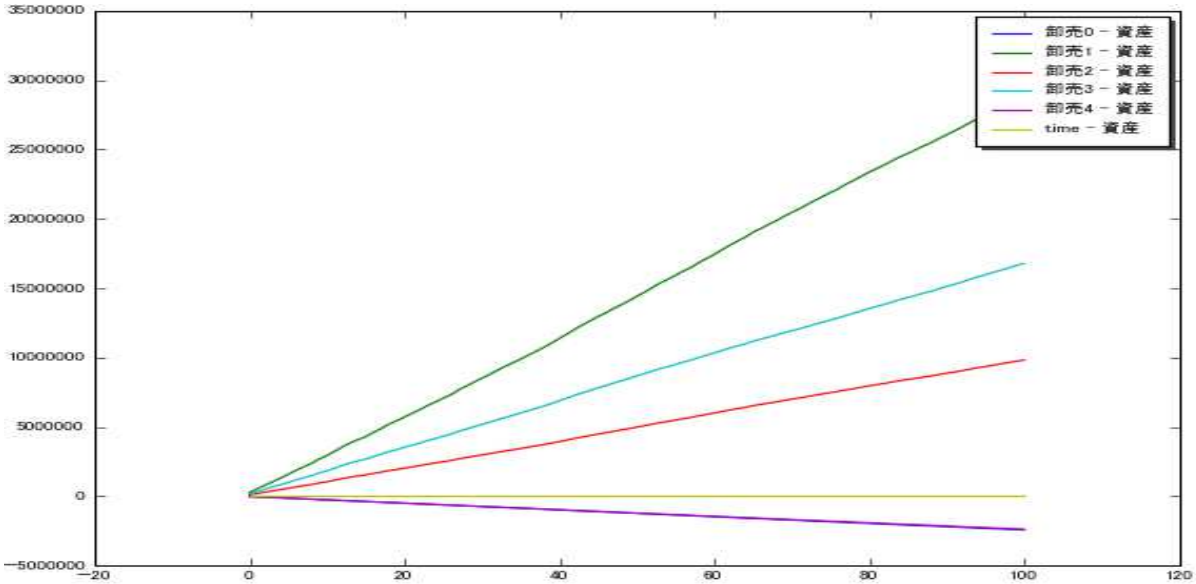

図 19　卸売業者の純資本の推移
横軸はシミュレーション期間、縦軸は卸売業者純資本である。

次に、期間の途中で順資本に大きなマイナスのショックを与えることにする。結果としては、卸売が債務超過になっても操業を継続するので、各業者間の相互関係に変化は起こらず、各卸売の純資本の傾きの変化は確認できなかった。そして、同様に生産者と量販店の純資産の増加率にも変化は確認できなかった。

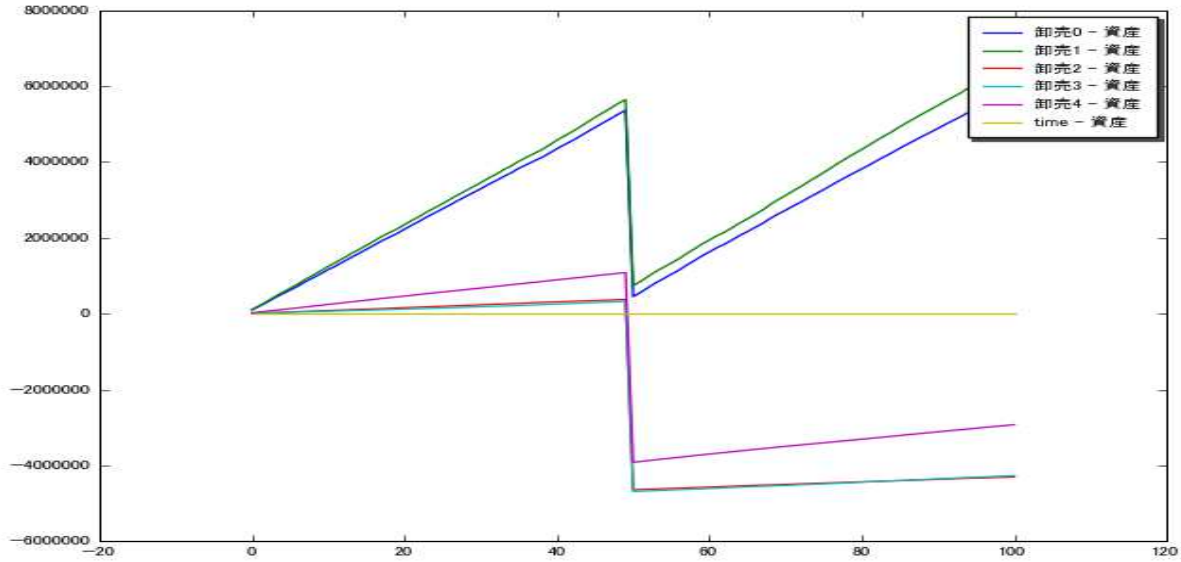

図 20　卸売業者の純資本の推移
横軸はシミュレーション期間、縦軸は卸売純資本である。

「債務超過操業停止有り」
卸売が債務超過になると操業を停止するので、エージェントの相互関係によりバランスに変化が起こる。残存した卸売業者の純資本増加率が増加していることがわかる。
一方で生産者と量販店の純資本増加率はする様子が確認できた。

