

SD 入門セミナー

(シリーズ3)

2015 年 4 月~6 月実施

目 次

1. 趣旨と内容	1
2. システムの挙動と因果関係	3
3. モデル	10
4. 経営をシミュレーションする	12
5. フィードバック構造	16
6. モデル深化	28
7. 遅延	49
8. 意思決定	59
9. 非線型性	69
10. 初期値設定	73
11. 品質	77
12. 販売	91
13. 製品市場	104
14. ビール・ゲーム	112

日本未来研究センター
研究員
末武 透

1. 趣旨と内容

(1) セミナーの目的と趣旨

本稿は、日本未来研究センター及びシステム・ダイナミクス学会日本支部との共催で2015年4月~6月に実施した、SD入門コース（第3シリーズ）の説明内容をまとめたものです。

本セミナー・ワークショップは、ST/SDの紹介、入門、そして、ST/SDに関心のある人が集まって、情報交換を行ったり、教え合ったりできる場として実施しています。

(2) セミナーで使ったテキスト

テキストは、

Craig Kirkwood の System Dynamics Methods: A Quick Introduction, 1998 及び Business Process Analysis Workshops: System Dynamics Models, by Craig W. Kirkwood, 1998 を使いました。このテキストは以下のサイトからアクロバット形式のファイルでダウンロードできます。
<http://www.public.asu.edu/~kirkwood/sysdyn/SDIntro/SDIntro.htm>

Craig Kirkwood はアリゾナ州立大学ビジネス・スクールの教授で、専門はサプライ・チェーンや、リスク管理です。

(3) 構成

第2章から第5章（第1回目実施）までは、システムは、指数変化、収斂、S型変化、振動の4つの基本的な振る舞いを示す、あるいはこの4つの基本的な振る舞いに組み合わせで起きる振る舞いを示すこと、この4つの基本的な振る舞いを引き起こす構造は決まっていること、システムをモデルとして記述する方法として、因果関係図とストック・フロー図があること、全てはストックとフローで記述されるモデルで表現できること、ストックとフローに数学的な定義式を与えることで、定量的な分析ができるのだが、数学的な定義に際し、単純に考えないで、現状に合うと思われるセンスの定義を行う必要があること、そして、経営では、週、月、季節などが現シでの顧客の嗜好や行動の変化のパターンがあり、そのことで、システムに変化や共鳴が起きることがあることを説明しています。

第6章（第2回目実施）は、モデル深化のやり方を示す意味で、小売店と工場だけのシンプルなモデルで、小売店が、顧客に販売した量だけを補充する方法、標準在庫を維持するために、在庫調整のための発注を行う方法、パイプラインとして常に受発注が流れていることを意識した発注方法、そして、納期遅れを予想した発注方法などを試しながら、モデルを深化させていきます。

第7章では、遅れと移動平均、平滑化について説明し、遅れには、パイプライン遅れ、指数遅れがあること、その違い、発注や製造計画では、変動を緩やかにするために、移動平均を使ったり、平滑化を使うのですが、平滑化のいくつかのタイプについて説明しています。

第8章（第3回目実施）では、意思決定に関する線型モデルと重み付け意思決定、傾斜係数を使った意志決定のモデルについて説明しています。また、第9章では、非線型を取り扱うやり方として、IF THEN ELSE の他に表関数と使う手法について説明しています。第10章では、初期値設定では、正しい初期値の計算方法、ストックを挟まない変数だけのフィードバック・ループができてしまうと、初期値同時設定問題が発生し、実行時エラーになってしまうことがあること、それを避ける方法について説明しています。

さらに、これまでの学習を踏まえ、**Business Process Analysis Workshops** の項目から、第 11 章として、製造工程での品質プロセスの改善、第 12 章で、販売プロセスでの納期の問題の改善、第 13 章で、製品市場でのリサイクル商品の影響などについて説明しています。

本稿は、日本未来研究センターの研究員である末武透がまとめたものですが、本稿で述べた見解や意見等に関し、私が全面的に責任を負うものであり、日本未来研究センターやシステムダイナミクス学会日本支部の見解を表したものではないことを述べておきます。

2. システムの挙動と因果関係

人間は、何か問題があるとすぐに解消したいと思うのが普通です。こういった、問題をすぐに解消するという性格を人間が持っていたからこそ、人類は発展することができたのだと言えます。従って、人間は、何か出来事があると、問題が発生したと考え、さらには、何かの出来事でその問題が発生したと考えます。つまり、事象（問題・課題）があると、それを引き起こす原因があると考えerわけです。例えば、「売上が悪化した」という問題は、「営業スタッフがたるんでいる」という原因で起きたと考えます。

こういった、単純に、問題があるとそれを引き起こした問題があるというやり方は、単純な構造の問題ではうまくいきますが、複雑な問題構造では必ずしもうまくいかないことがあります。例えば、GM はずっと何十年も、日本の自動車企業が米国参入を始めるまでは、米国でも世界でもトップの企業でした。日本の自動車メーカーが北米に進出し、たちまち車の販売台数で抜かれ、業績が伸び悩みます。GM は、いろいろ原因を分析し、もちろん原因を十分に理解していますし、その後、30 年以上かけて改善しようとしています。また、この改善期間中、改善のために、多くの GM の社員や幹部を解雇してきました。それでも、まだ企業体質を十分改善できていなく、日本企業やその後進出てきた韓国企業などと競争し、競争に十分打ち勝つという状況にまでは至っていません。この事例から、複雑な問題構造では、企業の行動性向を変えることはとても難しいことが分かると思います。

(1) システム思考：

複雑で困難な経営問題を理解する方法として、「システム思考」が 30 年以上前にフォレストラーによって発案され、その後、30 年近くかけ改善され、現在では十分使えるようなものになっています。しかし、この、システム思考という考え方は、通常の、企業の業績を見るような考え方とは違った見方をすることが要求されます。つまり、単独で何か問題が起きるのではないし、その問題が単独の原因で引き起こされるわけでもないと考えます。また、組織をシステムとして、全体的に捉える見方が要求されます。ここで言うシステムとは、独立したグループとか要因がある一つのパターンを形成するようなものです。ここでは、経営を取り上げているので、「要因」は「人」、「技術」、「デザイン」、「市場」、「製品」、「商品やサービスの提供」といったものとなります。実は、経営に関係するものはほとんど全て、ここに挙げたような種類であり、これらの全てが、システムを形成する要因となります。さらには、システムは一つだけではなく、複数になることもあります。

普通は、ルネ・デカルトが「方法序説」で打ち立てたとされる、対象を分解し、最少単位にし、その分解された要素を理解すれば、対象を理解できるとするアプローチで物事を理解しようとしています。この方法は「分析的アプローチ」と呼ばれます。これに対し、システム思考は、要素が連動していると考え、連動しているしくみの観点から対象を理解しようとしています。

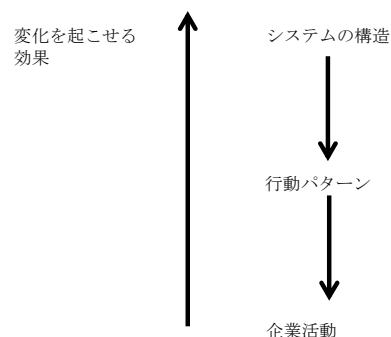


図 2-1：変化を起こす

何か問題が起きると、何か外からの起因や原因でそれが起きると考えがちですが、しかし、この考え方を改め、内部構造のせいで問題が起きると考える必要があります。あるいは、外部の要因よりも、内部構造の方より多く、その問題を引き起こした原因と考える必要があります。例えば、通常、人は、企業の業績を、企業活動が成果や問題などを引き起こしたと考えます。もっと深く考える人は、そういった企業活動や問題をパターンとして見て、振る舞いのパターンにうまくいくもの、問題となるものというように見るでしょう。さらには、そういった問題を引き起こす原因を追究していきます。ただ、因果関係で追いかけることは可能ですが、そうすると永続的な作業になりがちです。「新製品が売れない」という問題の原因は、「営業があまり積極的に売り込まない」からだと思いきや、さらに、「営業があまり真剣に取り組んでくれない」原因を追究し、「営業は他にも仕事を抱え過ぎていて、ゆとりが無い」ということが分かったとします。この営業の忙しさの原因を突き止めると、他の問題が出てくるというように、永続的な追っかけっこになりがちです。こういった追っかけっこをすることは必要ですが、それと共に、何でこんな構造が起きるのかという状況やしくみを考えるべきです。構造を変えることで初めて業績を変えられます。逆に構造を変えない限り、同じような問題は発生し続けます。図 1-1 に示したように、システムの構造を変えれば、企業活動や行動パターンは変わってきます。例えば、アップルは i-phone での音楽配信や自動的に電話料や契約料から料金を引き落とすしくみを作り上げました。このアップルの打ち出した製品とアップルが作ったしくみは、消費者がわざわざミュージック・ショップに赴き、CD を買うといった行動パターンを変えてしまいました。このことにより、音楽 CD を作っていたメーカーも、音楽だけではなく、ビデオも制作し、CD に入れる、あるいはネット配信に力を入れるというように、消費者や企業の行動パターンを変え、さらには企業活動を変えてしまいました。

もちろん、ファッションでブームを起こすなどで、行動パターンを変えることで企業活動を変えることもあります。先のようにシステムを変えた場合ほどには影響は恒久的ではありません。

図 2-2 は、原因ではなく、問題を引き起こす構造に関する説明です。もちろん、問題にはそれを引き起こす原因となっている、原因 A と D があるのですが、原因 A にはさらに、原因 B、C があり、原因 D には原因 E があるという構造で問題を捉えることが重要で、問題には、原因 A があり、それを追究したら、原因 B があることが分かったような、いわゆる泥縄式の問題発見で、問題を起こしている構造を見失ってはいは、問題を効果的に改善することは不可能でしょう。

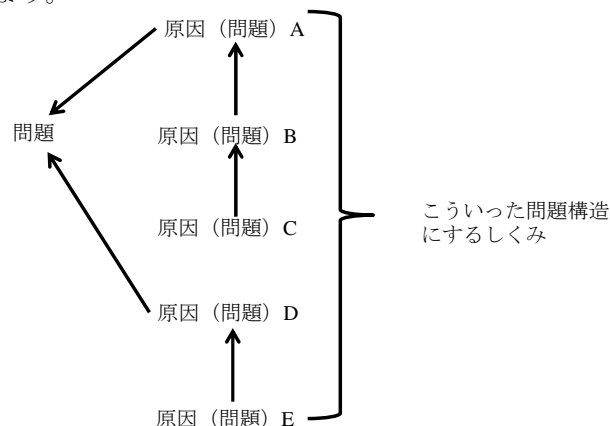


図 2-2：原因ではなく、問題を引き起こす構造

(2) 振る舞いのパターン：

システムの振る舞いにはパターンがあります。このパターンが状況と呼ばれるものです。ここでは経営を取り上げていますので、注目している要素（費用、売上、収益、市場占有等）の時間的変化のパターンです。

いろんな違った状況で起きるシステムの振る舞いのパターンは、実は近似しています。つまり、売上の変化、市場占有の変化などのパターンは、会社や業種、市場が違っていても、いくつかのパターンに収まるということが知られています。また、あるパターンの振る舞いは、あるシステム構造から引き起こされるということも分かっています。

典型的なパターンには、指数的变化、収斂、成長曲線あるいは S 型曲線、振動の 4 つあります。これが独立して現れることもあれば、これが複合して現れることもあります。

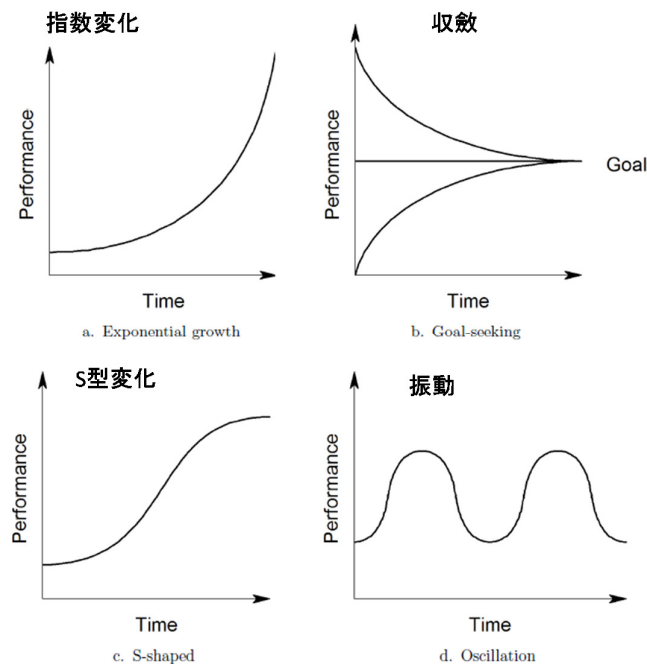


図 2-3：システムの基本パターン

・指数変化：

指数変化は、成長や加速が時間経過と共にどんどん早まる、高まるというもので、新製品の売上などで見られます。図 2-3 左上のような振る舞いをします。

・収斂：

収斂は、増加するものもあれば減少していくものもありますが、ある目標に向かって収斂していくものです。図 2-3 右上のような振る舞いをします。

・S 型変化：

成長曲線、あるいはロジスティック曲線と呼ばれる変化で、図 2-3 左下のような振る舞いをします。

・振動：

振動は、図 2-3 で右下のような振る舞いをします。

実際には、これらが複数組み合わせられたものとなります。例えば、成長曲線と振動が組

み合わさり発散する、目標収斂型と振動が組み合わさり収束する、などの振る舞いを示します。

(3) フィードバックと因果関係図：

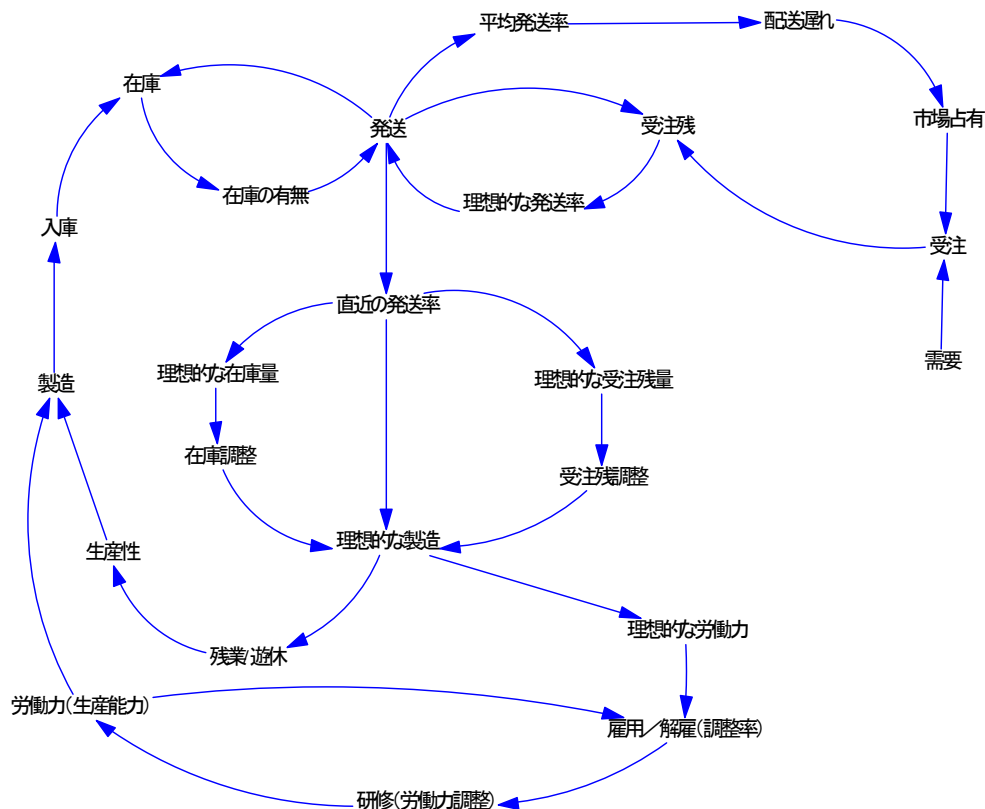


図 2-4：ある製造業の定性モデル（Richardson/Pugh による）

SD では、図 2-4 に示したような、因果図を描いて対象を分析することがよく行われます。図 2-4 は Richardson/Pugh の論文に掲載されていたもので、製造は生産性と労働力に関係するという関係を示しています。これを文章で説明するとなるととても説明が難しいか複雑になるのですが、このような図にすると、図なので目で理解できます。ここには、フィードバックを示す、一連の原因とその結果の関連がクローズ（閉鎖的）に繋がったもの、あるいは、円として行為と情報の繋がりが閉じたものが示されています。フィードバックにはオープン（閉じていない）な関連もあるでしょうが、そうであれば、関連性は一方的で、再び最初に戻って影響することはありません。SD では、クローズなフィードバックについて注目し、原因がまわりまわって元に戻ってくることを問題視しています。

こういったモデルでは、原因と結果の関係で要素を関連付けていきます。そこでは、原因の増加が結果を増加させるようなものは正として、「+」、「S」などの記号を付けた矢印で原因と結果を結び付けます。この関係では、原因の減少が結果を減少させるということでも同じです。

一方、原因の増加が結果を減少させるようなものは負として、「-」、「O」などの記号を付けた矢印で原因と結果を結び付けます。この関係では、原因の減少が結果を増加させるということでも同じです。

こうした関係でモデルを記述していったら、フィードバック・ループが完成した時に、ループ全体での振る舞いで、その振る舞いがどんどん増加していく、あるいは減少していくようなものを、増強ループと呼びます。因果関係で正になるものだけで形成される、あるいは、負になるような関係が偶数あるようなものは増強ループになります。

逆に、ループ全体での振る舞いで、その振る舞いが収斂に向かっていくようなものを、均衡ループと呼んでいます。因果関係で負になるような関係が奇数あるようなものは均衡ループになります。

(4) システムの構造と振る舞いのパターン：

この、増強ループ、均衡ループ、そしてその組み合わせによって、概ね 4 つの典型的なシステムの振る舞いのパターンが生まれます。

- ・増強ループ：人によっては、悪循環、好循環、雪崩現象などと呼んでいますが、指数的增长をします。
- ・均衡ループ：収斂型の振る舞いを示し、収束、負の成長曲線などになります。
- ・遅れを伴う均衡ループ：遅れを伴う均衡ループ振動をもたらします。振動は発散、あるいは収束することもあります。
- ・均衡ループと増強ループの組み合わせ：均衡ループと増強ループが複数個組み合わせると S 型変化になります。これは、増強ループが均衡ループによって制約を受け、最初は増強ループのせいで指数的に増加していたのが、均衡ループが効いてきて、収斂に向かいます。

(1) 増強ループの例：

預金に対して利子が付くので、複利であれば指数曲線で増えていくし、単利であれば、直線的に増えていきます。(図 2-5 左)

(2) 均衡ループの例：

室温調整の例で、実際の温度と設定した温度との差を見て温度調整が行われ、差がどんどん縮まっていき、最終的には実際の温度は設定した温度になるので、差はゼロになっていきます。(図 2-5 右)

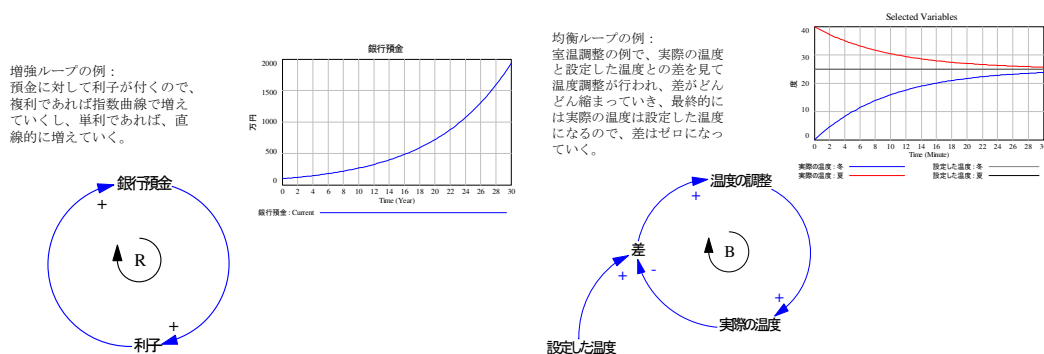


図 2-5：増強ループ例（左）と均衡ループ例（右）

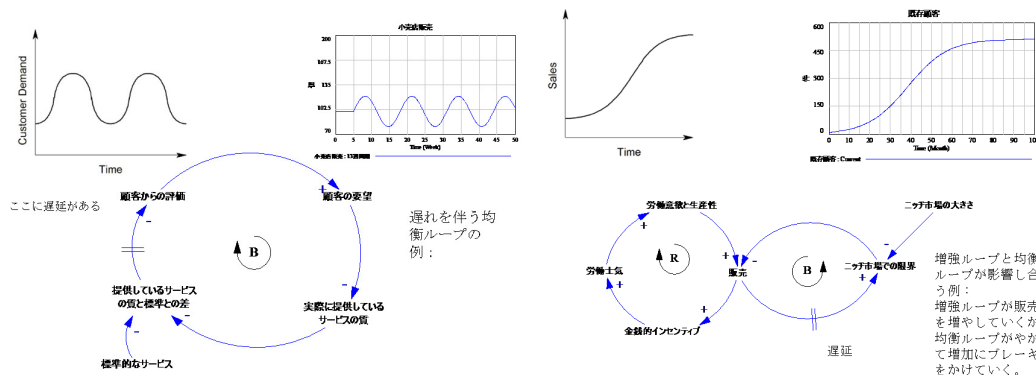


図 2-6：遅れを伴う均衡ループ例（左）と増強と均衡のループが干渉する例（右）

(3)遅れを伴う均衡ループの例：

顧客の要望に応え、サービスの品質を変えています、それがすぐに行われるわけではないので、遅延のせいで顧客の評価が振動します。それが製品の販売数やサービスの売上の揺れ動きになって現れます。（図 2-6 左）

(4) 増強と均衡のループが干渉する例：

金銭的インセンティブ制度を導入し、営業マンの労働意欲を高め、販売を上げていきますが、ニッチ市場という限界から、いくらインセンティブで営業マンの販売意欲を高めようとしても、売上はやがてサチュレートしてしまいます。（図 2-6 右）

(5) 因果関係図の描き方：

因果関係図（システム思考モデル、あるいは定性モデル）の記載のやり方やルールのようなものを説明します。

1) 要素は、キーワードであり、名詞で記載する

要素は変数になるのですが、ここでは厳密な変化までは気にしなくともいいでしょう。要素名には動詞ではなくなるべく名詞を使って下さい。つまり、要素自体ではなく、要素が矢印で結ばれることを前提として要素名を決めます。例えば、「費用増加」ではなく「費用」とし、矢印が結ばれ、費用が増加することが表現されるようにします。

どちらの方向を向いて増加するのかを明確にして要素を決めます。「犯罪に寛容」とする方が、「犯罪に対する態度」よりはいいでしょう。また、要素は増えるとか、ポジティブな名前にする方がいいでしょう。「増加」の方が「縮小」よりふさわしいでしょう。

因果関係は因果で決めます。時間変化では決めません。例えば、「A が起き、それによって B が起きる」のであり、「A が起きた時 B が起きる」ではありません。ただ、私自身のコメントとして、日本語ではなかなか難しい面があるので、ここは、最初はあまり気にしなくてもいいように思います。要素は、キーワードであり、名詞で記載すると言われても、そうもいかないこともいろいろあると思います。

2) 予期しにくいかもしれないが起きうるものを考える。それによって副作用を引き起こす因果の線を加えることができる。

3) 均衡ループには必ず目標（ゴール）が存在する。目標があれば、ギャップが存在する。（ただし、ゼロに収斂するなど、目標は暗黙的な場合もある。）

4) 実際に行為が行われる、あるいは実際に事象が発生する時と、それを知覚する時の間に

は、通常時間的遅れがある。このことによって、システムの振る舞いの変化がパターンとして生まれる。例えば、商品の品質を変えた後、時間遅れで顧客は、品質が変わったことを知覚する。

5) 通常、短期的な反応と長期的な反応は別である。従って、別のフィードバック・ループとして記述する方がいい。例えば、落込んだり、気分がすっきりしない時に、薬物によってすっきりさせることができるが、これに頼りすぎると、薬物中毒になってしまい、かえって長期的には悪影響を及ぼす。

6) 2つの要素の因果関係に関し、もし、多くの説明が必要であれば、中間に別の要素を挟み、単独での関連性を単純化かつ明確化する方がいい。

7) 因果関係図はなるべく単純な構造にする方がいい。全ての経営的な要因を因果関係図に取り込む必要はなく、主要な要因があり、目的とするフィードバック・ループの振る舞いを説明できればいい。

(6) まとめ

この章では、SD は、対象をシステムとして理解し、解析する方法であること、システムは基本的な振る舞い、あるいはその組み合わせで表現される振る舞いをする事、システムを定性モデルとして表現する方法として、CLD (因果関係図) の書き方を説明しました。

3.モデル

(1) ストック・フロー・モデル（定量モデル）：

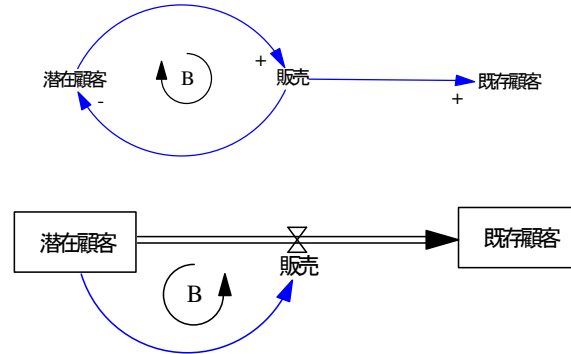


図 3-1：因果関係図とストック・フロー・モデル

どのようなビジネスであれ、そのモデルは、ストック、フロー、そして情報で構成されます。また、通常は、モデルは、時間に対して変化する変数で構成されます。先に説明した、因果関係図は、このストック・フロー・モデル（定量モデル）で解釈できます。因果関係図では、ストック、フロー、情報の区別はありませんでしたが、この、定量モデルでは、ストック、フロー、情報を区別します。ただ、この段階では、まだ等式でパラメータを規定していいので、このモデルをストック・フロー・モデルと呼びます。図 3-1 にサンプルの因果関係図とストック・フロー・モデルを示しています。

下のものが、ストック・フロー・モデルで、箱はストックを示します。ストックは水準（レベル）、あるいは累計を示すものです。蝶ネクタイのようなものがフローを表しています。フローは流量、あるいは率（レイト）を示すものです。

(2) ストック・フロー・モデル（定量モデル）：

SD モデルは、最低 1 個のストックとそれに連結する最低 1 個のフローで構成されます。つまり、SD モデルは、ストック、フロー、変数で構成されます。最初にこの表現方法をフォレストアが発明したのですが、この単純さは革命的で、他のコンピューターによるモデルではもっといろんなものを使います。例えば、フロー・チャートはもっと多くの種類の記号を使います。

(3) ストックとフロー：

1) ストック

ストックは物理的なもので、累積される性格のものです。基本的に計測できるものはストックであり、時間を止めた時に、物理的な量が存在するものです。工学では状態変数(state variable)と呼ばれるものです。

ここでは経営を対象に考えていますので、物、人、資産、受発注、金の 5 つが代表的なストックになります。

- ・物：原材料、仕掛品、完成品などで、工場で製品になる前であっても、製品になってからであっても、さらには、在庫や流通チャネルに流れても、置かれている状態に係らず、物として考え、ストックになります。
- ・人：実際の人、ただし労働時間はストックではありません。
- ・資産：工場、機械、設備など製品を製造したりサービスを提供するためのものです。

- ・受発注：製造指示、採用、不動産購入、設備投資（工場建設、機械購入）など
- ・金：基本的には現金。支払、入金などはフローになりますが、違ったストックの状態の金の間を移動する情報と解釈できるかもしれません。現金も、最近はクレジット・カード決算なので実際にお金を受け渡すわけではないのかも知れませんが、計測でき、時間を止めても金額が存在するのでストックになります。

物、人、資産は物理的なものなのですぐにストックと分かりやすいのですが、最後の受発注と金は、最近はコンピューターで処理され、物理的なものとしては扱われないので、少し分かりにくくなっています。

2) フロー

これに対し、フローは、累積されないもので、基本的に計測できないものはフローになり、時間を止めた時に、値が存在しないものです。

3) 情報

矢印が情報を示しています。図 3-1 のモデルでは、潜在顧客から販売に矢印があり、これは潜在顧客に関する情報が販売というフローにもたらされることを示しています。しかし、この図では、既存顧客から販売という矢印はないので、既存顧客に関する情報は販売には影響しないことになります。

かつては、中間管理職は、上層部と下層部の間に情報を取り持つことが仕事でしたが、電算化時代を迎え、その必要性は無くなったように思われます。また、かつては、情報をどう取り扱い、伝達するかが問題だったのですが、現在は、電算機を使い、いつでも、だれでも、瞬時に情報伝達が可能となり、情報伝達よりも、どのような情報をどう処理するかが重要となってきています。物的なものは、瞬時に移動するとか、同時移動するといったことができませんが、情報は瞬時に複数の相手に伝えられるので、モデルとして表現しにくくなっています。

(4) まとめ

この章では、モデルをストックとフローで記述できること、その方法を説明しました。

4.経営をシミュレーションする

ストック・フロー図は因果関係図よりは多くの情報を表現できますが、ただ、このままでは、システムの数量的な変化は分かりません。例えば、図 3-1 の下の、潜在顧客と既存顧客のモデルでは、潜在顧客がどう変化するのかまだ分からないでしょう。そのためには、グラフで数量的な変化を示す必要があります。

モデル構築では、どこまで細部をモデルに取り込むか、どう不確定要素を扱うかが重要な事項となります。経営モデルでは、中間レベルに焦点を合わせていくことが重要で、下層部に合わせると、機械の設置場所といった細かい事項に振り回されることになりますし、トップに合わせると、重役の個人的意志決定でがらっと変わってしまうといった不確定性に振り回されてしまいます。また、実際には会社は中間層で動いていて、この部分に問題改善の効果があります。この部分では、極めて細部にわたるモデルは必要ありません。例えば、新製品を出すためには販売能力強化、設備投資の意志決定や資金調達といった事項が関係してくるでしょう。ここで、販売能力強化に関し、営業スタッフへの人事方針と業務品質は関係するので、モデル構築で考慮すべきかもしれませんが、だからといって、個々の従業員の給与水準とか有給休暇のスケジュールといったものを考慮することまではしなくともいいでしょう。通常は、もっと統合的なアプローチで十分です。

経営モデル構築では、既存の業務プロセスを改善することに目的があり、詳細なモデルを構築することが目的ではありません。よく、詳細なモデルを作ろうとする人がいます。しかし、詳細なモデルを作ろうとするとデータ収集で時間を費やしてしまい、しかも完成した頃には、対象としていた経営問題はもうとっくに処理されてしまい、モデルは役に立たないということも多くあります。比較的単純なモデルを作って、タイムリーに経営改善に役立つようにすることの方が有益です。

SD でのシミュレーションは、実は 2 つのことを前提にしています。

- 1) プロセス内部の流れは連続している
数量は連続してプロセスに沿って時間変化する。
- 2) プロセスにはランダムな要素を含まない
数量はプロセスに沿って規則変化し、むちゃくちゃな、決まりもなにもないような状態では変化しない。deterministic variable=確定変数であり、不確定変数ではない。

実際には、経営での業務プロセスは、連続的というよりは離散的なのですが、それは人とか商品といったものであり、基本的には連続と考えて処理できると考えられます。

ランダム性（不確実性）も実際の経営によくあるのですが、それを分析し、不確実性を減らす、あるいは無くすることが経営のポイントとなるように思います。また、ある程度不確実性を含んでも、その程度が低く、業務プロセスで処理できるものであれば、あまり業務には影響はないと考えていいでしょう。例えば、従業員が風邪を引いて会社を休むということはありうるのですが、必ずそうなるわけではないので、不確定なわけです。ただ、従業員の 1 人や 2 人風邪になっても、業務全体にはあまり影響はないと考えられます。不確実性では、影響度もさりながら、何が不確実性を生むのかを発見することが重要です。

(1) ストックの定義式：

モデルは水道管のシステムのようなもので、ストックは水が入ったタンクや浴槽、フローはパイプやポンプと思ってもらえばいいでしょう。定量モデル構築では、最初にスト

ックの初期値を決め、次に、フローの定義式を決めます。

ストックは数式的には以下のように定義されます。すなわち、時間 0 から時間 t までの販売の $d\tau$ 単位時間による積分値です。今、図 3-1 の下のモデルの定義式を考えます。ここには、潜在顧客と既存顧客の 2 つのストックがあります。

$$\begin{aligned}\text{潜在顧客} &= \int_0^t \text{販売}(\tau) d\tau \\ \text{既存顧客} &= \int_0^t \text{販売}(\tau) d\tau\end{aligned}$$

ただし、Vensim PLE では自動的に積分計算の定義を行ってくれるので、通常はこの自動定義を用い、初期値だけを定義します。もちろん、定義式を修正することも可能です。初期値として、潜在顧客 10 万人、既存顧客 0 人とします。

(2) フローの定義式：

フローはきちんと定義式を与える必要があります。例えば、よくありがちな定義のやり方として、営業担当に聞いたら、月に 25,000 個製品が売れる（2,500 人が既存顧客になる）というのでその値を設定します。ただ、常識的に考えて、潜在顧客の数がゼロになったら販売はゼロになるので、

$$\text{販売}(t) = \left\{ \begin{array}{l} 25,000 \quad \text{潜在顧客}(t) > 0 \\ 0 \quad \quad \quad \text{その他の場合} \end{array} \right\}$$

といった定義にします。

この定義のやり方が間違っているというわけではないのですが、潜在顧客は、ある割合（例えば 2.5% の顧客）で製品を購入すると考える方がもっと現実的なので、

$$\text{販売}(t) = 0.025 \times \text{潜在顧客}(t)$$

としてみます。

2 つの定義のやり方を比較してみましょう。図 4-1 にその結果を示しています。最初のものでは、40 ケ月目で潜在顧客はゼロになりますが、2 番目のものでは、100 ケ月たっても、潜在顧客はゼロにはなっていません。明らかに、2 番目のものの方が現実に近いと思いませんか？

ここでは、販売率を 2.5% と固定にしていますが、これも変化すると考えられます。従って、モデルでは、2 番目のもののようにするのであれば、さらに、販売率を別の変数としてしまう方がよく、こうすれば、いろいろ販売率を変えて、シミュレーション結果の違いを見ることができます。

モデルを考えていく上で、要素の関係を単純に考えてしまうと、つい、2 つの要素の関係は、先のように、毎月 25,000 個づつ売れているといったように考えてしまいがちです。あるいは、そのように考えている人が多くこの世には存在します。構造を考えないと、こういった短絡的な考え方に陥りがちになり、だんだん減っていく、割合で変化するというようには考えないものです。注意しましょう。

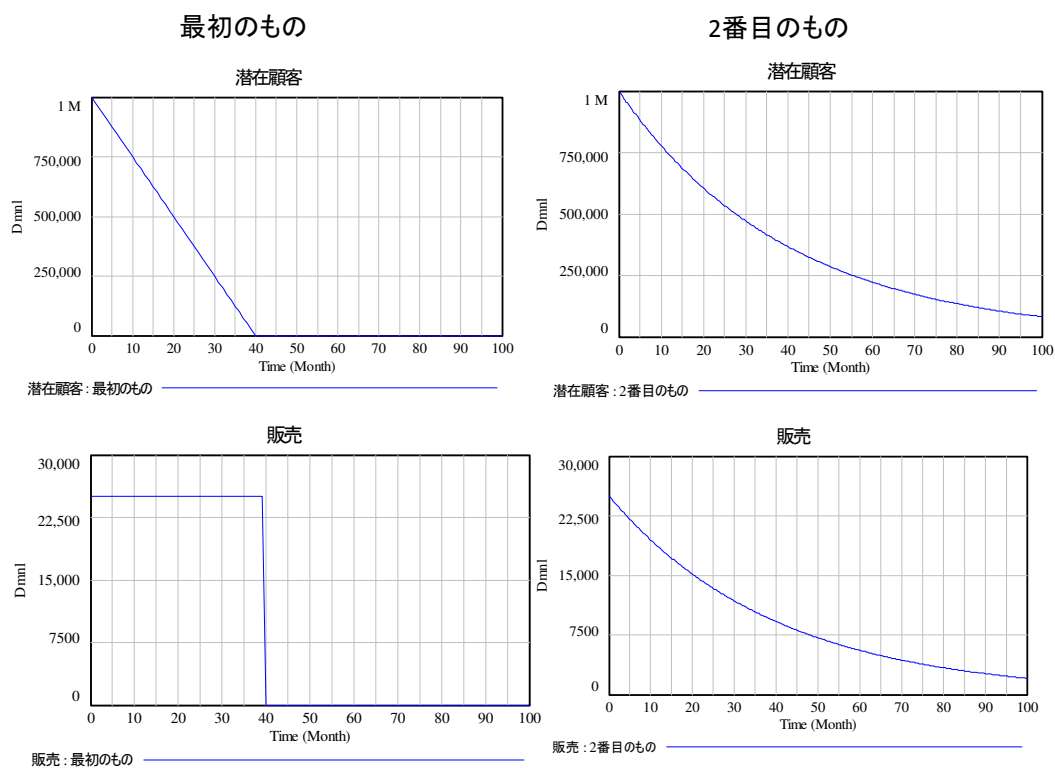


図 4-1：フロー販売に対する定義を変えたシミュレーション結果

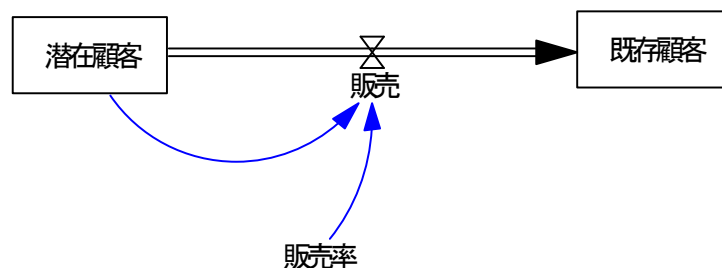


図 4-2：図 3-1 のモデルの改善

参考までに、2つのモデルの定義式を記載しておきます。赤で示した(7)式が違っている箇所です。

最初のものの定義式：

(1)FINAL TIME = 100

Units: Month

The final time for the simulation.