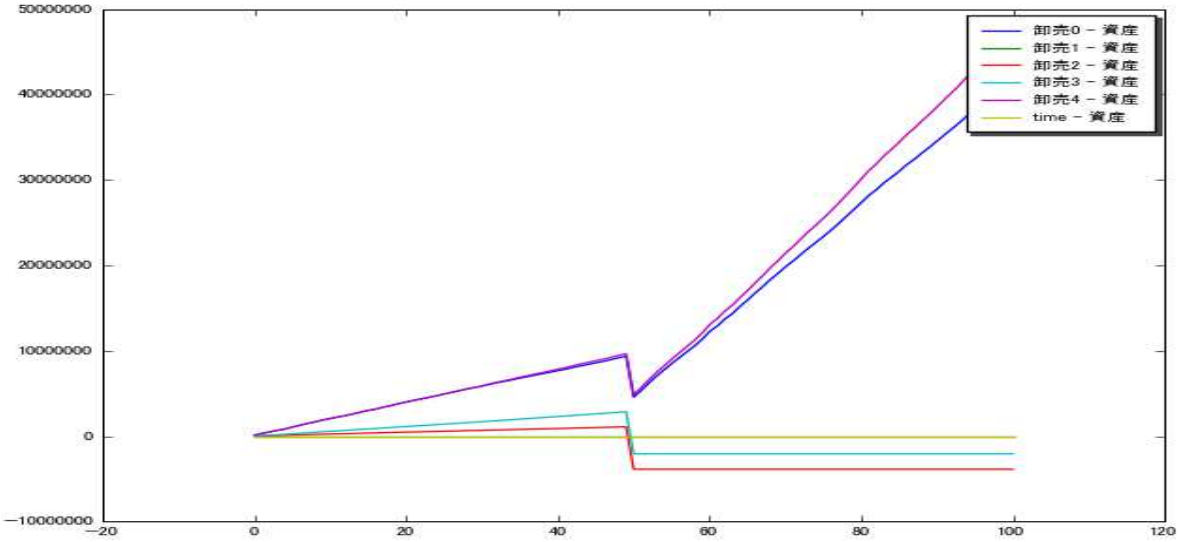

図 21　卸売業者の純資本の推移

横軸はシミュレーション期間、縦軸は生産者純資本である。

操業停止している卸売をよそに、残存した卸売の順資本の伸び率が増加していることが確認できた。

（２） 漁獲枠の設定方法の国際的な違いによる比較

漁業管理制度に関しては下記の図の通りではあるが大きく分けて二つの種類が存在している。まず、ＴＡＣという総漁獲量を資源の維持が可能な数量にて毎年設定するのは共通である。ここからが異なるのだが、一つ目は自由競争の中で漁獲高がＴＡＣに到達するまで漁獲する方法であるオリンピック方式である。もう一つはＴＡＣを個別の企業ごとに割り当てるＩＱ方式である。ＩＴＱ方式というのは、ＩＱ方式のにおいて各企業の漁獲割り当て高を企業間で融通することを可能にした方式である。

オリンピック方式及び、ＩＱ方式を再現し、それぞれの方式で各５回ずつシミュレーションを行った。

図表3 主要国における漁業管理制度の概要

	TAC設定	TAC管理手法		
		IQ方式	ITQ方式	オリンピック方式
アイスランド	●		●	
ノルウェー	●		●	
イギリス	●	●		
スペイン	●	●		
ニュージーランド	●		●	
オーストラリア	●		●	
アメリカ	●		●	
日本	●			●

- **TAC（Total Allowable Catch：総漁獲可能量）** 魚種ごとに漁獲できる総量を定めることにより資源の維持または回復を図ろうとするもの。この総量は、その年の資源量によって毎年変更される。
- **IQ方式（Individual Quota：個別割当方式）** TACを漁業者、漁業団体または漁船ごとに配分し、分与する方式。
- **ITQ方式（Individual Transferable Quota：譲渡可能個別割当方式）** IQ方式のうち、分与された該当量を他の漁業者にも譲渡できるように措置する方式。
- **オリンピック方式** 自由競争の中で関係漁業者の漁獲を認め、漁獲量がTACに達した時点で採捕を停止させる方式。

※ノルウェーではIVQ方式（Individual Vessel Quota：漁船別漁獲割当方式）

資料：日本経済調査協議会・水産業改革高木委員会より

図 22　主要国における漁業管理制度の概要　出典　日本水産㈱ホームページ

結果としては、オリンピック方式においては漁獲高のランダム数の出方に左右され、価格が上下を大きく繰り返すことになった、その結果として、卸売業者の純利益も大きな幅で上下を繰り返した。卸売業者の純利益の推移を下記に５回分示しているが、各試行ごとにおおきくばらつく結果となった。

一方でＩＱ（ＩＴＱ）方式においては、漁獲枠において各自の割り当て分に関しては自由競争の要素がないので、漁獲高が多く価格が値崩れしてる際には無理して漁獲をしないので、値崩れが継続しにくく、高価格帯で安定して推移する様子が確認できた。下記に卸売業者の純利益の推移の５回分の結果を示しているが、安定して純利益が積みあがっていく様子が見て取れると同時に、試行ごとのばらつきもオリンピック方式と比較して非常に少ない様子が確認できた。