(2)INITIAL TIME = 0

Units: Month

The initial time for the simulation.

(3)SAVEPER = TIME STEP

Units: Month [0,?]

The frequency with which output is stored.

(4)TIME STEP = 1

Units: Month [0,?]

The time step for the simulation.

(5)既存顧客= INTEG (販売,0)

Units: Dmnl

(6)潜在顧客= INTEG (-販売,1e+006)

Units: Dmnl

(7)販売= IF THEN ELSE(潜在顧客>0, 25000, 0)

Units: Dmnl

2 番目のものの定義式：

(1)FINAL TIME = 100

Units: Month

The final time for the simulation.

(2)INITIAL TIME = 0

Units: Month

The initial time for the simulation.

(3)SAVEPER = TIME STEP

Units: Month [0,?]

The frequency with which output is stored.

(4)TIME STEP = 1

Units: Month [0,?]

The time step for the simulation.

(5)既存顧客= INTEG (販売,0)

Units: Dmnl

(6)潜在顧客= INTEG (-販売,1e+006)

Units: Dmnl

(7)販売= 潜在顧客*0.025

Units: Dmnl

(3) まとめ

この章では、ストックとフローで表されるモデルの定義のやり方を説明しました。等式定義では、単に月に何個製品が売れるといったような単純な定義にするのではなく、販売率で販売数が変わっていくように定義するなど、より現実に合うような定義を行うことが重要です。

5.フィードバック構造

因果関係図ではフィードバック構造を説明することができ、可能性としてありうるシステムの振る舞いを定性的に説明はできても、定量的には説明できないという欠点があります。経営では、それで意志決定を行なわなければならないので、何かをきちんと説明するためには定量的な説明が求められます。

4つの基本的なシステムの振る舞いがあり、それは、フィードバックの構造で決まると説明しました。それについて解説を加えてみましょう。

(1) 指数変化：

指数的变化は、倍、倍になっていくような変化で、最初はゆっくり増加しますが、時間経過と共に増加が増えていきます。また、1が2に増える期間と、1,000が2,000に増える期間、1,000,000が2,000,000に増える期間は同じです。図4-2の指数変化のグラフを見て下さい。これは、図4-1の定量モデルをそれぞれ、40年、200年でシミュレーションしたものです。詳細に見れば、指数変化しているのですが、200年の期間（図4-2右）で見た場合、最初の40年は変化していないように見えます。しかし、40年の期間でシミュレーションした結果（図4-2左）では確かに指数変化しています。

図4-3の上に出したような構造の、こういった、ストックから入力側のフローに増強ループのフィードバック・ループがある構造のシステムの振る舞いは必ず指数的变化になります。ただし、フローが正である必要があります。また、フィードバックがなければ、直線変化になります。

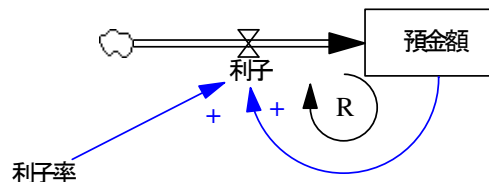


図 5-1:指数変化のモデル

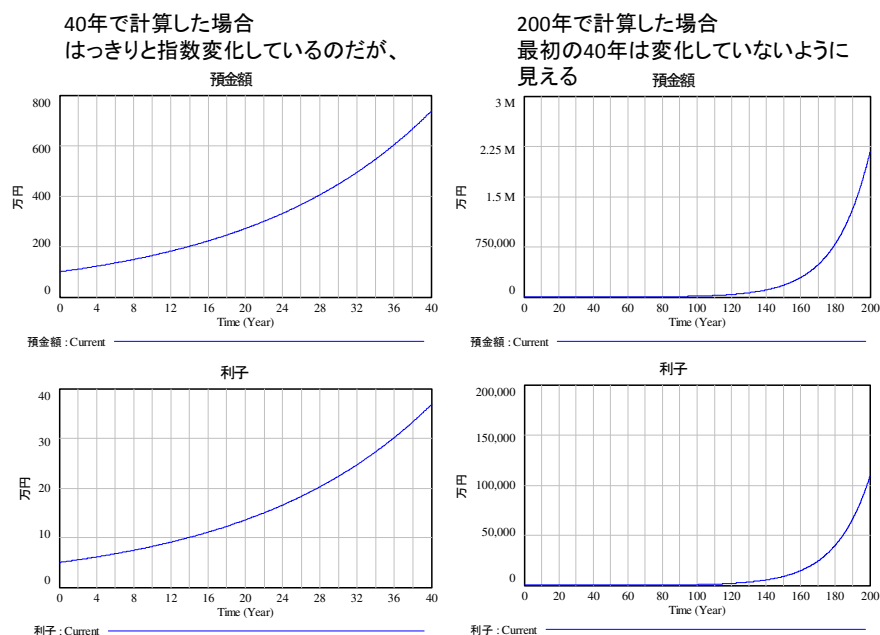


図 5-2:指数変化のシミュレーション結果

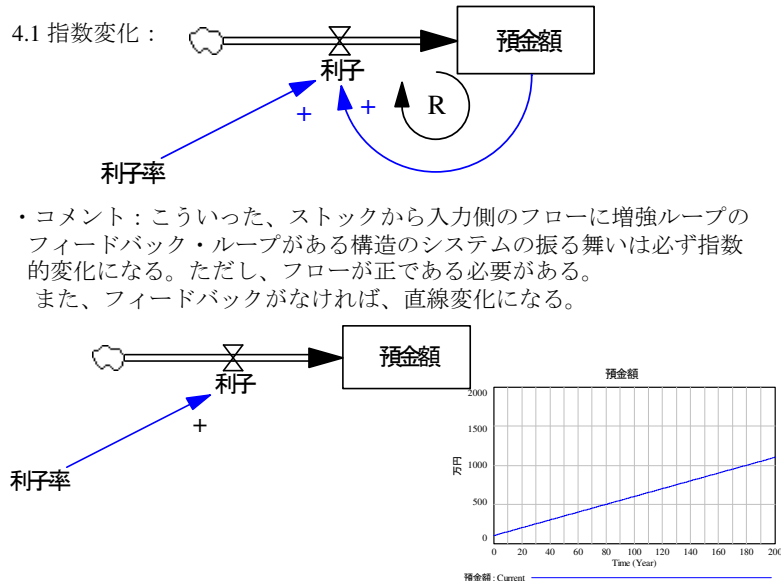


図 5-3:指数変化と直線変化

念のため、図 5-1 の定量モデルの等式を示しておきます。このシミュレーション結果は、図 5-2 の左で示したようになります。(1)式で 40 になっていますが、シミュレーション実行条件定義画面で、終了時を 200 に変更すると、(1)式の値が 200 となり、シミュレーション結果は、図 5-2 の右にしめたようになります。

(1)FINAL TIME = 40

Units: Year

The final time for the simulation.

(2)INITIAL TIME = 0

Units: Year

The initial time for the simulation.

(3)SAVEPER = TIME STEP

Units: Year [0,?]

The frequency with which output is stored.

(4)TIME STEP = 0.0625

Units: Year [0,?]

The time step for the simulation.

(5)利子= 利子率*預金額

Units: 万円

(6)利子率=0.05

Units: Dmnl

(7)預金額= INTEG (利子,100)

Units: 万円

(2) 収斂：

収斂は、何かに向かって増加する、あるいは減少するような振る舞いです。この振る舞いを表現するために、図 5-4 のようなモデルを例に取り上げてみます。ここでは、販売を、販売期間で、平均販売量というストックを使って平準化しています。経営モデルでは、顧客からの変動する受注をそのまま発注するわけにはいかないので、ある期間の平均を取る

などの平準化を行い、そこで発注数を決めるというやり方を行うのが通常のやり方です。しかし、顧客がある時、注文数を変えました。ここでは、販売（量）を時間 10 で増加（減少）させています。モデルでは、この増加（あるいは減少）した値が、目標値になって、この目標値に向かって収斂していくように振る舞います。

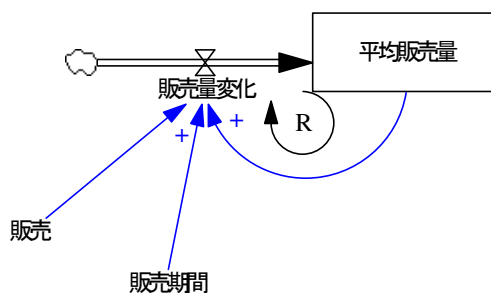


図 5-4:収斂のモデル

図 5-5 にそれぞれ、増減させた場合、減少させた場合のシミュレーション結果を示しています。収斂は、通常は目標値（理想値）があり、それに向かって達成していくような構造の均衡ループで見られる振る舞いです。例えば、図 5-6 はその典型的なもので、こういった、目標値があり、入力側に、現在の実績値とのギャップを埋めるような活動を行う均衡ループが存在すれば、そのモデルの振る舞いは必ずプラスの収斂を示しますし、従って、通常はモデルには何か目標値がパラメーターとして存在します。しかし、目標値はゼロなど暗黙的である場合があります、その場合はストック・フロー図には示さないことも多くあります。

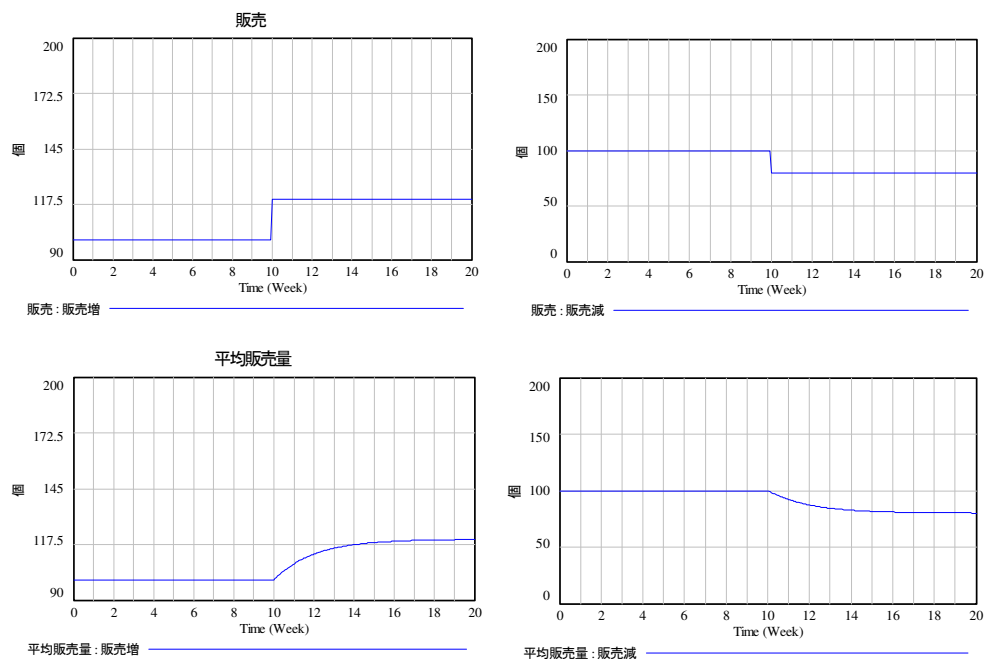
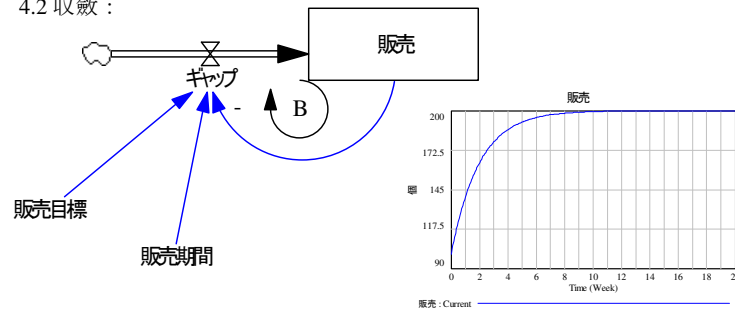


図 5-5:収斂のモデルのシミュレーション結果

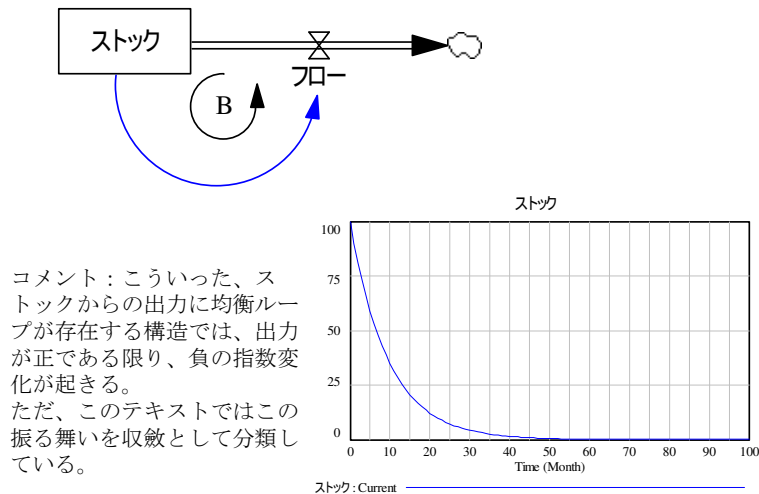
4.2 収斂：



- ・ コメント：収斂は、通常は目標値（理想値）があり、それに向かって達成していくような構造の均衡ループで見られる振る舞いである。
- ・ 従って、通常は何か目標値がある。
- ・ しかし、目標値はゼロなど暗黙的である場合があり、その場合はストック・フロー図には示さないことも多い。

図 5-6:収斂

例えば、収斂を示すということであれば、図 5-7 に示したような、ストックからの出力側で均衡ループを形成するモデルの構造もよくあるパターンです。この構造では、必ずモデルの振る舞いはマイナスの収斂を示します。これが経営モデルだとして、このような構造で、こういったモデルでは、常識的にストックが負になることはないので、ゼロが暗黙的な目標値となり、こういった暗黙的な目標値はわざわざモデルには記載しません。



コメント：こういった、ストックからの出力に均衡ループが存在する構造では、出力が正である限り、負の指数変化が起きる。
ただ、このテキストではこの振る舞いを収斂として分類している。

図 5-7:収斂となるモデル構造

念のために、図 5-4 のモデルの定義式を記載しておきます。

(1)FINAL TIME = 20

Units: Week

The final time for the simulation.

(2)INITIAL TIME = 0

Units: Week

The initial time for the simulation.

(3)SAVEPER = TIME STEP

Units: Week [0,?]

The frequency with which output is stored.

(4)TIME STEP = 0.0625

Units: Week [0,?]

The time step for the simulation.

(5)平均販売量= INTEG (販売量変化,100)

Units: 個

(6)販売= 100+STEP(20,10)

Units: 個

(7)販売期間=2

Units: Week

(8)販売量変化=(販売 - 平均販売量)/販売期間

Units: 個

ここで、図 5-5 の右のグラフでは、式(6)を以下のように変えています。

(6)販売= 100 - STEP(20,10)

いずれも、(8)式で販売と平均販売量の差が販売期間で平準化されます。そのため、販売期間に指定した 2 で、半分ずつ調整されていきます。10 週目に 20 増えます（変更後は 20 減る）。従って、10、5、2.5、1.25、0.625 と調整され、6 週以降は、調整数は目立たなくなり、ほぼ 120 個（もしくは 80 個）に安定していきます。

(3) S 型変化（増加）：

S 型変化は、最初は順調に増加するがどこかで収敛してしまうものです。図 4-8 に示したような構造は、普及型あるいは伝染型(contagion)と呼ばれるもので、マスメディアを使った宣伝ではなく、口コミで新製品や会社名が広がるとか、伝染病のように人の接触で患者が増えるものです。ここでは、当社の製品を使って好印象を持った既存顧客が、まだ製品を使っていない潜在顧客と接触し、接触した潜在顧客の中である割合の人が製品を購入してくれ、既存顧客になるようなことを想定しています。この式は、

購入割合を p として

販売= $p \times$ 潜在顧客 $\times$ 既存顧客

となります。

この構造では、増強ループと均衡ループが影響し合います。最初は増強ループが優勢なので、販売は増加しますが、やがて均衡ループが優勢になり、販売の伸びにブレーキがかかるようになります。従って、既存顧客は、最初は増えていきますが、潜在顧客が減少するので、新規の顧客が減っていき、既存顧客の増加が緩やかになっていきます。図 4-9 でその変化を確かめて下さい。

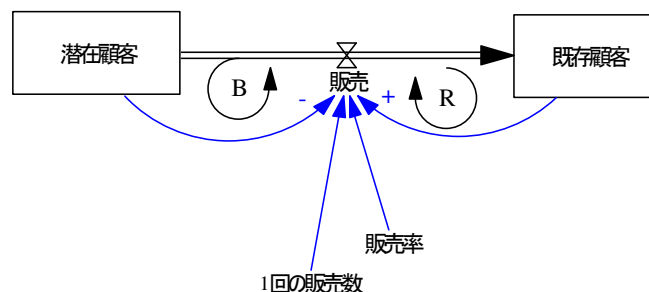


図 5-8:普及モデルと S 型変化

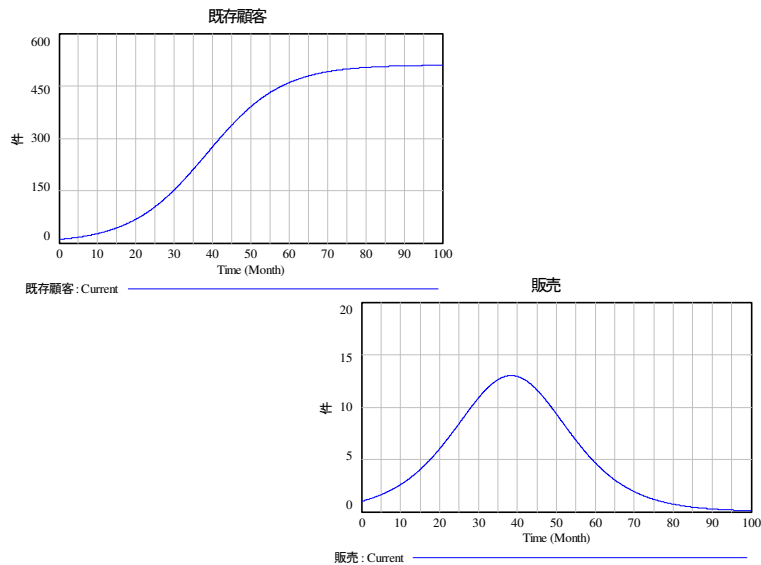


図 5-9:普及モデルと S 型変化

念のため、図 5-8 のモデルの等式を示しておきます。

(01)"1 回の販売数"=0.01

Units: 件

(02)FINAL TIME = 100

Units: Month

The final time for the simulation.

(03)INITIAL TIME = 0

Units: Month

The initial time for the simulation.

(04)SAVEPER = TIME STEP

Units: Month [0,?]

The frequency with which output is stored.

(05)TIME STEP = 0.0625

Units: Month [0,?]

The time step for the simulation.

(06)既存顧客= INTEG (販売,10)

Units: 件

(07)潜在顧客= INTEG (-販売,500)

Units: 件

(08)販売=販売率*"1 回の販売数"*既存顧客*潜在顧客

Units: 件

(09)販売率=0.02

Units: Dmnl

(4) S 型変化（釣鐘型変化）：製品販売のモデルとして

S 型変化のなかで、増加の後、収斂し、さらに減少に転じ、減少がゆっくり収斂していくもので、釣鐘に似ているので、釣鐘型(bell shape)と呼ばれています。図 5-10 は、前に示した図 5-8 のものと基本的に構造が同一です。違いは、製品を使わなくなる顧客の要素を追加しただけです。図 5-8 で説明した通り、販売は、既存顧客からの増強ループで最初は増えて

いいますが、潜在顧客からの均衡ループで増加にブレーキがかかり、さらに、潜在顧客の減少により、販売も減少していきます。

念のため、図 5-10 の定義式を示しておきます。

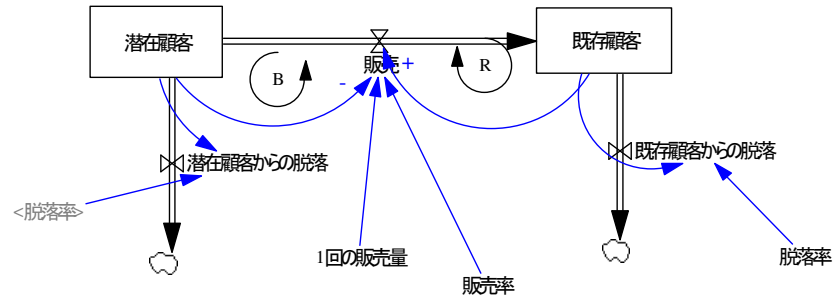


図 5-10:普及モデルと S 型変化

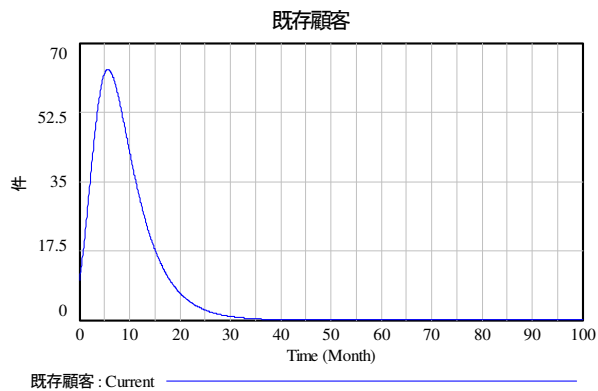


図 5-11:普及モデルと S 型変化

(01)"1 回の販売量"=0.1

Units: 件

(02)FINAL TIME = 100

Units: Month

The final time for the simulation.

(03)INITIAL TIME = 0

Units: Month

The initial time for the simulation.

(04)SAVEPER =

TIME STEP

Units: Month [0,?]

The frequency with which output is stored.

(05)TIME STEP = 0.0625

Units: Month [0,?]

The time step for the simulation.

(06)既存顧客= INTEG (販売-既存顧客からの脱落,10)

Units: 件

(07)既存顧客からの脱落=既存顧客*脱落率

Units: 件

(08)潜在顧客= INTEG (-潜在顧客からの脱落-販売,500)

Units: 件

(09)潜在顧客からの脱落=潜在顧客*脱落率

Units: 件

(10)脱落率=0.2

Units: Dmnl

(11)販売=販売率*"1回の販売量"*既存顧客*潜在顧客

Units: 件

(12)販売率=0.02

Units: Dmnl

(5) 振動：サプライ・チェーンの構造として

ここでは、図 5-12 のような簡単な小売・製造モデルを考えます。ここでは、工場は、小売店からの受注をそのまま製造するとしています。ただし、製造期間と配送時間で合計 3 週がかかり、従って、製品が小売店に届くまでに 3 週間の遅延があるとしています。

小売店の顧客は、商品を毎週 100 個発注してくれていましたが、5 週目から発注数が 120 個に増えました。

小売店は、理想的な在庫（標準在庫）を 200 個としていて、この在庫数と実際の在庫数の差を 2 週間で調整した数量に、顧客から受けた受注数を上乗せした数量を工場に発注することにしています。この構造では、フィードバック・ループに遅延（と調整）が干渉し、システムの振る舞いは振動を示します。シミュレーション結果を、図 4-13 に示しています。

振動を起こすには、この例のように、遅延を挿入してもいいし、ストックとフローの組を干渉させても振動は発生します。ただその場合は、発散するか収束することが多く、遅延もストックとフローの組の干渉も、経営モデルではよくある構造なので、振動現象が現れるのもごく普通のことだと考えて下さい。

念のため、図 5-12 のモデルの定義式を示しておきます。

01)FINAL TIME = 50

Units: Week

The final time for the simulation.

(02)INITIAL TIME = 0

Units: Week

The initial time for the simulation.

(03)SAVEPER = TIME STEP

Units: Week [0,?]

The frequency with which output is stored.

(04)TIME STEP = 0.0625

Units: Week [0,?]

The time step for the simulation.

(05) 周期=13

Units: Week

(06)在庫調整期間=2

Units: Week

(07)小売店からの受注=小売店販売+(理想的な在庫量-小売店在庫)/在庫調整期間

Units: 個

(08)小売店在庫= INTEG (発送-小売店販売,200)

Units: 個

(09)小売店販売=100+STEP(20,5)*SIN(2*3.14159*(Time-5)/周期)

Units: 個

(10)理想的な在庫量=200

Units: 個

(11)発送=DELAY FIXED(製造指示 , 配送遅れ, 製造指示)

Units: **undefined**

(12)製造品= INTEG (製造指示-発送,300)

Units: 個

(13)製造指示=小売店からの受注

Units: 個

(14)配送遅れ=3

Units: Week

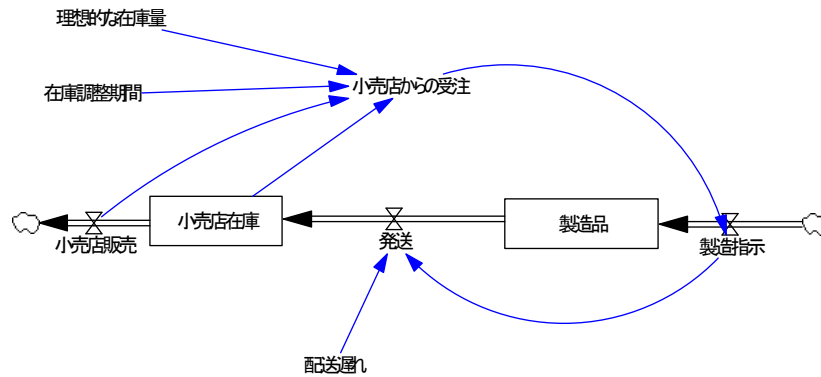


図 5-12:振動

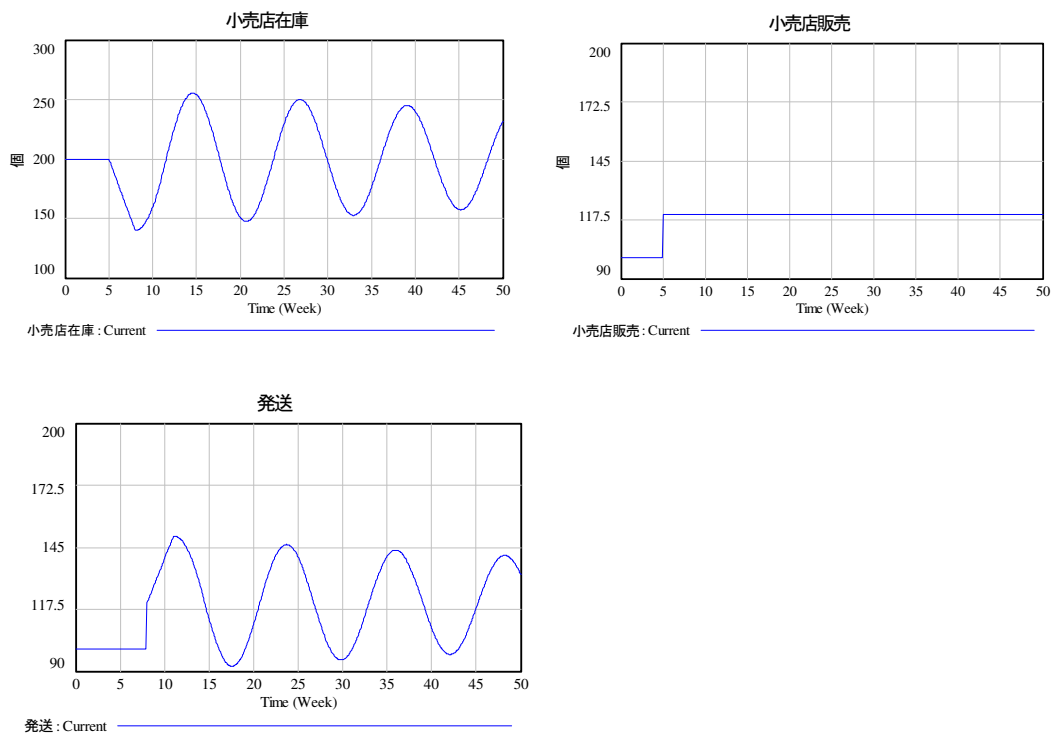


図 5-13:振動

通常は、顧客の購買には波があります。ここでは、これを周期変動で表してみます。図 5-12 のモデルの説明では小売店販売で 20 個増加するとしています、この 20 個の増加に周期があるとします。いくつか変動パターンを考えてみます。ここでは、4 週周期 (月次)、

13 週周期（四半期）、52 週周期（年）の 3 種類の変化のケースを考えてみます。

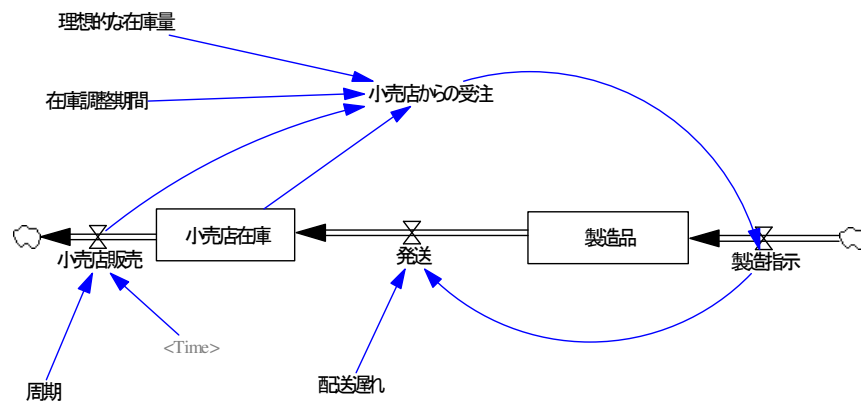


図 5-14: 共鳴

図 5-12 のモデルと基本的に同じで、しかし、小売店販売で 20 個増加する変動の方法を指定するために、「周期」というパラメーターを追加したのが図 5-14 のモデルです。そして、周期を 4、13、52 で変化させ、シミュレーションを行った結果を示したのが図 5-15 です。

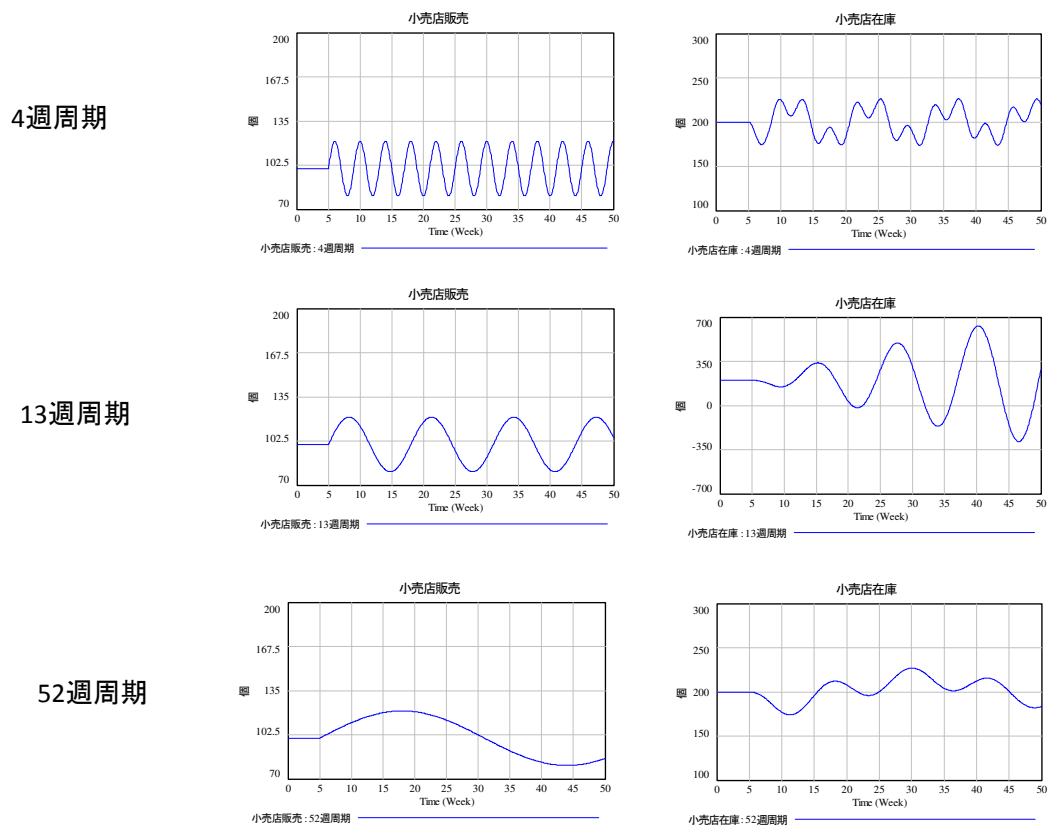


図 5-15: 共鳴

振動があれば共鳴現象が起きることがよくあります。このモデルの場合では 12 週周期で

共鳴が起き、この近くでは、振動が大きくなっています。このモデルの場合、厳密には、12週に近い13週周期でシミュレーションした場合に共鳴が起き、小売店在庫に発散現象が見られます。それより周期が短くとも、長くとも、共鳴は起きません。

工学では、共鳴により異常振動などが起き、飛行機の墜落などを引き起こすことがあります。経営活動にもこういった周期現象はあり、それぞれ違います。こういった共鳴によって、経営にとって、予期した以上の変化がもたらされ、さらに、共鳴のきっかけが外からの原因でおきるという望みしからざる状態をもたらします。共鳴現象は概ね人間や社会の活動の周期である、月、季節、年などの周期で発生するので、企業は、その変化に対応しようとしします。しかし、この企業対応が変動をさらに大きくすることもあります。例えば、自動車の販売数には大きな変化が見られることがよくあり、それは、顧客の嗜好の変化によるものなのか、それとも自動車会社の販売変動への対応によって起きるのかは微妙な話ですが、後者によって顕在化している側面もあると思います。

ただ、いろんな振動は、企業の場合、打ち消し合って消滅している面もあります。

念のため、図 4-14 のモデルの定義式を示しておきます。

(01)FINAL TIME = 50

Units: Week

The final time for the simulation.

(02)INITIAL TIME = 0

Units: Week

The initial time for the simulation.

(03)SAVEPER = TIME STEP

Units: Week [0,?]

The frequency with which output is stored.

(04)TIME STEP = 0.0625

Units: Week [0,?]

The time step for the simulation.

(05) 周期=13

Units: Week

(06)在庫調整期間=2

Units: Week

(07)小売店からの受注=小売店販売+(理想的な在庫量-小売店在庫)/在庫調整期間

Units: 個

(08)小売店在庫= INTEG (発送-小売店販売,200)

Units: 個

(09)小売店販売=100+STEP(20,5)*SIN(2*3.14159*(Time-5)/周期)

Units: 個

(10)理想的な在庫量=200

Units: 個

(11)発送=DELAY FIXED(製造指示 , 配送遅れ, 製造指示)

Units: **undefined**

(12)製造品= INTEG (製造指示-発送,300)

Units: 個

(13)製造指示=小売店からの受注

Units: 個

(14)配送遅れ=3

Units: Week

(05)式が図 4-12 のモデルに対して加えられた式で、(09)式に周期で割るという変更が加わります。(5)式を、4、13、52 と変化させます。

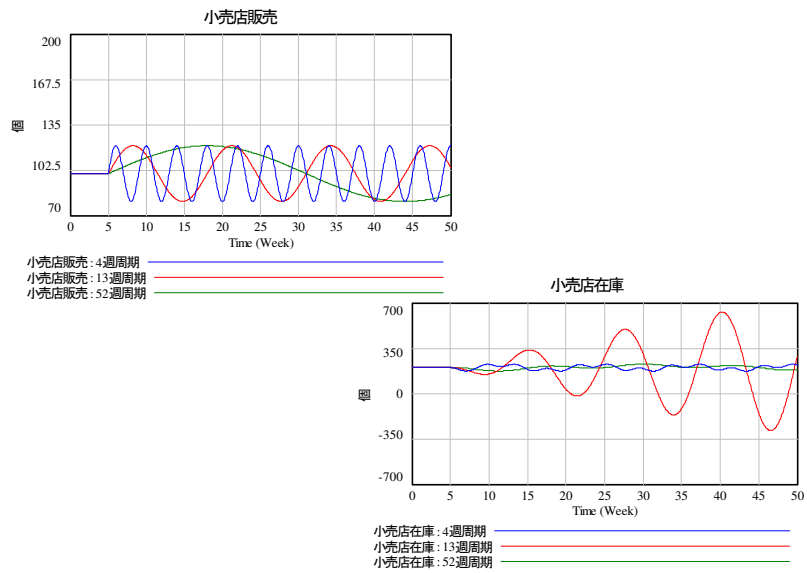


図 5-16: 共鳴

(6) まとめ

この章では、システムの基本的な振る舞いである、指数変化、収斂、振動と S 型変化について、その性格や表現法を説明しました。

6. モデル深化

経営では、特に、製造・流通システムが基本的なビジネス・プロセスになります。製造業ではない会社も存在しますが、この社会では、ほとんどの会社は、この 2 つの機能を基本的なビジネス活動にしている、製品を作る（あるいは調達して）販売しています。こういった、製品を生産し、製造した製品を販売するという形態の会社では、外部の変化が内部構造に影響を与えます。そこで、非常に単純な製造・流通モデルをベースに、製造・流通システムでは、何が必要なのかを、モデル深化を行っていく過程も含め、説明します。

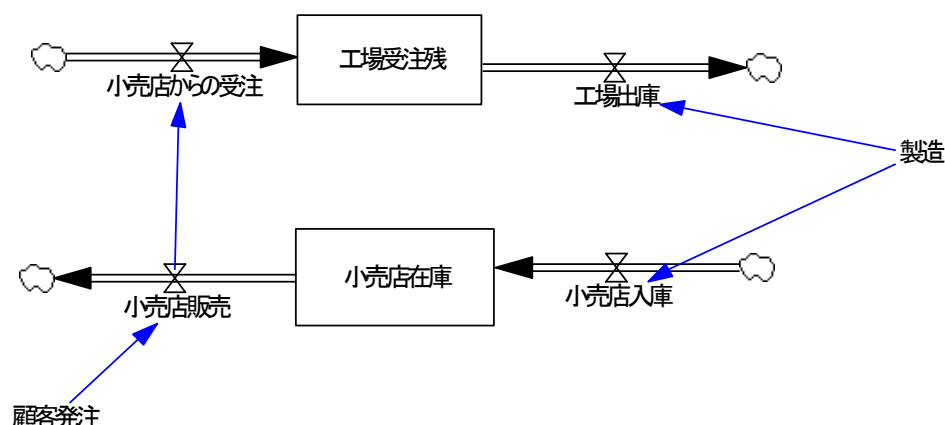


図 6-1：コンセプト・モデル

図 6-1 に、基本的なモデルを示しています。このモデルでは、上部に製造プロセス、下に流通プロセスを示す、ストックとフローが示されています。

上の製造プロセスでは、受発注の流れになっていて、小売店から受注し、工場受注残に入れ、工場受注残を見ながら製造していきます。製品が製造されると、工場出庫として、受注残のストックが製造された分減少していきます。