以下の図群においては横軸がシミュレーション期間、縦軸が卸売業者順資本である。

「人工オリンピック方式モデル試行結果　５回分」

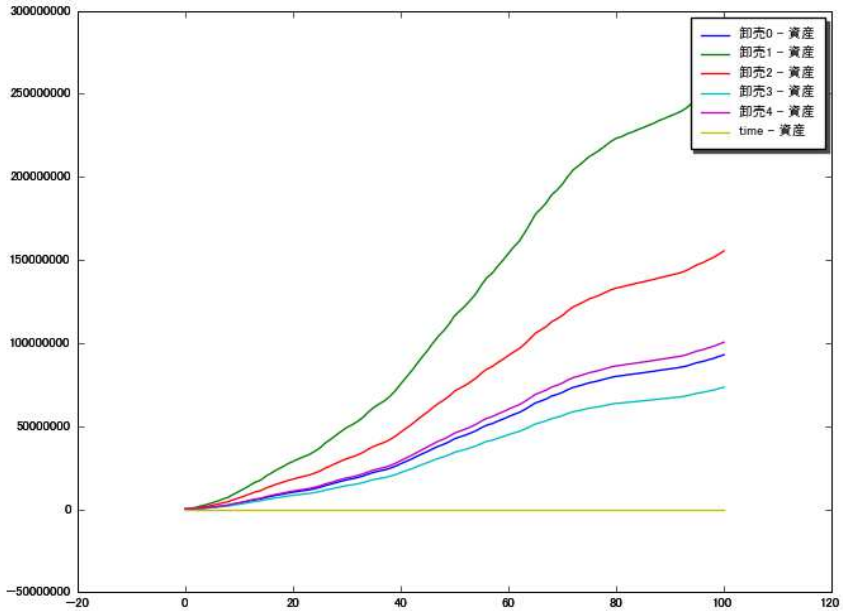

図 23　オリンピック方式における卸売業者の資本推移（１回目）

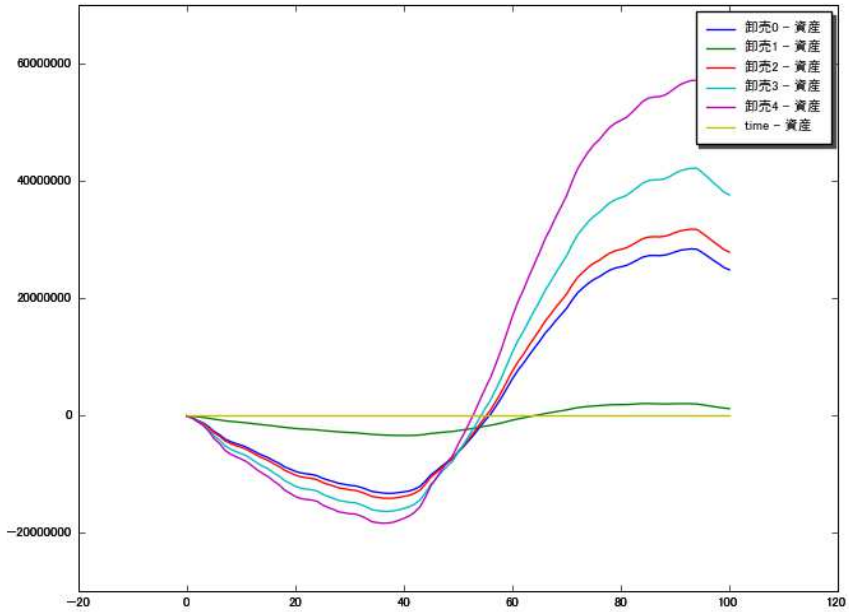

図 24　オリンピック方式における卸売業者の資本推移（２回目）

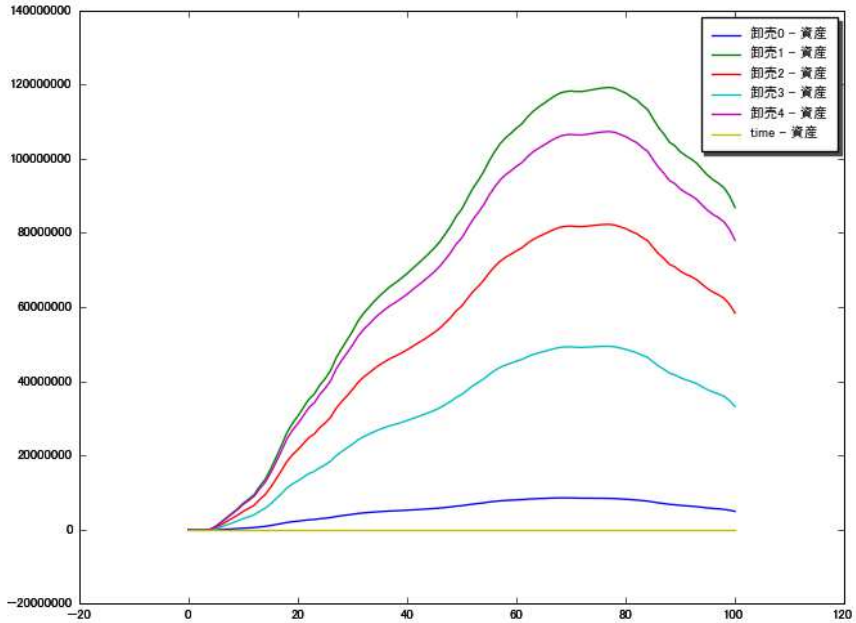

図 25　オリンピック方式における卸売業者の資本推移（３回目）

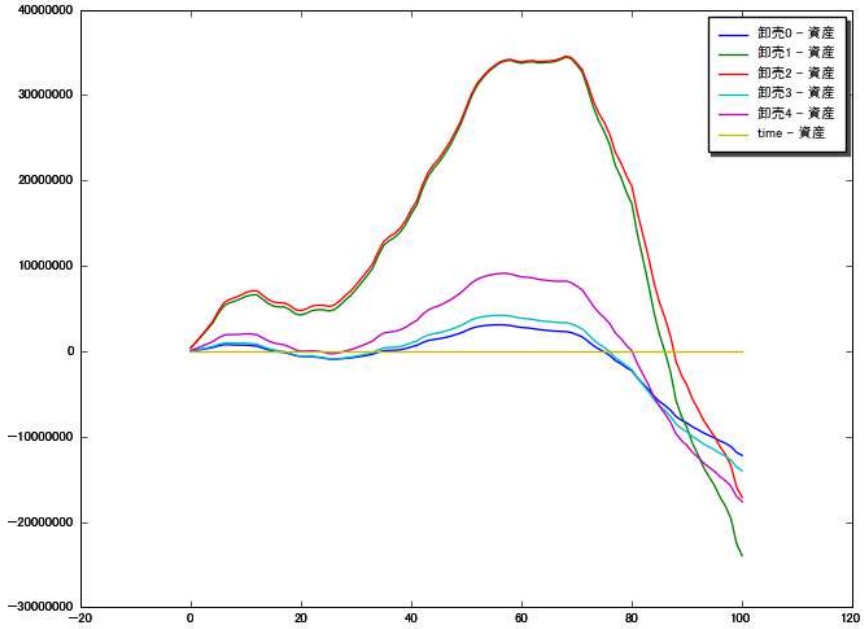

図 26　オリンピック方式における卸売業者の資本推移（４回目）

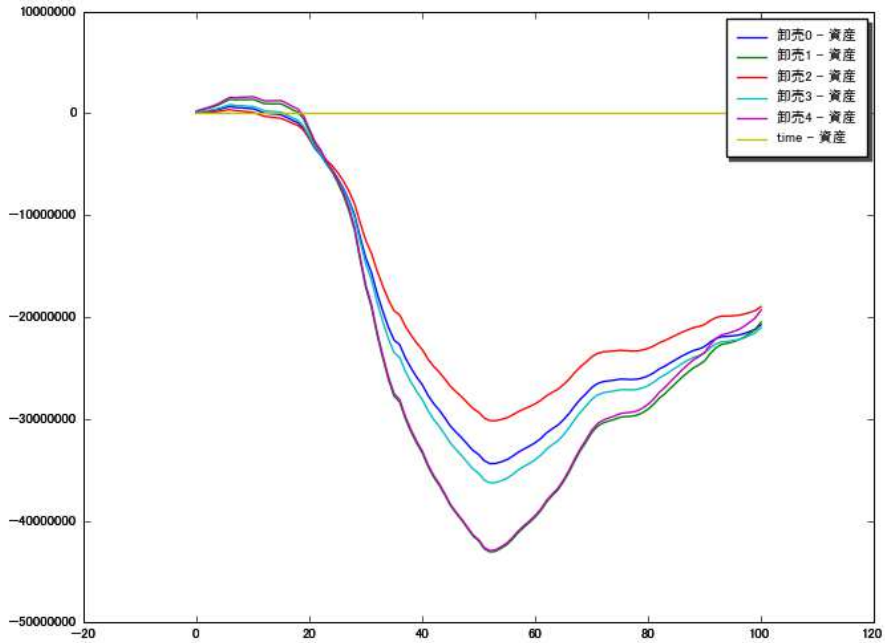

図 27　オリンピック方式における卸売業者の資本推移（５回目）

「人工ＩＱ（ＩＴＱ）モデル方式結果　５回分」

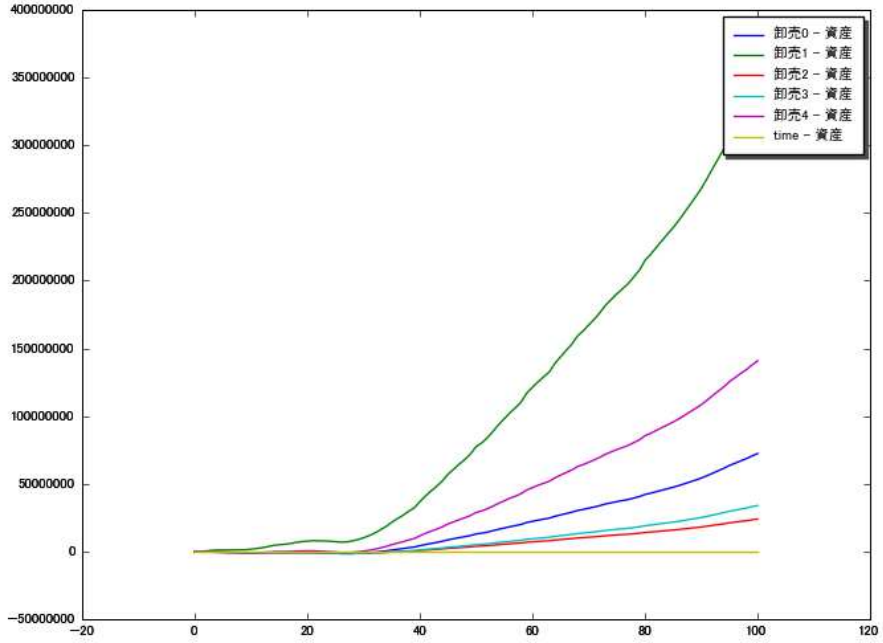

図 28　ＩＱ方式における卸売業者の資本推移（１回目）

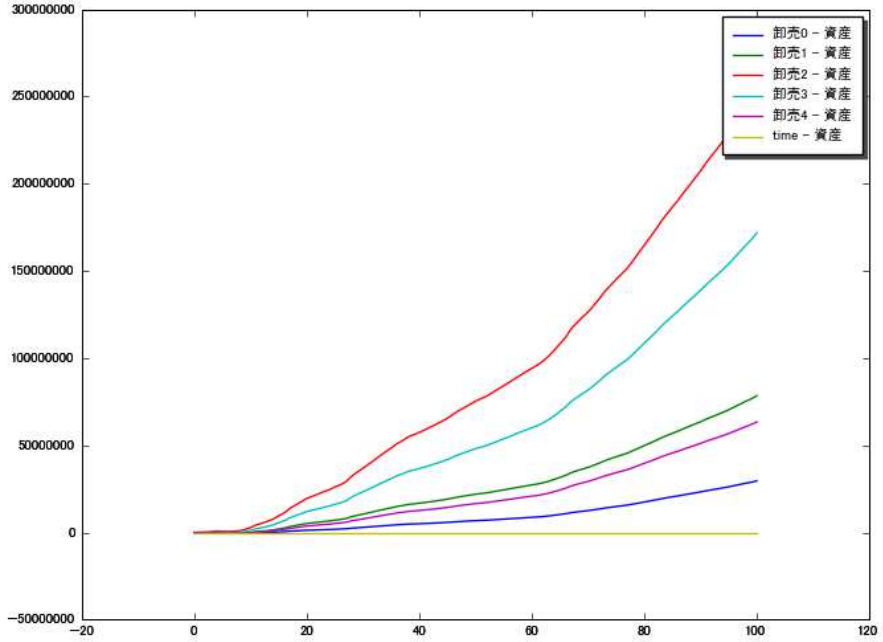

図 29　ＩＱ方式における卸売業者の資本推移（２回目）

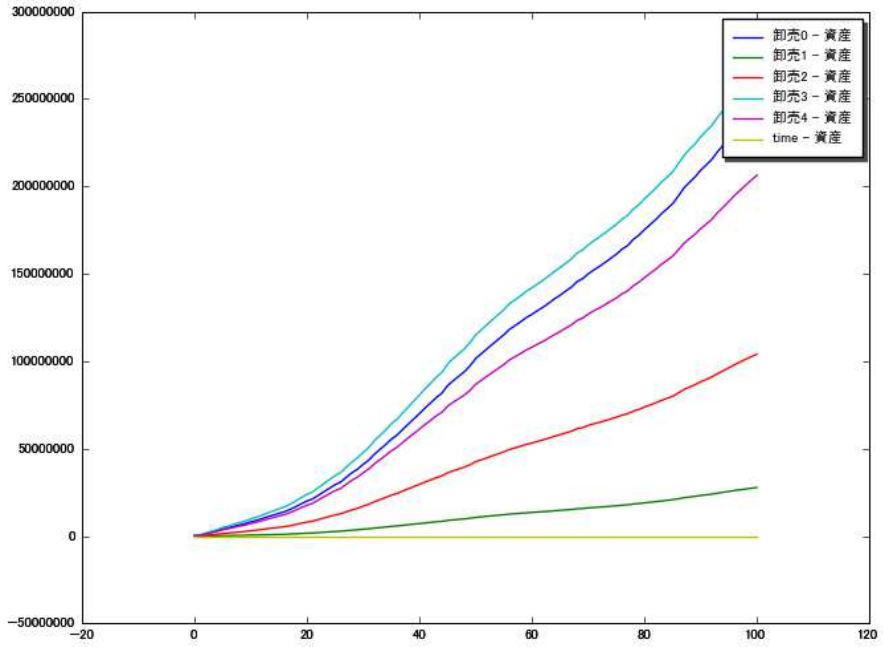

図 30　ＩＱ方式における卸売業者の資本推移（３回目）

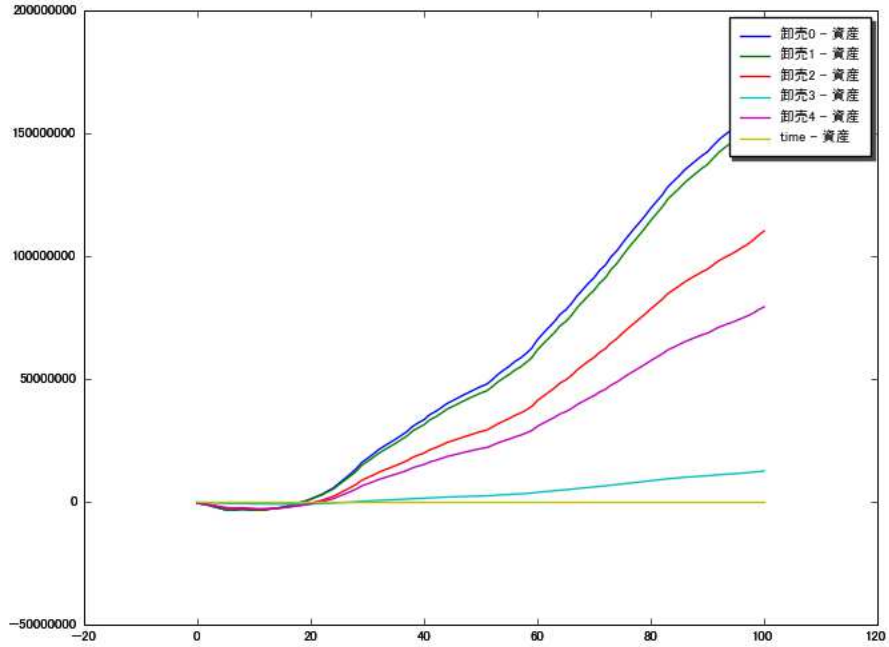

図 31　ＩＱ方式における卸売業者の資本推移（４回目）

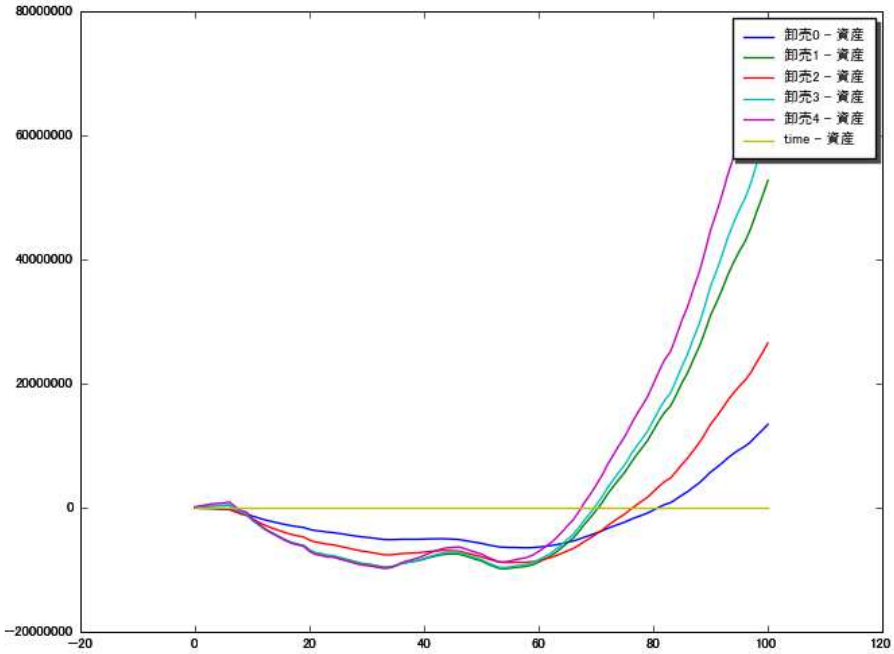

図 32　ＩＱ方式における卸売業者の資本推移（５回目）

７まとめと考察

期間全体でみると、冷凍及びフレッシュ全体の魚価と、餌の主成分３要素との間には、共和分の関係が確認できた。このことから、水産物買付の際には各魚種単体の価格ではなく、セグメント横断的に全体の魚価を考慮すべきであるのかもしれない。

各種４魚種のセグメントごとに、特徴が分かれ、養殖、生鮭鱒は Fishmeal、天然、冷凍鮭鱒は soybean との関係がグレンジャー因果関係見られたが、それぞれインパルス応答係数の波形は異なっており、養殖鮭鱒と天然鮭鱒間の餌から受ける影響に差があることが推測できた。さらに、冷凍鮭鱒には国内在庫の影響を受ける様子が確認できた。

先行研究にてノルウエー鮭鱒において 1985 年～2000 年頃にかけて魚価と餌の関係が増加傾向にあるとのことであったが、本研究において、日本市場において魚価と餌の関係においては、全般的に時系列の変化は見られなかった。本研究の時間軸が 2002 年以降であることから、整合的であるといえると考える。