ここでは、モデルを単純化するために、工場在庫は無い、つまり受注生産とします。製品が製造されると、ストックになった受注残が減るのですが、ただし、ストックである受注残には、再び小売店からの発注を受け、受注分増加し、蓄積されます。

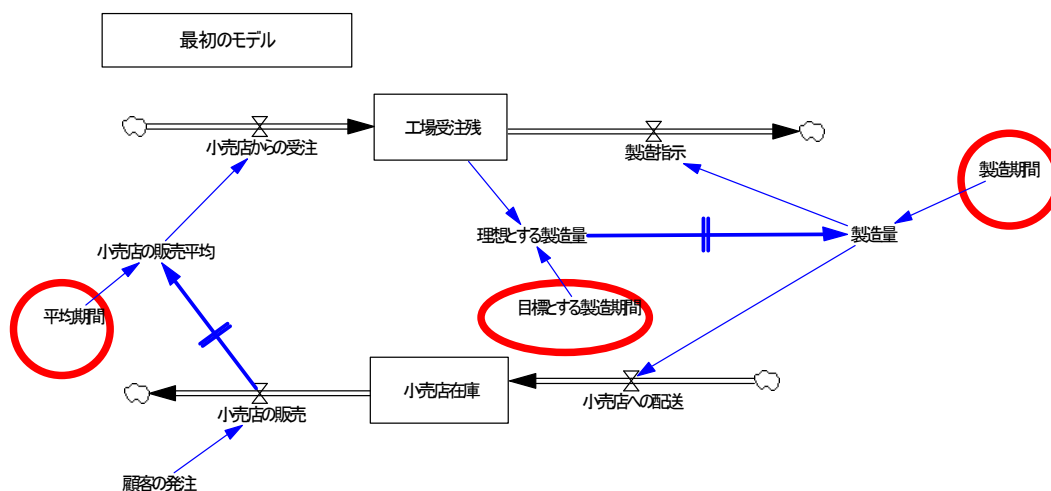
図 6-1 を見ると、この製造プロセスの両側にあるフローの端に雲のようなものが記されています。例えば、この製造プロセスの右端にある雲は、工場出荷された後、製造プロセスの手を離れ、その後何があろうと、製造プロセスでは一切関係しないという意味です。もちろん、実際には工場出荷後、請求業務に引き継がれますが、それはこのモデルの対象外となるという意味です。このように、このモデルでは対象外からの始まりや終わりを示すものです。ここでは単純に、発生源であり、吸収源であると考えて下さい。

下の流通プロセスでは、物の流れになっています。上の工場で製造された製品が配送され、入庫処理を経て、小売店のストックに、在庫として記録されます。つまり、在庫が増えます。小売店には顧客がいて、顧客が商品を発注し、小売店は顧客に商品を販売します。小売店が顧客に商品を販売すると、小売店在庫が減ります。小売店は、工場に商品を発注します。

このモデルは現実をかなり簡略したもので、現実の世界では、小売と工場の間には卸が存在するでしょうし、工場も在庫を持ち、受注を受けて、在庫があればすぐに出荷するはずです。ただ、ここでは、それは考えないことにして、このような小売と工場だけの簡単

なモデルを出発点に、小売店の発注の仕方と、工場での製造の仕方の点で、いろいろ、現実合うように改善していくこととします。

(1) 最初のモデル：小売店の発注方式-1



ていき、それを期間で割っているの、このことは、移動平均を計算していることと同じになります。つまり、「理想とする製造量」は、移動平均で、小売店の発注量を予想して、それを製造に廻していることと同じです。売上や経費などの将来予測を行う時は、よく通常移動平均を使いますが、ここでも、移動平均を使い、小売店からの受注を見ながら、2週間後の小売店の発注を予測しているわけです。

2) 製造指示から製造まで：

製造指示が出された後、その量（ここでは「理想とする製造量」）が工場で生産されます、生産は製造指示で出された製造量（ここでは「理想とする製造量」）を製造期間で平滑化し生産します。通常の工場生産では、製造指示された製造量がどんと一度に製造されるわけではありません。急激な製造指示量の変化があった場合、標準製造期間で製造しようとしても、積み残しなどが発生し、それが量は少しだが、中にはだらだら続くものもあります。こういった変化は、負の指数変化と考えられるので、それを表現するために **SMOOTH** 関数を使います。ここでは、工場生産で製造期間（4 週間）の、平滑化された遅延が発生します。これは、製造が 4 週間遅れるという意味ではなく、ほとんどは 4 週間以内に製造されるのですが、中にはだらだらもっと時間をかけて製造されるものも発生するという意味です。

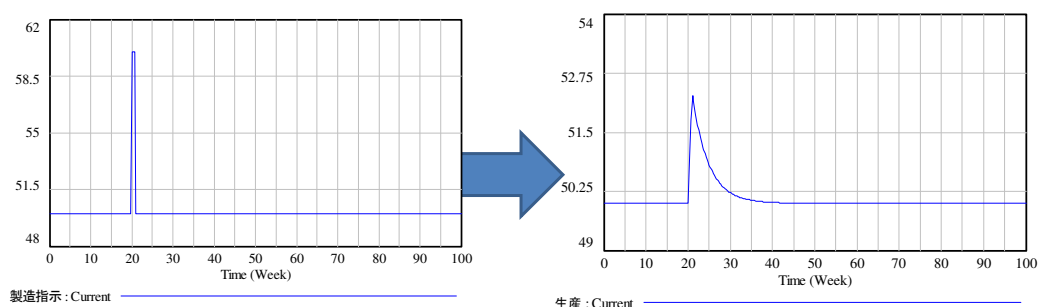


図 6-3：平滑化

図 6-3 に、通常は週 50 個の製品を製造しているのだが、ある時、一次的に（ここでは 20 週目）に製造指示量を 50 個から 60 個に増やした例を示しています。工場は、通常を生産を行うものと思っていて、そういった体制になっているので、急に 60 個になると、これは大変だということで、残業するなどの対応を取ります。ただ、増産が一時的なので、人員を増やすとか、設備投資を行うとかは考えません。また、やろうとしても、設備投資や雇用には時間がかかるので、この急な増産には間に合わないでしょう。残業などをして、何とか、増えた量を製造しますが、それは一度に対応できるのではなく、最初の週は 5 個余分に生産をこなしても、毎日毎日残業だと従業員はだんだん疲れてくるし、毎日毎日全員を残業させるわけにもいけないので、生産性がだんだん落ちてきます。従って、次の週には 3 個に落ち、3 週目には 1 個しか余分に生産できなく、とだらだらと増えた分の処理にかかり、例えば、1 ケ月間かかって、増えた分の生産を終え、やっと通常生産に戻ります。例えば、4 週間で平滑化した場合、生産は図 6-3 に示したような変化で通常生産に戻っていくと考えます。同じようなことは、広告を打った場合の宣伝効果などにも言えます。広告が新聞に掲載された直後はいろいろ問い合わせがあるのですが、だんだん引き合いが無くなっていきます。

前のものは、減少する変化ですが、今度は増加し、それが続く場合を考えてみます。工場には製造増加分をこなせる十分な人員が揃っていますが、すぐ新しい生産のペースに慣れる工員がいる反面、新しい生産のペースになかなか慣れない工員もいて、全員が増加した量を普通に生産できるようになるまでに時間がかかります。このような変化は、図 1-4 に

示したようになると考えられます。

(小売店発注)

3) 小売店の工場への発注

小売店は、普通、売ただけの量を発注しているわけではありません。売れた量とその傾向を見ながら発注量を決めています。ここでは、顧客に売った数量を平滑化して、発注するとします。こうすると、最近顧客に売った数量の方が、過去に売った数よりも重みが高くなり、一種の移動平均となるので、顧客の発注予想を行い、それを元に工場に発注していることに近くなります。ここでは、「平均期間」(1 週間)で、「顧客の発注」を平滑化し、その量を工場に発注するとしていて、SMOOTH 関数で指数化した増加や減少をさせるとします。

実際の顧客の発注はもっと複雑な変化をしているのですが、ここでは変化のインパクトを見る目的のため、単純化し、10 週目に発注量が 20 増えるとします。(ビール・ゲームを意識しています。) このような変化では STEP 関数を使います。

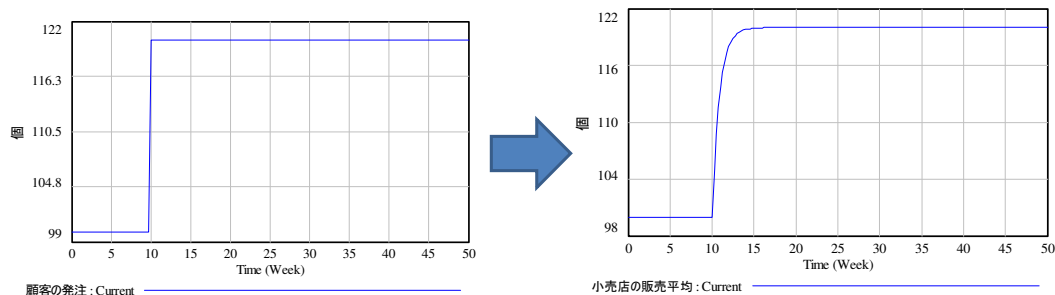


図 6-4：ステップ変数と平滑化

顧客は、それまで 100 個注文していて、しかし、10 週目から 120 個に注文を増やしたという変化を 1 週間で平滑化した場合、小売店の販売平均は図 6-4 のような変化の成長曲線で増えた量に追いつくと考えられます。ここでは、離散変化ではなく連続変化として考えています。個々の小売店ではなく、小売店全体と考えると、1 週間の中で各小売店の発注はばらばらであろうとも、現実的にも全体ではこんな変化になります。

工場の受注残の初期値を 200 とします。顧客が 11 週目で注文を 120 に増やすまでは、小売店の発注には変化はなく 100 個とします。そして、11 週から 12 週にかけて、図 6-5 右上に示したような受注を受けます。受注残は 11 週目から増え、最大 240 まで増えます。(図 6-5 右上)

これを 2 週間で割った数を製造するように製造指示を行います。その変化は、図 6-5 左上の赤で示したようになります。この数量が、4 週間で平滑化されて製造され、製造数は、最大 126 までに上がります。緑で示したのがそうですが、図 6-5 右下の、小売店への配送とおなじなので、こちらの方が、単独で示され、変化が分かりやすいでしょう。25 週以降はほぼ 120 個製造するように戻ります。

小売店在庫の初期値を 400 とします。製造に 4 週間かかっていますので、発注から 4 週間遅れで入荷します。しかも、発注通りではなく、工場で平滑化された形で入荷します。現実にも、工場の生産能力の限界があるので、待ってくれそうな小売店には少し納期を伸ばすなどするので、個々ではなく、小売店の集合全体でみるとこんな感じになります。入荷遅れが影響し、11 週目から 24 週目までは小売店の在庫は減少し、25 週目に回復しますが、しかし、少しの変動は実はその後も続きます。ここでは約 1 年間を想定し、50 週でシミュ

レーションしています。

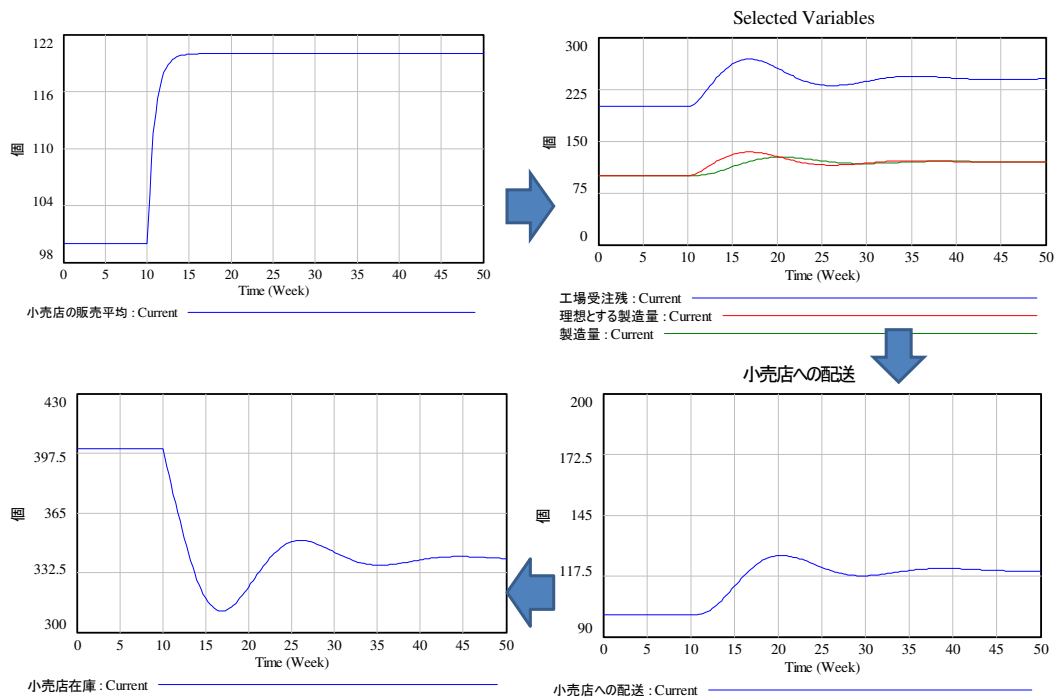


図 6-5：小売店の発注から、受注残、製造、小売店への発送と小売店在庫まで

この SD モデルの式は以下のようになります。

(01)	FINAL TIME = 50 Units: Week The final time for the simulation.
(02)	INITIAL TIME = 0 Units: Week The initial time for the simulation.
(03)	SAVEPER = TIME STEP Units: Week [0,?] The frequency with which output is stored.
(04)	TIME STEP = 0.25 Units: Week [0,?] The time step for the simulation.
(05)	小売店からの受注=小売店の販売平均 Units: 個
(06)	小売店の販売=顧客の発注 Units: 個
(07)	小売店の販売平均=SMOOTH(小売店の販売, 平均期間) Units: 個
(08)	小売店への配送=製造量 Units: 個
(09)	小売店在庫= INTEG (小売店への配送-小売店の販売,400) Units: 個
(10)	工場受注残= INTEG (小売店からの受注-製造指示,200)

	Units: 個
(11)	平均期間=1
	Units: Week
(12)	理想とする製造量=工場受注残/目標とする製造期間
	Units: 個
(13)	目標とする製造期間=2
	Units: Week
(14)	製造指示=製造量
	Units: 個
(15)	製造期間=4
	Units: Week
(16)	製造量=SMOOTH(理想とする製造量 , 製造期間)
	Units: 個
(17)	顧客の発注=100+STEP(20 ,10)
	Units: 個

小売店の中には、この顧客に売れた分のみを補充するために発注するというやり方を行っているところもありますが、ちょっと大きな小売店では、店頭とは別にちょっとした倉庫を持っていますので、こういった発注はしないのが普通です。こういった発注方式をやっていると、発注をしそこねると、永久に商品が補充されない危険があり、店頭から売れる商品がなくなってしまう危険性があるからです。そこで、よく売れる商品に関しては、必ず予備の在庫を持ち、絶対に店頭から商品が無くならないようにします。ただ、ビール・ゲームでの経験では、この、注文を受けただけ発注するという方針はそう悪くない結果になっています。

(2) 2 番目のモデル：小売店の発注方式-2

最初のモデルでは在庫調整を考慮していなかったもので、ここでは、目標適正在庫を考え、それを維持するような発注をすることを考えます。工場受注残に計上する前の「小売店からの受注」を考えます。

小売店の先の顧客からの受注を補充するための発注に、さらに、標準在庫を維持するための発注として、「目標適正在庫」と「小売店在庫」を比べ、差があれば、「在庫調整期間」内で調整するようにしたものを加えます。在庫調整期間はここでは2週間とします。

「小売店の販売平均」で発注する量（これは、顧客に販売した分の補充分と考えていい）に加えてこの在庫調整分を上乗せして発注するので、式は、下のようになります。

$$\text{「小売店からの受注」} = \text{「小売店の販売平均」} + \frac{\text{「目標適正在庫」} - \text{「小売店在庫」}}{\text{「在庫調整期間」}}$$

図 6-6 にモデルを、図 6-7 にシミュレーション結果を示していますが、結果は、この在庫調整を入れると在庫の変動が激しくなり、前よりも悪くなっています。

念のため、2 番目のモデルの等式を示しておきます。06 行目の小売店からの受注の式が、小売店からの受注=小売店の販売平均+(目標適正在庫-小売店在庫)/在庫調整期間のように書き変わります。

(01)	FINAL TIME = 50
	Units: Week
	The final time for the simulation.
(02)	INITIAL TIME = 0
	Units: Week
	The initial time for the simulation.
(03)	SAVEPER = TIME STEP
	Units: Week [0,?]
	The frequency with which output is stored.