本文では円ベースでの分析をしたが、別途＄ベースで同様の分析を行ったが全体的に同様の傾向が確認できた。商社にとって為替は非常に重要なファクターであり、私自信営業マンとしてもお客に相場を説明する際によく説明するポイントであったのだが、本分析においてはあまり影響がなかったことは個人的には以外であった。

水産市場において債務超過に陥った業者が操業を停止するかしないかが、各プレーヤーの相対的力関係のバランスの変化の有無に影響を与える様子が確認できた。競争

力の低い業者の安直な延命措置は市場の機能を妨げるのかもしれないことが推測されるので、痛みを和らげながら卸売等のプレーヤー統廃業、転業を促すような策こそが大切なのかもしれない。

また、漁業資源管理方式のオリンピック方式とＩＱ（ＩＴＱ）方式を再現した結果、一般的に水産経済学で言われているようにオリンピック方式に方が魚価の値崩れもしやすく、卸売業者の純利益も安定しなかった。日本の漁獲方式もノルウエーなどに習いＩＱ（ＩＴＱ）方式を導入した方が、各プレーヤーも利益が出やすくなり、水産業界全体の繁栄に寄与するであろうことを、視覚的に表現することができた。

また、今回は一魚種と仮定したモデルではあったが、魚種や流通形態ごとに市場経由率は大きく異なるので（一般的に冷凍物は市場外流通が低く、フレッシュ物は市場経由率が高い）このファクターの違いも統計的分析、時系列分析においての各魚種４種の性質の差に寄与しているのかもしれない。

８今後の課題

天然鮭鱒についての今回示した結果は、これまで行われておらず、その意味で本研究の意義は大きい。本分析では、時系列計量分析において、天然物についての季節変動や、フレッシュ鮭鱒の流動性等は今回の分析には考慮されていない。これらを考慮した分析は今後の課題として挙げられる。

また、養殖鮭鱒の方が餌の主要素の影響が大きく、その中でも Fishmeal の影響が大きいという点は先行研究及び仮説と整合的な結果であったが、冷凍天然鮭鱒と Soybean との関連については、分析の余地がある。これらの点も今後の課題として挙げられる。

水産市場シミュレーション分析において、現実の側面を取り込んだ分析を実施しているが、考慮する要因を拡張した分析は、今後の課題として挙げられる。

また、今回はマクロ的なアプローチとミクロ的なアプローチを独立した２つの分析として取り扱ったが、この２つの分析を統合していくようなことも今後の課題として挙げておきたい。

９謝辞

本分析を行うにあたり様々な方々のお力を拝借させて頂きました。我らが高橋教授、Ｍ３８高橋ゼミの皆様、Ｍ３５の五島様、㈱ＮＴＴデータ数理システムの嶋田様、山本様、国立研究開発法人産業技術総合研究所の宮下様。皆様にはついつい惰性に陥ってしまいがちな私に刺激と目標及び良い知恵と環境を提供して頂きました、皆様のお助けなしでは、より貧弱で視野の狭い研究となっていたことは疑いありません、この場をお借りしてお礼を申し上げます。ありがとうございました、感謝します。

１０参考文献

Frank Asche,Atle Oglend The relationship between input-factor and output prices in commodity industries The case of Norwegian salmon aquaculture　Journal of commodity markets 2016

Kazuhiko Ohashi,Tatsuyoshi Okimoto Increasing trends in the excess comovement of comm odity prices Journal of commodity markets 2016

清水幾太郎　生鮮サケ類の産地価格形成要因に関わる 輸入量と在庫量の影響さけ・ます資源管理センター 研究報告　第 6 号　1-11 頁 2004 年 3 月

清水幾太郎　サケの産地価格にみられる短期および長期の変動要因　さけ・ます資源管理センター 研究報告　第 5 号　13-19 頁 2002 年 12 月

“Alan P.Kirman,Nicolass J.Vriend” ”Learning to be loyal. A study of the Marseille Fish Market” Interaction and Market structure 2000

東洋経済オンライン　三菱商事、鮭赤字で資源安でとダブルパンチ 2015 年 9 月 12 日 http://toyokeizai.net/articles/-/83201

日本水産㈱ホームページ　http://www.nissui.co.jp/academy/data/04/

池尾和人・大野早，編著，コモディティテイ市場と投資戦略「金融市場化」の検証，東京，勁草書房，2014，315p

沖本竜義　経済・ファイナンスデータの為の計量時系列分析，朝倉書店，199p

J.D ハミルトン（著）沖本竜義,井上智夫（翻訳）時系列解析（上）定常課程編,シーエーピー出版,477p

J.D ハミルトン（著）沖本竜義,井上智夫（翻訳）時系列解析（下）非定常/応用定常課程編,シーエーピー出版,444p

国立研究開発法人　水産研究・教育機構　北海道区水産研究所　ホームページ http://hnf.fra.affrc.go.jp/index.html

東京都卸売市場統計情報　ホームページ http://www.shijou-tokei.metro.tokyo.jp/

水産庁統計情報　ホームページ
http://www.jfa.maff.go.jp/j/kikaku/toukei/

world bank commodity markets home page
http://www.worldbank.org/en/research/commodity-markets