(04)	TIME STEP = 0.25 Units: Week [0,?] The time step for the simulation.
(05)	在庫調整期間=2 Units: Week
(06)	小売店からの受注=小売店の販売平均+(目標適正在庫-小売店在庫)/在庫調整期間
(07)	小売店の販売=顧客の発注 Units: 個
(08)	小売店の販売平均=SMOOTH(小売店の販売, 平均期間)
(09)	小売店への配送=製造量 Units: 個
(10)	小売店在庫= INTEG (小売店への配送-小売店の販売,400) Units: 個
(11)	工場受注残= INTEG (小売店からの受注-製造指示,200) Units: 個
(12)	平均期間=1 Units: Week
(13)	理想とする製造量=工場受注残/目標とする製造期間
(14)	目標とする製造期間=2 Units: Week
(15)	目標適正在庫=400 Units: 個
(16)	製造指示=製造量 Units: 個
(17)	製造期間=4 Units: Week
(18)	製造量=SMOOTH(理想とする製造量, 製造期間)
(19)	顧客の発注=100+STEP(20 ,10) Units: 個

2番目のモデル

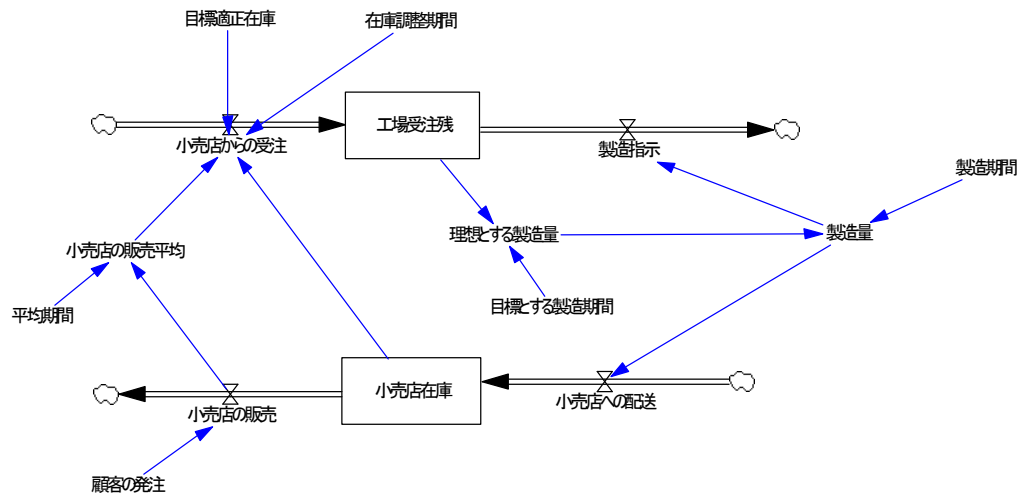


図 6-6 : 2 番目のモデル

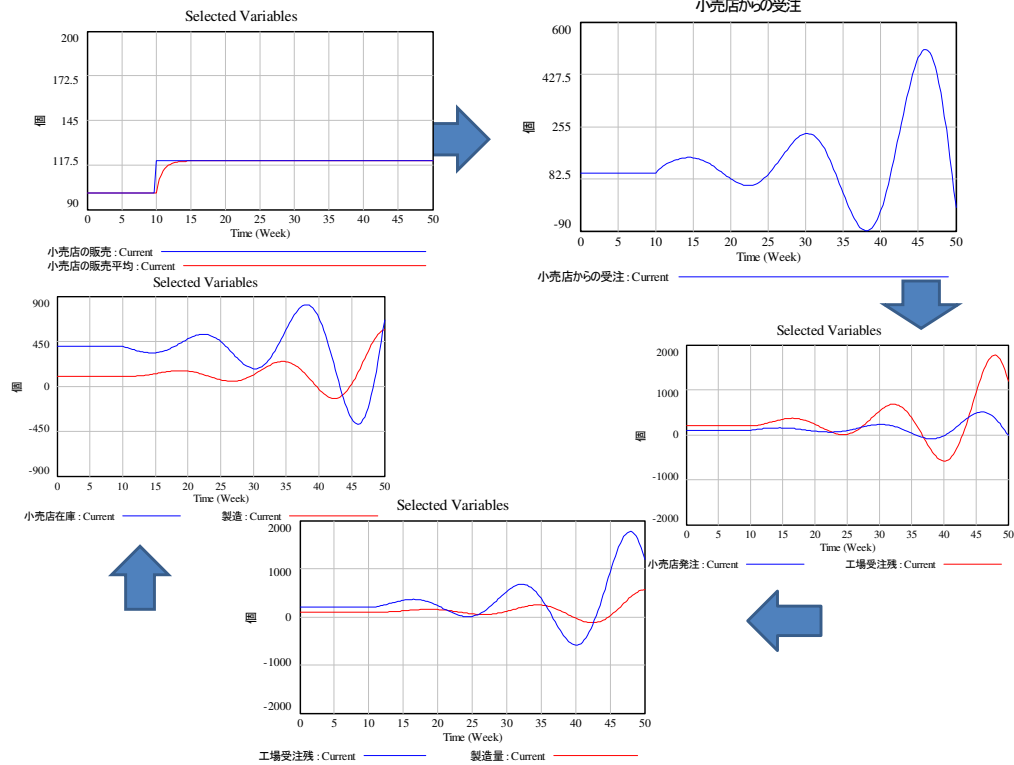


図 6-7 : 2 番目のモデルのシミュレーション結果

(3) 3 番目のモデル：小売店の発注方式-3

2 番目のモデルで在庫調整を考慮したら、事態が悪くなりました。小売店の販売平均（顧客に売った分の補充）を上乗せしたのが悪かったのではないかと、在庫調整だけを考えればいいのかという仮説を立て、3 番目のモデルでは、在庫調整だけに行ってみました。

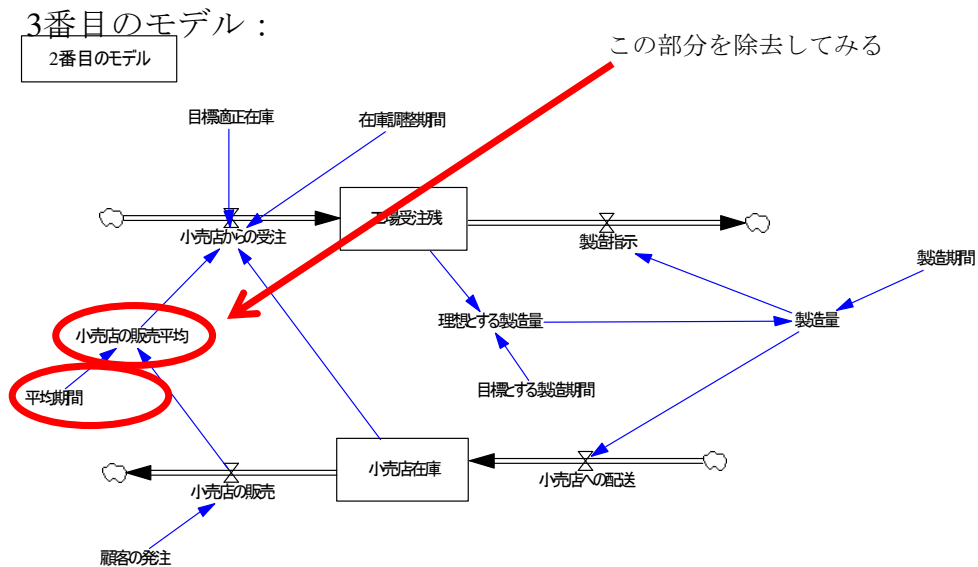


図 6-8 : 2 番目のモデルに対する仮説

3番目のモデル：

3番目のモデル

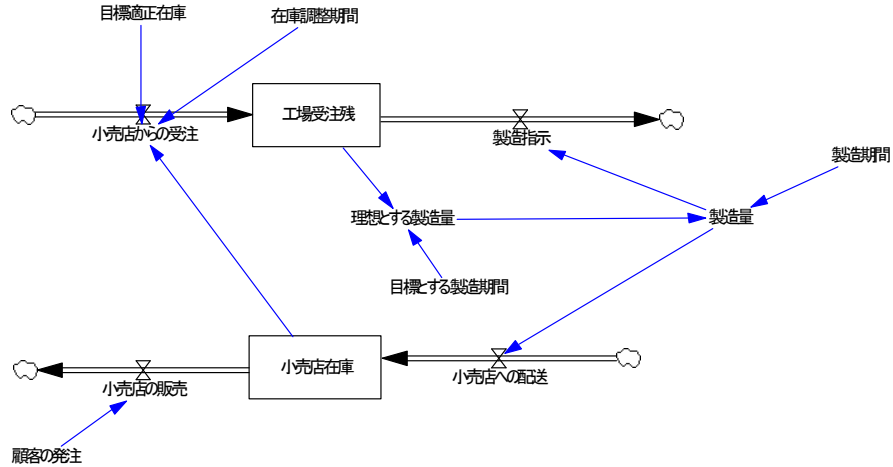


図 6-9：3 番目のモデル

図 6-8 に示した、2 番目のモデルの赤で囲った、顧客発注の補充分を取り外してみます。それが、図 6-9 に示した 3 番目のモデルです。このモデルでは、純粋に標準在庫を保つだけにします。2 番目のモデルでのシミュレーション結果よりも、3 番目のモデルのシミュレーション結果はさらに悪くなっています。

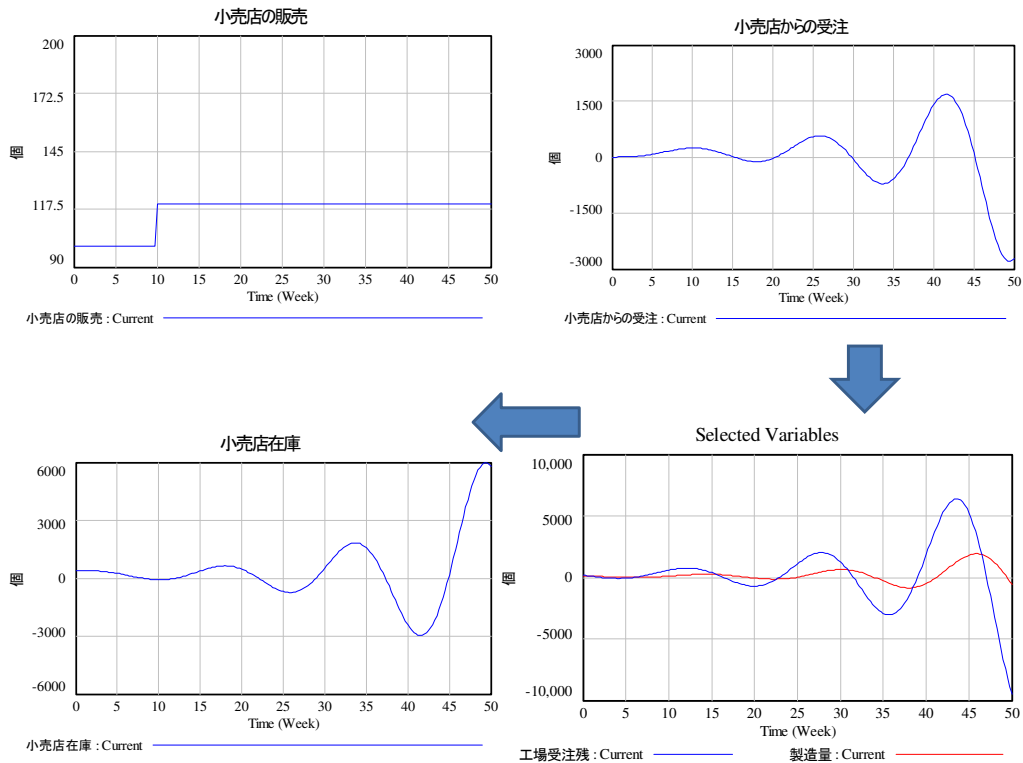


図 6-10：3 番目のモデルでのシミュレーション結果

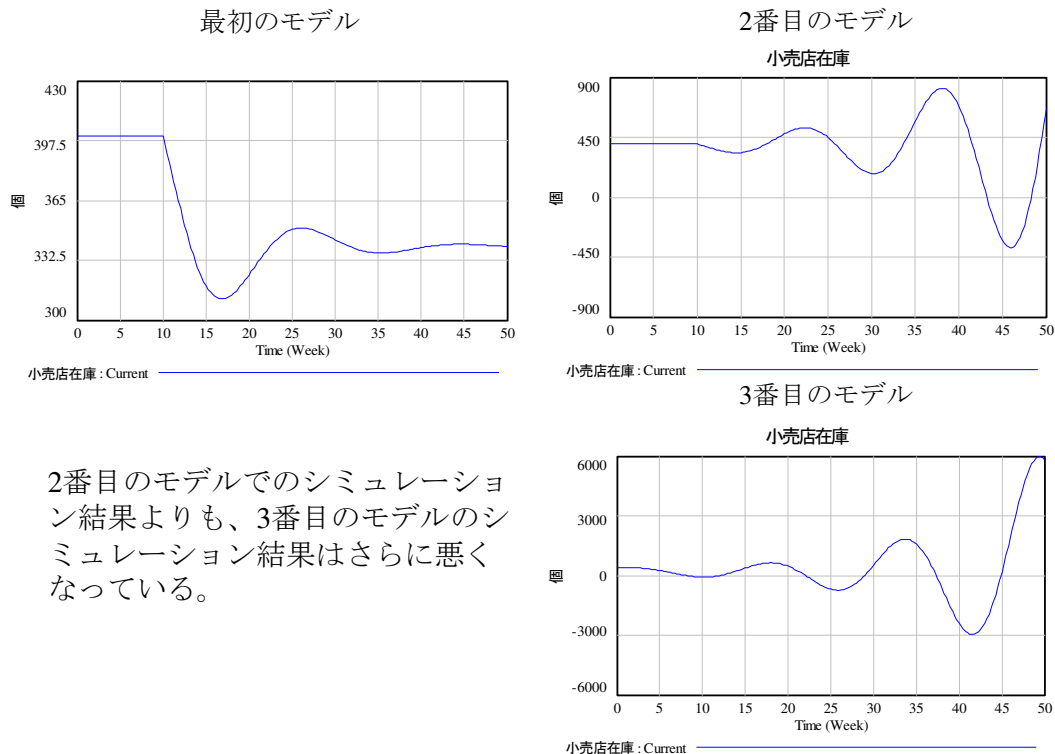


図 6-11：3 つのモデルでのシミュレーション結果の比較

小売店は適正在庫として 400 個を保とうとします。しかし、発注から入荷までの遅れがあり、その間を在庫調整しようとするので、発注がどんどん膨れ上がってしまいます。

通常のビジネスでは、受発注は継続していて、常に発注され、常に受注されています。これはパイプライン効果と呼ばれているものです。しかし、このモデルではそのことを考慮していないで、単独の受発注として取り扱っています。そこで、4 番目のモデルとして、連続受発注がなされていて、その上で、小売は、在庫調整のための分と、顧客に売った分の補充分だけを工場に発注するとします。工場は、連続受発注を考慮して製造し、その上で、小売店から受けた注文を加えて、受注残にし、受注残から製造していくとします。

このパイプライン効果は少し分りにくいかもしれませんね。図 6-1 に示した、コンセプトual・モデルの式を見てみましょう。

- (05) 小売店からの受注 = 小売店販売
- (06) 小売店入庫 = 製造
- (07) 小売店在庫 = INTEG (小売店入庫 - 小売店販売, 400)
- (08) 小売店販売 = 顧客発注
- (09) 工場出庫 = 製造
- (10) 工場受注残 = INTEG (小売店からの受注 - 工場出庫, 200)
- (11) 製造 = 100
- (12) 顧客発注 = 100

ここには、顧客発注の変動も、小売店の変な平滑化も、工場での平滑化もありません。従って、シミュレーション結果は、均衡状態になり、何も変化しない定常状態になります。

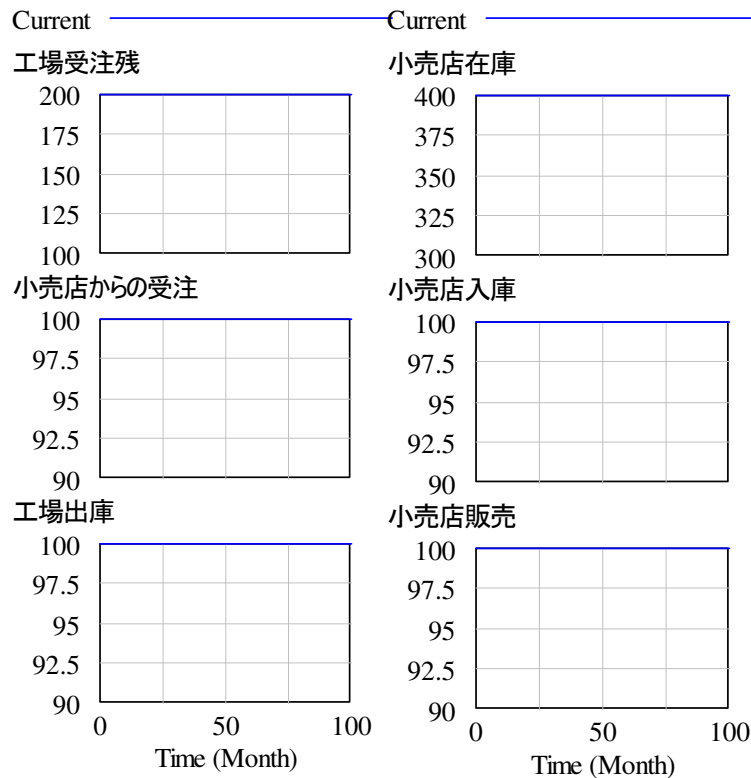


図 6-12：コンセプトチュアル・モデルのシミュレーション結果

図 6-12 にそのシミュレーション結果を示していますが、確かに均衡状態で、何も変化していません。最初のモデルも、2 番目と 3 番目も、このコンセプトチュアル・モデルをベースに作成していますので、本来であれば、モデルはこのように均衡するはずのものです。均衡しているのは、入荷分と在庫構成分、顧客に売った分の補充を均衡状態に保つように発注しているわけです。そのためのことを忘れ、顧客からの注文の補充分と、在庫調整分の発注だけを考えていたわけです。

ただ、実際の小売店や卸などでは、この在庫調整のみを考えて発注する方法はよくある話で、どう売れようと、とにかく標準在庫のみを維持できるように考えて発注します。特に、仕入れが在庫管理も兼任しているような会社では、この発注方式を採択しているところが多いように見受けられます。

念のため、3 番目のモデルの等式を示しておきます。

(01)	FINAL TIME = 50
(02)	INITIAL TIME = 0
(03)	SAVEPER = TIME STEP
(04)	TIME STEP = 0.25
(05)	在庫調整期間=2
(06)	小売店からの受注=(目標適正在庫-小売店在庫)/在庫調整期間
(07)	小売店の販売=顧客の発注
(08)	小売店への配送=製造量
(09)	小売店在庫= INTEG (小売店への配送-小売店の販売,400)
(10)	工場受注残= INTEG (小売店からの受注-製造指示,200)

- | | |
|------|----------------------------|
| (11) | 理想とする製造量=工場受注残/目標とする製造期間 |
| (12) | 目標とする製造期間=2 |
| (13) | 目標適正在庫=400 |
| (14) | 製造指示=製造量 |
| (15) | 製造期間=4 |
| (16) | 製造量=SMOOTH(理想とする製造量, 製造期間) |
| (17) | 顧客の発注=100+STEP(20,10) |