１１Appendix

Ａ－１　冷凍鮭鱒とフレッシュ鮭鱒のラグについて

回帰分析においての分析結果にて、冷凍鮭鱒は在庫の影響を受けるがフレッシュ鮭鱒は影響を受けにくいという結果が確認できたわけだが、分析に用いたデータは月次データであった。冷凍鮭鱒とフレッシュ鮭鱒において、在庫から受ける影響にラグがあるのではないかと思い、１２か月移動平均のデータで回帰分析を行った結果、フレッシュ鮭鱒全体においても、在庫の影響が優位水準を満たした、Ｒ２は在庫を説明変数に加える前からかなり高かったもののわずかに上昇した。

VAR 解析においては、AIC が最小になるようにラグを決定してるので、ラグの要素が考慮されていると思われるが、その結果から示唆を得るまでには至らなかったので、VAR モデルのラグの考察も今後の課題として追加しておきたい。

表 13　重回帰分析結果（12 か月移動平均）魚価と餌の主要素とトレンド

冷凍養殖チリ銀鮭				冷凍天然ロシア紅鮭			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	385.14	12.55	2e-16 ***	Const	440.88	14.90	2e-16 ***
Trend	-0.02	-0.05	0.96	Trend	0.13	0.40	0.69
Fishmeal	1.98	6.29	2.82e-09 ***	Fishmeal	0.94	3.09	0.00233 **
Soybean	1.89	1.21	0.23	Soybean	11.73	7.74	1.01e-12 ***
Wheat	-6.69	-3.44	0.000748 ***	Wheat	-12.19	-6.49	1.00e-09 ***
R2	0.52			R2	0.75		
F-test	44.51			F-test	120.10		
フレッシュ養殖輸入鮭鱒				フレッシュ国産白鮭			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	403.23	11.97	2e-16 ***	Const	660.00	17.81	2e-16 ***
Trend	0.00	-0.01	1.00	Trend	0.60	1.50	0.13
Fishmeal	3.39	9.79	2e-16 ***	Fishmeal	2.52	6.62	4.93e-10 ***
Soybean	8.64	5.01	1.43e-06 ***	Soybean	4.13	-2.18	0.0311 *
Wheat	-7.37	-3.44	0.00073 ***	Wheat	2.91	-1.24	0.22
R2	0.85			R2	0.50		
F-test	232.40			F-test	41.00		
冷凍鮭鱒全体				フレッシュ鮭鱒全体			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	464.12	11.59	2e-16 ***	Const	437.91	14.54	2e-16 ***
Trend	-0.34	-0.79	0.43	Trend	0.79	2.42	0.01645 *
Fishmeal	2.01	4.88	2.48e-06 ***	Fishmeal	2.83	9.13	2.6e-16 ***
Soybean	1.82	0.89	0.38	Soybean	5.04	3.27	0.00132 **
Wheat	-5.81	-2.29	0.0236 *	Wheat	-6.25	-3.27	0.00131 **
R2	0.32			R2	0.86		
F-test	18.77			F-test	245.50		

表 14　重回帰分析結果（12 か月移動平均）魚価と餌の主要素とトレンド、国内在庫

冷凍養殖チリ銀鮭				冷凍天然ロシア紅鮭			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	871.70	14.24	2e-16 ***	Const	769.90	14.53	2e-16 ***
Trend	2.77	6.37	2.18e-09 ***	Trend	1.97	5.24	5.32e-07 ***
Fishmeal	1.70	6.31	2.95e-09 ***	Fishmeal	0.95	4.07	7.66e-05 ***
Soybean	-5.33	-3.51	0.000598 ***	Soybean	5.53	4.20	4.50e-05 ***
Wheat	0.38	0.21	0.84	Wheat	-3.49	-2.18	0.031226 *
stock1	0.00	-4.54	1.16e-05 ***	stock1	0.00	-9.28	2e-16 ***
stock2	-0.03	-7.42	8.28e-12 ***	stock2	-0.01	-3.68	0.000332 ***
R2	0.6851,			R2	0.87		
F-test	54.03			F-test	166.20		
フレッシュ養殖輸入鮭鱒				フレッシュ国産白鮭			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	1083.00	19.03	2e-16 ***	Const	864.30	9.57	2e-16 ***
Trend	4.00	9.91	2e-16 ***	Trend	2.03	3.17	0.00187 **
Fishmeal	2.80	11.19	2e-16 ***	Fishmeal	2.32	5.85	2.96e-08 ***
Soybean	-0.55	-0.39	0.70	Soybean	-7.26	-3.24	0.00148 **
Wheat	-0.22	-0.13	0.90	Wheat	0.34	-0.13	0.90
stock1	0.00	-2.89	0.00443 **	stock1	0.00	-0.59	0.56
stock2	-0.05	-12.66	2e-16 ***	stock2	-0.02	-2.56	0.01154 *
R2	0.9221,			R2	0.5246,		
F-test	294.10			F-test	27.40		
冷凍鮭鱒全体				フレッシュ鮭鱒全体			
	Coef	t-Statistic	p-Value		Coef	t-Statistic	p-Value
Const	899.80	15.69	2e-16 ***	Const	976.28	17.08	2e-16 ***
Trend	3.33	8.18	1.13e-13 ***	Trend	3.97	9.79	2e-16 ***
Fishmeal	1.45	5.77	4.47e-08 ***	Fishmeal	2.39	9.50	2e-16 ***
Soybean	-5.81	-4.08	7.34e-05 ***	Soybean	-2.43	-1.71	0.08875 .
Wheat	1.35	0.78	0.44	Wheat	0.09	-0.05	0.96
stock1	0.00	-3.11	0.00228 **	stock1	0.00	-2.95	0.00372 **
stock2	-0.03	-8.82	2.88e-15 ***	stock2	-0.04	-9.77	2e-16 ***
R2	0.7481,			R2	0.9071,		
F-test	73.77			F-test	242.50		

Ａ－２　本章では時系列分析で用いたＲのコードを以下に添付する。

```
#主成分分析
Prcomp()

#ADF unit-root test
PP.test()

#VAR モデル
VAR(){vars}

#VECM モデル
vecvar(){vars}
```

```
#グレンジャー因果性検定
causality(){vars}

#インパルス応答関数
irf(){vars}

#Johansen trace-tests
ca.jo(){urca}

#予測誤差分散分解
fevd(){vars}
```