(4) 4 番目のモデル：工場側でのパイプライン効果の取り入れ

モデルを 2 番目のものに戻し、「望ましい連続発注量」、「望ましい連続発注にするための調整期間」、「小売店発注の時間遅れ」を追加します。

「望ましい連続発注量」は、変動がなければ常に、小売店に工場から配送する量ですが、変動しますので、平滑化した顧客の発注を時間遅れを考慮して、量を増やしたものとします。すると式は：

$$\text{「望ましい連続発注量」} = \text{「小売店の販売平均」} \times \text{「小売店発注の時間遅れ」}$$

となり、これは、工場から見ると、望ましい製造量とほぼ同じになります。

工場側は、工場の受注残と「望ましい連続発注量」を比較して、期間調整を行い、製造指示の量にするはずですが、従って工場は受注残に計上する量として：

$$(\text{「望ましい連続発注量」} - \text{「工場受注残」}) / \text{「望ましい連続発注量にするための調整期間」}$$

をベースに、小売店が実際に行う発注（小売店が行う在庫調整分の発注と顧客に売った分の補充分）を加えて、受注残に計上するはずですが、

調整期間は在庫調整の期間と同じ 2 週間とします。

4 番目のモデル：

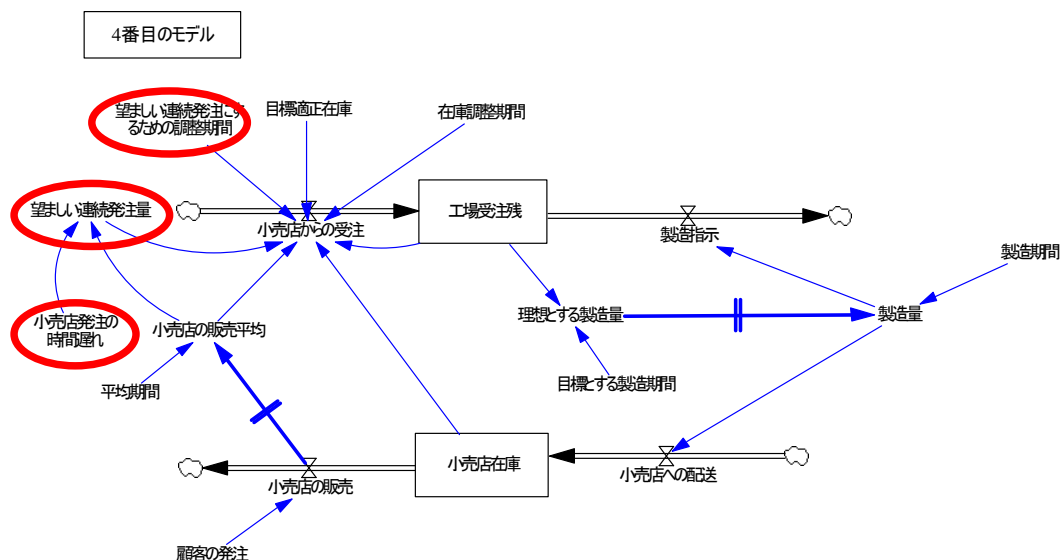


図 6-13：第 4 のモデル：パイプライン発注を加える

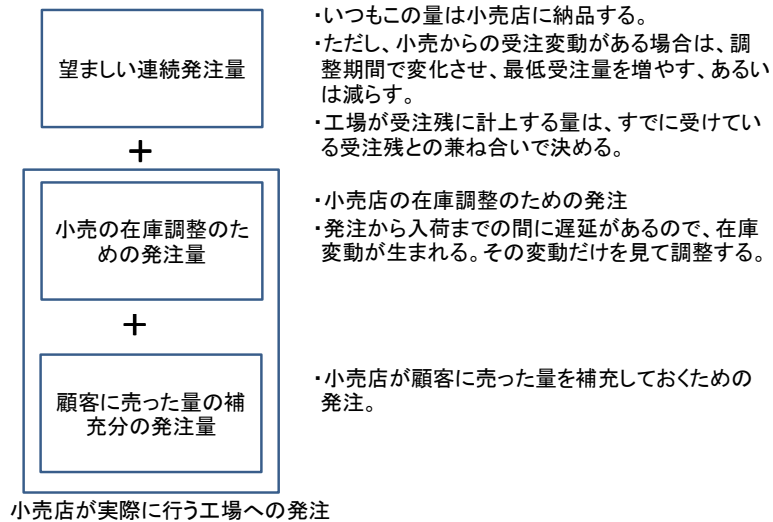


図 6-14：パイプライン発注

シミュレーション結果は、顧客の発注変動に対応して、工場の小売店からの受注は一時的に変動しますが、前のものに比べ安定し、連続受発注に近くなっています。

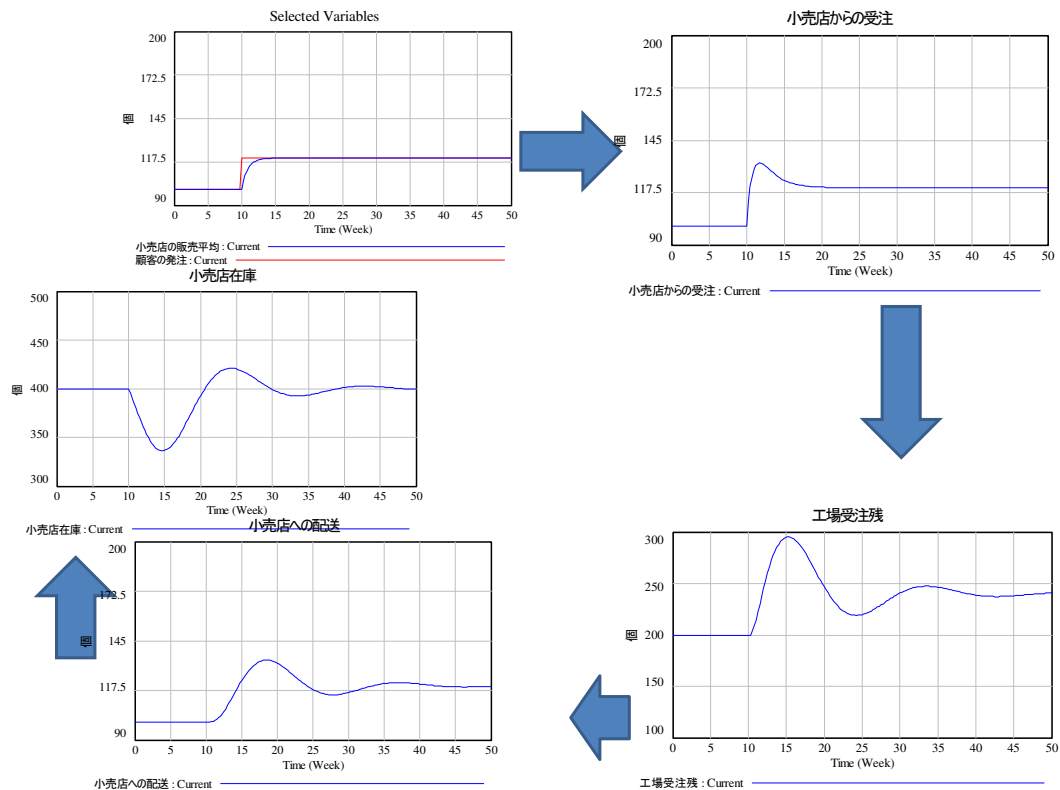


図 6-15：第 4 のモデルのシミュレーション

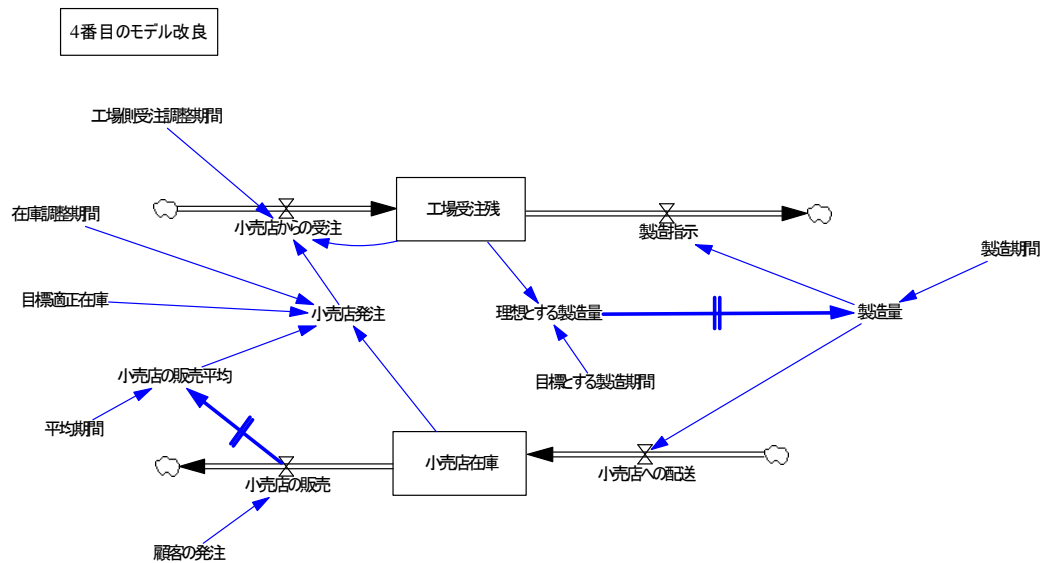


図 6-16：第 4 のモデルの変更

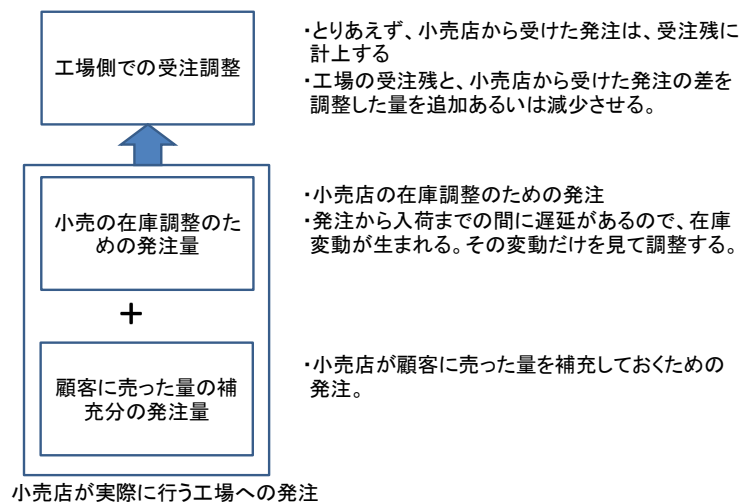


図 6-17：第 4 のモデルの変更

ただ、私の個人的感想ですが、通常は、先のようにまでは考えないように思います。むしろ、小売店は、売れた分の補充と在庫調整分のみを発注し、工場側は、小売店から受けた注文を、工場受注残の勘案し、工場側の受注調整期間で調整し、受注残に計上するのが普通のように思います。そうすると、第 4 のモデルは、図 6-16 のようになり、シミュレーション結果は、図 6-18 のようになります。

パイプライン発注を考慮した第 4 のモデルの方が受発注や在庫は安定しますが、通常はこちらの形に近い変化になると思います。

あるいは、小売店での発注で、定常発注量を考慮する場合、工場側でどう調整しているかは小売店側では不明なので、把握できる工場からの入荷数を平準化し、その数量を定常発注量とします。そして、顧客に販売した商品の補充及び在庫調整のための発注と加えて、工場に発注します。ただ、工場からの入荷数を平準化しただけだと、過剰発注になるので、係数をかけて、数量を減らします。同じく、在庫調整のための発注もそのままですと過剰

発注になってしまうので、やはり係数をかけて調整し、発注が平準化されるように工夫します。図 6-19 はそのようにしたモデルで、図 6-20 はそのシミュレーション結果です。こちらの方が、図 6-16 のシミュレーション結果に近くなっていると思います。ただ、この方法では、適切な発注係数を決めることが少し難しいかも知れません。

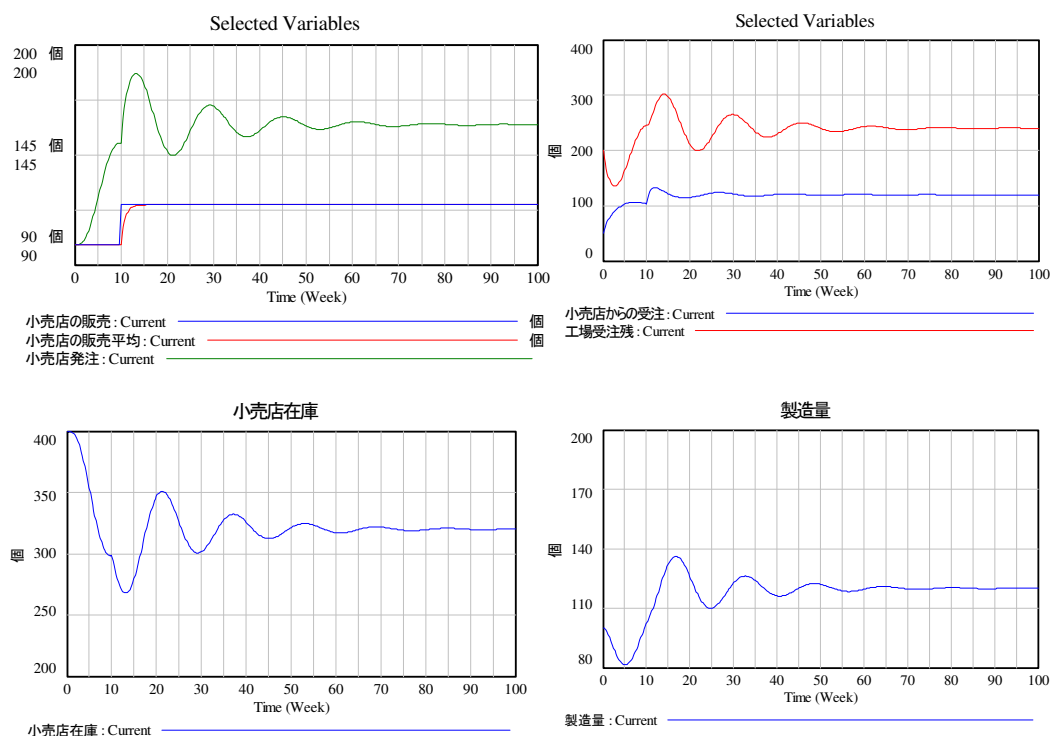


図 6-18：変更した第 4 のモデルでのシミュレーション

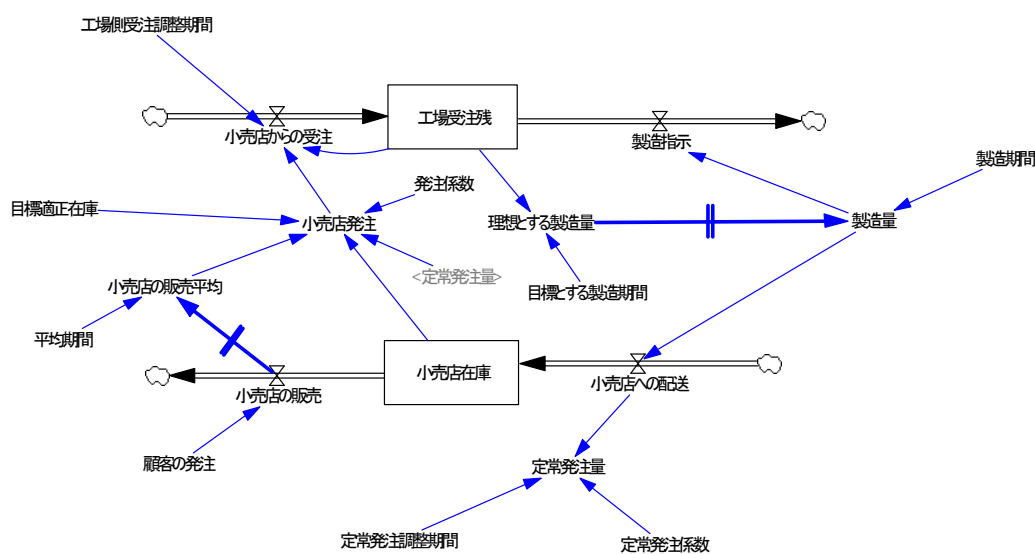


図 6-19：別の第 4 のモデルの変更

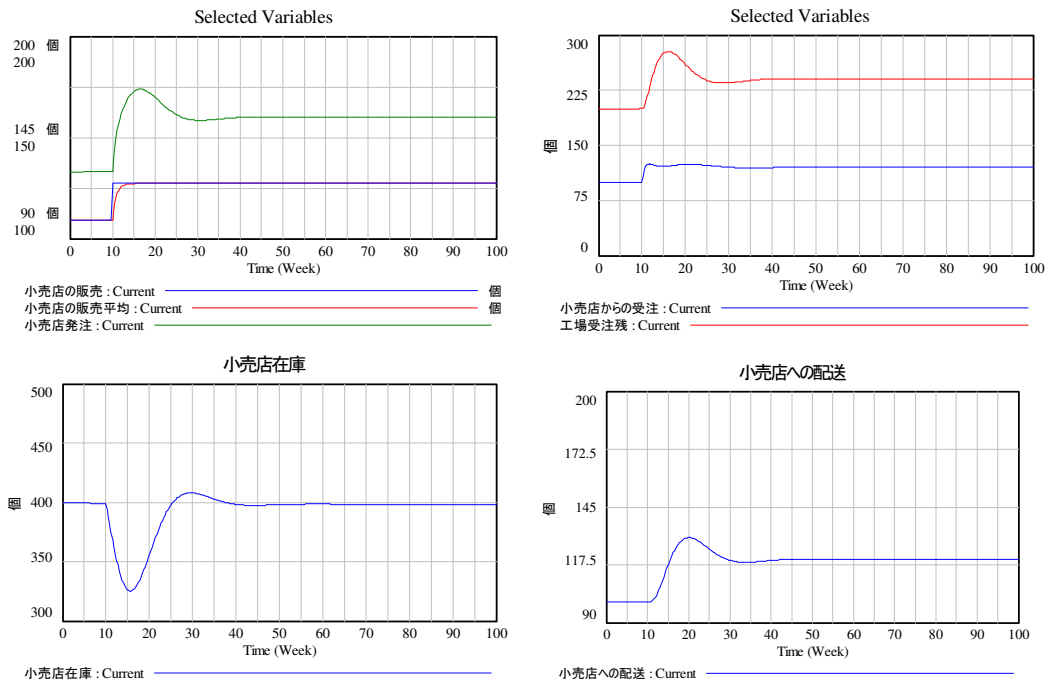


図 6-20：別の第 4 のモデルを変更したもののでのシミュレーション結果

念のため、第 4 のモデルの等式を示しておきます。第 6 式が複雑になっています。

(01)	FINAL TIME = 50
(02)	INITIAL TIME = 0
(03)	SAVEPER = TIME STEP
(04)	TIME STEP = 0.25
(05)	在庫調整期間=2
(06)	小売店からの受注=小売店の販売平均+(目標適正在庫-小売店在庫)/在庫調整期間 +(望ましい連続発注量-工場受注残)/望ましい連続発注にするための調整期間
(07)	小売店の販売=顧客の発注
(08)	小売店の販売平均=SMOOTH(小売店の販売, 平均期間)
(09)	小売店への配送=製造量
(10)	小売店在庫= INTEG (小売店への配送-小売店の販売,400)
(11)	小売店発注の時間遅れ=2
(12)	工場受注残= INTEG (小売店からの受注-製造指示,200)
(13)	平均期間=1
(14)	望ましい連続発注にするための調整期間=2
(15)	望ましい連続発注量=小売店発注の時間遅れ*小売店の販売平均
(16)	理想とする製造量=工場受注残/目標とする製造期間
(17)	目標とする製造期間=2
(18)	目標適正在庫=400
(19)	製造指示=製造量
(20)	製造期間=4
(21)	製造量=SMOOTH(理想とする製造量, 製造期間)
(22)	顧客の発注=100+STEP(20 ,10)