Ａ－３　本章ではシミュレーション分析で用いたスクリプトを以下に添付する。以下は卸売が債務超過の際に操業停止するモデルである。同様の際に操業停止しないモデルは、下記スクリプトから、#倒産フラグ等のコードを削除することで再現可能である。

```
# -*- Mode: Python; coding: utf-8-dos; -*-

from psim import *
from graph import graphclient
graphclient.GraphServer.createServer(uri='http://127.0.0.1:62758')

initialize()

setGlobalSeed(0)

# カスタムコード
# なし

# シミュレーションパラメータの定義
class SimulationParam:
    def __init__(self,
                 inputDir = u"input¥¥default",
                 outputDir = u"output¥¥default"):
        self.n = constantValue(4).next() # 生産者数
        self.m = constantValue(5).next() # 卸売数
        self.l = constantValue(6).next() # 量販店数
```

```
        self.jishukisei = constantValue(False).next() #

# シミュレーターの定義
class Simulator (SimulatorBase):
    def __init__(self,
                 param = SimulationParam(),
                 inputDir = u"input¥¥default",
                 outputDir = u"output¥¥default",
                 showGraph = True,
                 warm = lambda self: 0.0,
                 until = lambda self: 100.0,
                 odesolver = lambda self: dopri54Solver(dt = 0.1, rtol = 0.001,
atol = 1e-06),
                 dataDir = u"data"):
        SimulatorBase.__init__(self, param, inputDir, outputDir, showGraph,
warm, until, odesolver, dataDir)

        # 資源の定義

        # オブジェクトの生成
        # リンクの生成
        # 環境 の定義

        class Environment (EnvironmentBase):
            def __init__(self, **keys):
                EnvironmentBase.__init__(self, u"環境", **keys)
            def initAfter(self, **keys):
                u"""環境生成後に呼ばれる。"""
                pass
            def initAttribute(self, agent):
                u"""新規のエージェントが作成されたら、必ずこのメソッドが呼ば
れる。

                    エージェントは必ずひとつの環境に属す。

                    各エージェントに環境に固有の属性を設定する。"""
                pass
            def setPosition(self, agent, pos):
```

```
            u"""エージェント agent の環境に固有の位置属性を更新する。

                エージェントの環境上の位置属性が変わったら、必ずこのメソッドが呼ばれる。

                pos は、環境上の位置属性(座標やノード番号など)であり、そのフォーマットは、環境ごとに異なる。"""
            pass
        def getPosition(self, agent):
            u"""エージェント agent の環境に固有の位置属性を返す。

                環境上の位置属性(座標やノード番号など)であり、そのフォーマットは、環境ごとに異なる。"""
            pass
        def getRandomPosition(self):
            u"""環境内のランダムな位置属性を返す。

                環境上の位置属性(座標やノード番号など)であり、そのフォーマットは、環境ごとに異なる。"""
            pass
        def draw(self, panel):
            u"""この環境を panel 上に描画する。"""
            pass
        def drawAgents(self, panel, agents):
            u"""この環境上で、エージェントのリスト agents を panel 上に描画する。"""
            pass
        def custom(self):
            pass

    self.eEnvironment = Environment(simulator = self)

    self.addEnv(self.eEnvironment)
    # 同期エージェント の定義

    class SynchronousAgentSet (SynchronousAgentSetBase):
        def _defAgent(self):
            class _Agent (SynchronousAgentBase):
```

```
            def initAfter(self, *args, **keys):
                u"""エージェントが作成され、environmentBase.initAttribute の呼び出し後に、このメソッドが呼ばれる。"""
                pass
            def step(self):
                u"""エージェントのステップ処理"""
                pass
            def custom(self):
                pass
        return _Agent
    def initAfter(self, **keys):
        self.interval = 1 # ステップ間隔
        self.freezeAddAgent = True # エージェントの追加を次ステップまで凍結する
        self.freezeDelAgent = True # エージェントの削除を次ステップまで凍結する
        u"""エージェント集合の作成後にこのメソッドが呼ばれる。

          エージェント集合固有の初期化を行う。"""
        self.agentFreezeVars = [""] # エージェントの凍結する変数リスト
        # エージェントの生成
        self.generateAgents(0)

        #生産者資本
        self.E2 = numpy.random.uniform(100, 5000, size = param.n)
        # 生産者規模
        self.SP = numpy.random.uniform(100, 5000, size = param.n)
        # 生産者相対仕入れ量
        self.WP = self.SP / self.SP.sum()
        #  生産者相対仕入れ量変動率(volatility)
        self.sigmaP = 0.01

        # 卸売資本
        self.E = numpy.random.uniform(100, 5000, size = param.m)
        # 卸売規模
        self.SW = self.E.copy()
        # 卸売相対仕入れ量
        self.WW = numpy.repeat(self.SW / self.SW.sum(),
```

```
param.n).reshape(param.m, param.n)
                # 　卸売相対仕入れ量変動率(volatility)
                self.sigmaW = 0.01

                # 量販店資本
                self.E3 = numpy.random.uniform(100, 5000, size = param.l)
                # 量販店規模
                self.SM = numpy.random.uniform(30, 1000, size = param.l)
                # 量販店相対仕入れ量
                self.WM = numpy.repeat(self.SM / self.SM.sum(),
param.m).reshape(param.l, param.m)
                # 量販店相対仕入れ量変動率(volatility)
                self.sigmaM = 0.01

                # 卸の倒産フラグ
                self.working = numpy.ones(param.m, dtype = bool)

                #観測モニター生産者
                self.PP_monitor = Monitor(["time"] + [u"生産者%d" % i for i in
xrange(param.n)],
                                          ["i"] + ["f"] * param.n,
                                          name = u"仕入れ価格 1")
                self.PW2_monitor = Monitor(["time"] + [u"生産者%d" % i for i in
xrange(param.n)],
                                          ["i"] + ["f"] * param.n,
                                          name = u"販売価格 2")
                self.E2_monitor = Monitor(["time"] + [u"生産者%d" % i for i in
xrange(param.n)],
                                          ["i"] + ["f"] * param.n,
                                          name = u"資産 2")
                self.simulator.addMonitor(self.PP_monitor)
                self.simulator.addMonitor(self.PW2_monitor)
                self.simulator.addMonitor(self.E2_monitor)

                # 観測モニター
                self.PW_monitor = Monitor(["time"] + [u"卸売%d" % i for i in
xrange(param.m)],
                                          ["i"] + ["f"] * param.m,
```

```
                                        name = u"仕入れ価格")
        self.PM_monitor = Monitor(["time"] + [u"卸売%d" % i for i in
xrange(param.m)],
                                        ["i"] + ["f"] * param.m,
                                        name = u"販売価格")
        self.E_monitor = Monitor(["time"] + [u"卸売%d" % i for i in
xrange(param.m)],
                                        ["i"] + ["f"] * param.m,
                                        name = u"資産")
        self.simulator.addMonitor(self.PW_monitor)
        self.simulator.addMonitor(self.PM_monitor)
        self.simulator.addMonitor(self.E_monitor)

        # 観測モニター量販店
        self.PM2_monitor = Monitor(["time"] + [u"量販店%d" % i for i in
xrange(param.l)],
                                        ["i"] + ["f"] * param.l,
                                        name = u"仕入れ価格 3")
        self.PM3_monitor = Monitor(["time"] + [u"量販店%d" % i for i in
xrange(param.l)],
                                        ["i"] + ["f"] * param.l,
                                        name = u"販売価格 3")
        self.E3_monitor = Monitor(["time"] + [u"量販店%d" % i for i in
xrange(param.l)],
                                        ["i"] + ["f"] * param.l,
                                        name = u"資産 3")
        self.simulator.addMonitor(self.PM2_monitor)
        self.simulator.addMonitor(self.PM3_monitor)
        self.simulator.addMonitor(self.E3_monitor)

        # VS のシグマの記録用
        self.bufferVSsigma = [0,0]

        self.B = 0.1

        self.deltaB=0
    def step(self):
        u"""エージェント集合のステップ処理を行う。"""
```

```
def fPP(VP, VS, SP):
    a = 0.1
    b = 0.1
    c = 0.1
    d = 1000
    return - a * VP - b * VS - c * SP + d

def fPW(VW, VS, SW, SP):
    alpha = 0.1
    beta = 0.1
    gamma = 0.1
    zeta = 0.1
    lmd = 500
    return - alpha * VW - beta * VS - gamma * SW.reshape(param.m,1) + zeta * SP.reshape(1,param.n) + lmd

def fPM(VM, VS, SM, SW, B):
    A = 0.1
    C = 0.1
    D = 0.1
    E = 500
    return - A * VM - B * VS - C * SM.reshape(param.l,1) + D * SW.reshape(1,param.m) + E

def calcDeltaB(sigma1, sigma2):
    # sigma1: 1 期前、sigma2: 2 期前
    # sigma1 と sigma2 が下振れ（下振れが 2 期連続）->高騰->deltaB は減少方向
    # sigma1 と sigma2 が上振れ（上ブレが 2 期連続)->下落->deltaB は増加方向

    # B スケールパラメータ、s:1 期前を 1 として 2 きまえの影響度
    B = 0.5
    s = 0.5

    deltaB = B * (sigma1 + sigma2 * s)
```

```
        return deltaB

    # 漁獲高
    self.VS = numpy.random.normal(1000, 100)

    # 自主規制の効果
    if param.jishukisei == True:
        if self.deltaB > 0:
            if self.VS > 1000:
                self.VS = 1000

    # 生産者
    self.WP *= numpy.exp(self.sigmaP * numpy.random.randn(param.n)) # 幾何ランダムウォーク
    self.VP = self.WP / self.WP.sum() * self.VS
    self.PP = fPP(self.VP, self.VS, self.SP)

    # 卸売
    self.WW *= numpy.exp(self.sigmaW * numpy.random.randn(param.m, param.n))
    self.WW[numpy.logical_not(self.working), :] = 0 # 倒産した卸の仕入れ量は 0
    self.VW = (self.WW / self.WW.sum(axis=0)) * self.VP
    self.PW = fPW(self.VW, self.VS, self.SW, self.SP)

    # 量販店
    self.WM *= numpy.exp(self.sigmaM * numpy.random.randn(param.l, param.m))
    self.WM[:, np.logical_not(self.working)] = 0 # 倒産した卸からの仕入れ量は 0
    self.VM = self.WM / self.WM.sum(axis=1)[:, numpy.newaxis] * self.VW.sum(axis=1)

    #  漁獲高のブレが何σか( -:下振れ->不作, +:上ブレ->豊作)
    sigmaVS = (self.VS-1000)/100

    self.bufferVSsigma.append(sigmaVS)
    self.deltaB =
```

```
calcDeltaB(self.bufferVSsigma[len(self.bufferVSsigma)-1],
self.bufferVSsigma[len(self.bufferVSsigma)-2])
                    self.B = self.B + self.deltaB


                    self.PM = fPM(self.VM, self.VS, self.SM, self.SW, self.B)

                    # 生産者 i の仕入れ費用
                    self.PC = (self.VP * self.PP)
                    # 生産者 i の売り上げ
                    self.PS = (self.VW * self.PW).sum(axis=0)
                    # 生産者 i の損益
                    self.PF = self.PS - self.PC - 20000
                    # 生産者 i の資産
                    self.E2 += self.PF

                    # 卸売 j の仕入れ費用
                    self.WC = (self.VW * self.PW).sum(axis=1)
                    # 卸売 j の売り上げ
                    self.WS = (self.VM * self.PM).sum(axis=0)
                    # 卸売 j の損益(倒産した卸については、以降変化しない)
                    if now() ==50:
                        self.WF=self.WS-self.WC-np.where(self.working,20000,0)
                    else:
                        self.WF=self.WS-self.WC-np.where(self.working,20000,0)
                    # 資産
                    self.E += self.WF

                    # 倒産判断
                    #self.working = self.E >= 0

                    #量販店 z の仕入れ費用
                    self.MC = (self.VM * self.PM).sum(axis=1)
                    # 量販店 z の売り上げ
                    mu=1000
                    sigma=100
                    r=numpy.random.normal(mu,sigma,(param.l,param.m))
                    self.MS = (self.VM *r*1.05*self.PM).sum(axis=1)
```

```
        # 量販店 z の損益
        self.MF = self.MS - self.MC - 20000

        # 量販店 z の資産
        self.E3 += self.MF

        # 観測生産者
        self.PP_monitor.observe(now(), *self.PP.tolist())
        self.PW2_monitor.observe(now(), * self.PW.sum(axis=0).tolist())
        self.E2_monitor.observe(now(), *self.E2.tolist())

        # 観測卸売
        self.PW_monitor.observe(now(), *self.PW.sum(axis=1).tolist())
        self.PM_monitor.observe(now(), *self.PM.sum(axis=0).tolist())
        self.E_monitor.observe(now(), *self.E.tolist())

        # 観測量販店
        self.PM2_monitor.observe(now(), * self.PM.sum(axis=1).tolist())
        self.PM3_monitor.observe(now(),                                    *
(1.05*self.PM.sum(axis=1)).tolist())
        self.E3_monitor.observe(now(), *self.E3.tolist())
    def start(self):
        u"""同期エージェント集合の動作を開始する。"""
        # エージェント集合のプロセス
        def proc():
            try:
                while True:
                    # ステップ処理
                    self.step()
                    yield pause(self.interval)
            except Exception, e:
                fatal(u"[%s]エージェントエラー: エージェント集合のプ
ロセスでエラーが発生しました。" % self.name,
                      "error in agentset process", tb = True)
        # エージェント集合のプロセスを起動
        activate(proc)()
    def view(self):
        u"""エージェントの描画を開始する。"""
```

```
                interval = 1 # 表示間隔
                #screen = self.getAgentScreen(interval = interval, xlim = None,
ylim = None)
                #screen.addAgentSet(self)
                #screen.start()
            def custom(self):
                pass

        self.aSynchronousAgentSet = SynchronousAgentSet(u"同期エージェント
", self.eEnvironment, simulator = self)
        self.addAgentSet(self.aSynchronousAgentSet)

        def startSynchronousAgentSet():
            self.aSynchronousAgentSet.start()
            self.aSynchronousAgentSet.view()

        self.procs.append(startSynchronousAgentSet)
    def run(self):
        # オブジェクトの起動
        pass
        def startProcs():
            for proc in self.procs:
                proc()
            yield alwaysTrue()
        activate(startProcs)()
        # シミュレーション開始
        start(until = self.until, odesolver = self.odesolver)
        # 結果の保存
        self.save()
        self.flush()
    def objective(self):
        return 0

if __name__ == "__main__":
    # 実行時の処理
    simulator = Simulator(param = SimulationParam(),
        inputDir = u"input¥¥default",
        outputDir = u"output¥¥default",
```

```
        warm = lambda self: 0.0,
        until = lambda self: 100.0,
        odesolver = lambda self: dopri54Solver(dt = 0.1, rtol = 0.001, atol = 1e-06))

    simulator.run()
    result = simulator.objective()
    print result
    simulator.saveResult(result, u"objective")
    if simulator.agentApp is not None:
        print u"エージェント画面の終了待ち受け..."
    simulator.join()
    sys.exit()
```