(5) 5 番目のモデル：送品納入遅れを予想しての発注

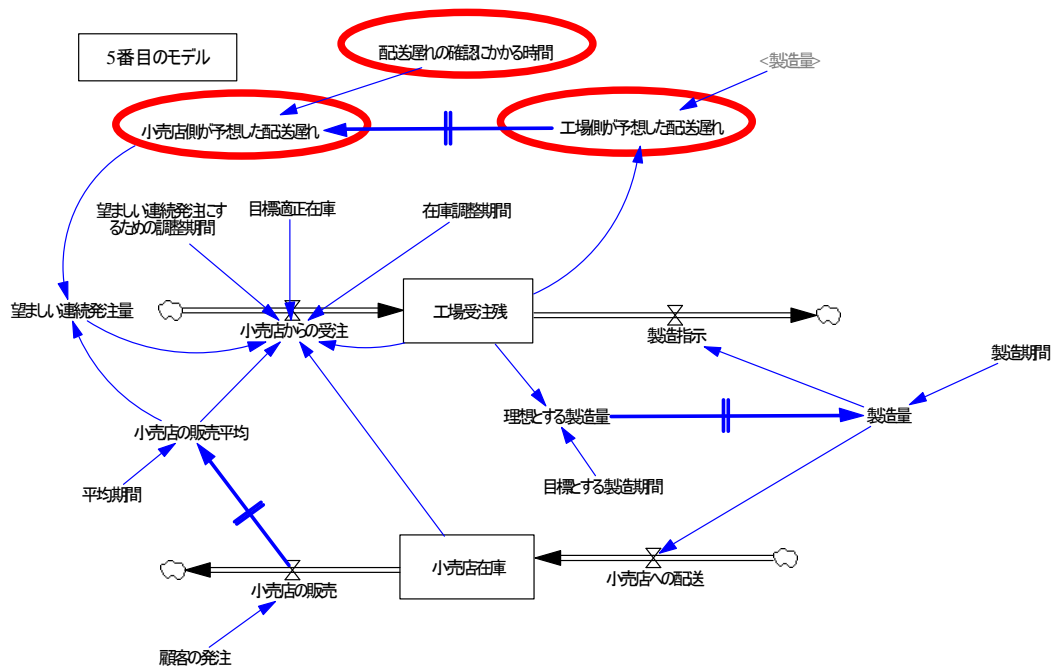


図 6-21：第 5 のモデル：納期遅れを予想して発注

4 番目のモデルでは、望ましい連続発注量として、平滑化した顧客からの注文を単純に、発注の時間遅れを見越して、その分増やしていただけなので、ここをもっと精緻化し、配送遅れを予測し、それに対応するような発注を考えてみます。

配送遅れは、工場がどれだけ受注残を抱えているかに関係していて、受注残を抱えれば抱えるほど、製造遅れのため、配送が遅れることが予想されます。この遅れる係数は：

$$\text{「予想している配送遅れ」} = \text{「工場受注残」} / \text{「製造量」}$$

となります。さらに、小売店側にとってみれば、配送遅れの確認に時間がかかり、この予想はもっと遅延します。工場受注残も製造量も連続している話なので、計算は平滑化変数を使って計算します。配送遅れの確認時間で平滑化した係数で望ましい発注量を増やすとします。そして、配送遅れの確認に 2 週間かかるとします。モデルを図 6-21 に、シミュレーション結果を図 6-22 に示しています。受注残と製造量との関係での製造負荷からくる配送遅れに対し、それを予想して調整するようしくみを組み込んだら、前のよりも悪くなり、周期変動するようになっていきます。この例が示すように、予測して何か対策を取るということは必ずしもうまくいくとは限らないものです。

実際のビール・ゲームでも、だれかが、何かを予想して発注量を調整すると、それが変動要因になって、変動が拡大するという現象がよく見られます。

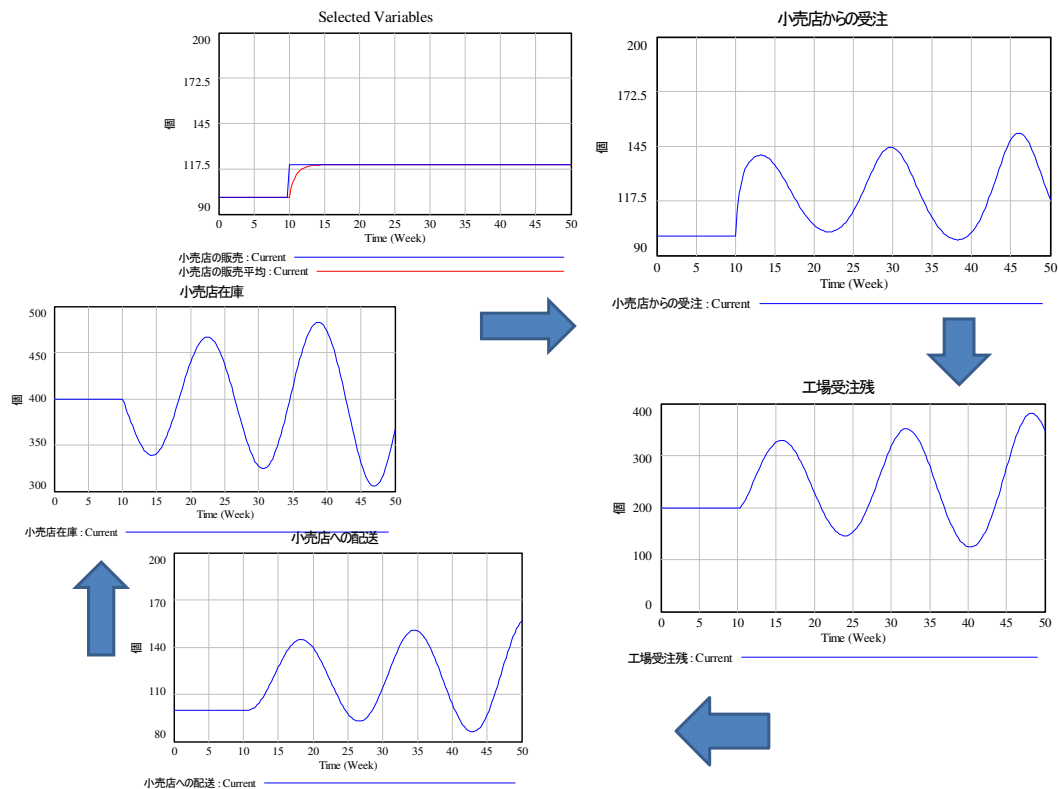


図 6-22 : 第 5 のモデルのシミュレーション結果

念のため、第 5 のモデルの等式を示しておきます。

- | | |
|------|--|
| (01) | FINAL TIME = 50 |
| (02) | INITIAL TIME = 0 |
| (03) | SAVEPER = TIME STEP |
| (04) | TIME STEP = 0.25 |
| (05) | 在庫調整期間=2 |
| (06) | 小売店からの受注=小売店の販売平均+(目標適正在庫-小売店在庫)/在庫調整期間
+(望ましい連続発注量-工場受注残)/望ましい連続発注にするための調整期間 |
| (07) | 小売店の販売=顧客の発注 |
| (08) | 小売店の販売平均=SMOOTH(小売店の販売, 平均期間) |
| (09) | 小売店への配送=製造量 |
| (10) | 小売店側が予想した配送遅れ=SMOOTH(工場側が予想した配送遅れ,配送遅れの
確認にかかる時間) |
| (11) | 小売店在庫= INTEG (小売店への配送-小売店の販売,400) |
| (12) | 工場側が予想した配送遅れ=工場受注残/製造量 |
| (13) | 工場受注残= INTEG (小売店からの受注-製造指示,200) |
| (14) | 平均期間=1 |
| (15) | 望ましい連続発注にするための調整期間=2 |
| (16) | 望ましい連続発注量=小売店側が予想した配送遅れ*小売店の販売平均 |
| (17) | 理想とする製造量=工場受注残/目標とする製造期間 |
| (18) | 目標とする製造期間=2 |
| (19) | 目標適正在庫=400 |
| (20) | 製造指示=製造量 |

- | | |
|------|------------------------------|
| (21) | 製造期間=4 |
| (22) | 製造量=SMOOTH(理想とする製造量 , 製造期間) |
| (23) | 配送遅れの確認にかかる時間=2 |
| (24) | 顧客の発注=100+STEP(20 ,10) |

(6) 受注変動

顧客発注に変動があることは普通のことです。ここでは2番目と4番目のモデルで、100個を基準にプラスの20個を乱数変化させて影響を見てみます。2番目の19式、4番目の22式を以下のように書き換えます。STEPで20、顧客は発注量を増やしていますが、この増加分が、乱数によって0から1まで変化します。

顧客の発注=100+STEP(20 ,10)*RANDOM UNIFORM(0 , 1 , 0)

図 6-23、6-24 に結果を示していますが、顧客への販売を乱数変化させても、全体的な傾向に変化があるわけではなく、ただ、小さな変化として現れるだけであることが分かります。2番目のモデルでは、拡散現象が現れていますが、これは、顧客の注文変化というよりは、小売店と工場で形成されるしくみが生み出したものであることが、このことから分かるといえます。つまり、外部からの何かがこういった変化を作り出したのではなく、業務プロセスである小売店の発注と工場の受注のしくみでこういった変化になっています。

このように、いろいろモデルをいじり、その都度シミュレーションを行って結果を確かめながら深化させていき、現実に近い状態を十分表現できるようになったと思われる時点で、完成とします。残念ながら、モデル深化に何か標準的なやり方があるわけではなく、モデルの完成度に関しても、何か明確な基準があるわけではありません。ただ、モデルに改善を加えていくやり方として、通常は、以下のようにします。

- 1)概念モデルと呼ばれる、そこから始まるような簡単なモデルを作る。このモデルは、特にシミュレーションできなくともいいが、対象にしているものを解釈した骨組や主要要素が含まれている必要がある。
- 2)主要機能に分けて考え、主要機能毎に精緻化していく。
- 3)モデルがあまりにも複雑になるようであれば、主機能別にモジュールという形で分けていく。

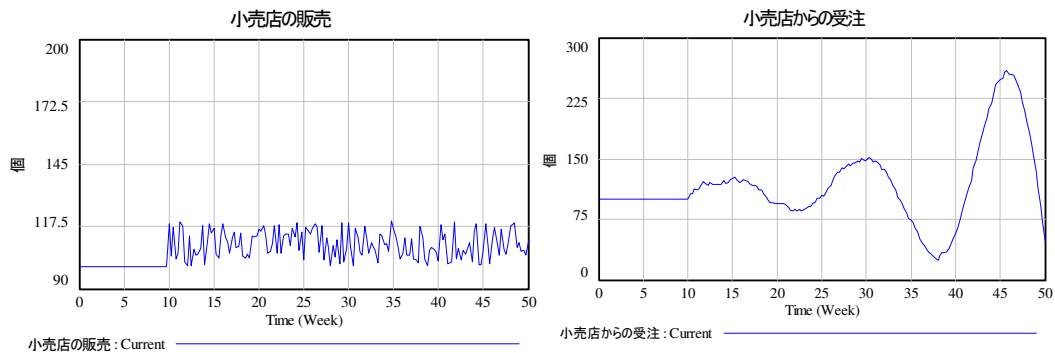
この考え方に従えば、このモデルでは、小売店、工場に分け、小売店は、顧客からの受注、発注、在庫調整に、工場は、小売店からの受注、受注残監理、製造、配送に分け、詳細化していきます。

小売店では、発注も重要ですが、通常は、標準在庫、安全在庫という2段階の在庫を行うので、標準在庫から減ったら、減った分を単純に発注するのだけではなく、安全在庫を切ったら多めに発注するというメカニズムになります。

顧客からの受注に関しては、通常は平滑化するし、移動平均を使った発注をします。移動平均ではなく、最近売れるようになったから増やすかという感覚でやっている、いわゆる、認知の遅れで処理しているモデルもよく見ます。

ここでは、受注生産（工場在庫を持たない）としているので、工場在庫管理は考えていませんが、そういった製造方法もあるので、これでいいでしょう。ただ、現実には工場在庫の部分付け加えられるべきでしょう。

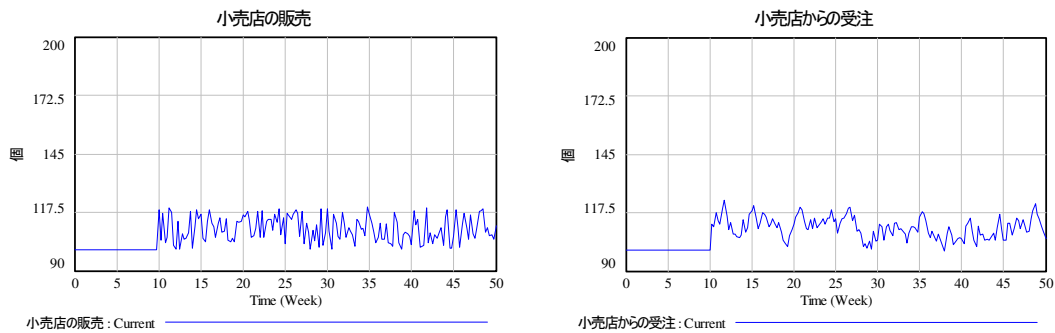
このモデルでは受注から製造に持っていくやり方、もっと言えば、製造平滑化しか考えていませんが、通常は、製造プロセス、配送プロセスも重要になります。



発注変動：

現実には、顧客の注文はばらばらで、乱数的と考えられるので、そういった乱数的変化を、顧客の小売店への発注パターンに加えてみた。
ここでは2番目のモデルで、100個を基準にプラスの20個を乱数変化させてみた。

図 6-23：第 2 のモデルで顧客の発注に乱数変化を加えたもののシミュレーション結果



発注変動：

ここでは4番目のモデルで、100個を基準にプラスの20個を乱数変化させてみた。

図 6-24：第 4 のモデルで顧客の発注に乱数変化を加えたもののシミュレーション結果

工場受注では、このモデルのような感じで、小売店からの受注に、受注残を勘案して製造計画を立てるのが普通のやり方です。工場の基本的な考え方は、製造の平滑化で、受注量の急激な変化には耐えられないので、なるべく同じ量を同じ製造期間で製造しようとしています。この際のテクニックが平滑化と移動平均です。受注残というストックに受注をフローで入れていき、製造をフローで出していく。そういった状態でのストックを製造期間で割ると、まさしく移動平均に近いものとなります。ここではさらに製造時に、製造指示を平滑しています。

このモデルでは範囲外ですが、工場での製造、配送では、ボトルネック改善とスループット改善が重要視されます。ただ、その改善手法に関しては、フロー・チャートや定性モデル、PERT で十分だと思います。

どこまでモデルを精緻化させたら完成なのかという基準もないので、SD のモデリングはどこまでいっても未完成品のようなものかも知れません。一応、データが存在する過去から現在までの実際のデータとシミュレーション結果とを比べて、誤差が例えば 3-5%以内であれば完成にするといった方針で開発することもあります。ただ、SD の目的が、対象のメカニズムをいかにうまく説明できるかにあり、どこまで正確に何かを予測できるかにはないので、別に誤差が先の範囲に収まっていなくとも、傾向さえ説明できていればそれでいいように思います。

(7) まとめ

ここでは、小売店と工場だけの簡単なモデルで、受発注と在庫安定を見てきました。モデル構築では、最初にコンセプトチャール・モデルを構築し、それを土台にシミュレーション結果を見ながら現実的になるように機能を精緻化し、モデルの完成度を高めていきます。

7. 遅延

(1) 純な遅延 (pipeline delay)

単純な遅延はパイプライン型遅延と呼ばれていて、DELAY FIXED 関数で表現します。これは、単純に入力が遅延して出力になるだけです。定義式は：

DELAY FIXED (値、遅れ時間、遅れ発生までの初期値)

となります。第 1 引数が、遅れの後に取り値、第 2 引数が遅れ時間、第 3 引数が初期値です。第 1 引数は変数で構いませんが、第 2 引数の遅れ時間は定数になります。

図 7-1 に単純な遅延モデルを示しています。入力遅延では 10 週目に 100 から 120 に数量を変化させています。これを、出力側遅延が、遅延時間を 10 として受けます。ここでは DELAY FIXED を使っているの、単純に 10 週遅れるだけです。従って、出力側遅延では 20 週目に 100 から 120 になります。

このモデル式は以下の通りです。

(01)	FINAL TIME = 40
(02)	INITIAL TIME = 0
(03)	SAVEPER = TIME STEP
(04)	TIME STEP = 0.5
(05)	入力=100+STEP(20, 10)
(06)	入力側ストック= INTEG (-入力側遅延,10000)
(07)	入力側遅延=入力
(08)	出力側ストック= INTEG (出力側遅延,0)
(09)	出力側遅延=DELAY FIXED(入力側遅延, 遅延時間,入力側遅延)
(10)	遅延時間=10

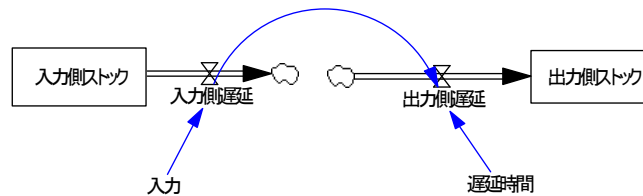


図 7-1：単純な遅延モデル

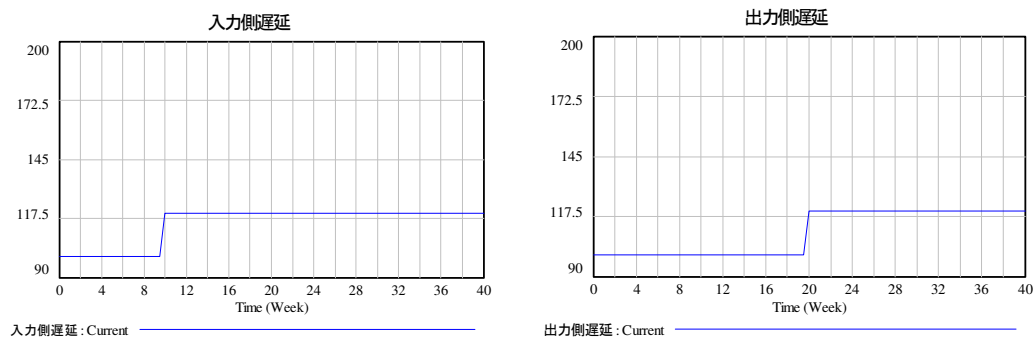


図 7-2：単純な遅延モデルのシミュレーション結果

これに対し、遅れ時間が変数である場合は DELAY MATERIAL を使います。2 項目が変数であること以外は DELAY FIXED と同じです。例えば、今、工場の生産性が従業員の数で決まるとします。入力に変動（生産指示での生産量など）があり、入力側は、遅延無しで、出力側遅延にこれを伝えます。出力側遅延は、ここでは、生産量と考えて下さい。そして、生産量は従業員の数で決まります。現在 5 人で、30 週後に 90 人に増やします。当然、生産性は上がるのですが、問題は周期です。周期変動があるので、人数が増えると、製品が早く製作でき、周期が短くなります。

出力側遅延=DELAY MATERIAL(入力側遅延 ,人数の変動による作業時間の変動, 入力側遅延, 入力側遅延)

人員数=5+STEP(90,30)

人数の変動による作業時間の変動=30/人員数

DELAY MATERIAL を使った例で、30 週目から人員が増え、遅延のサイクルが短くなっています。

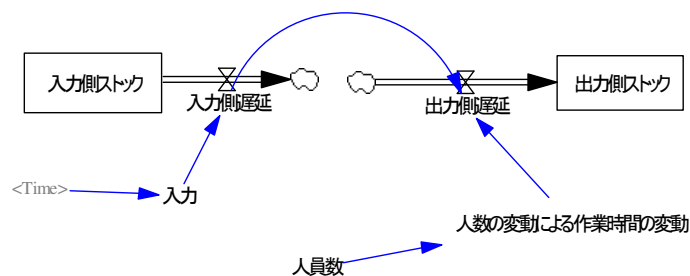


図 7-3 : DELAY MATERIAL のモデル

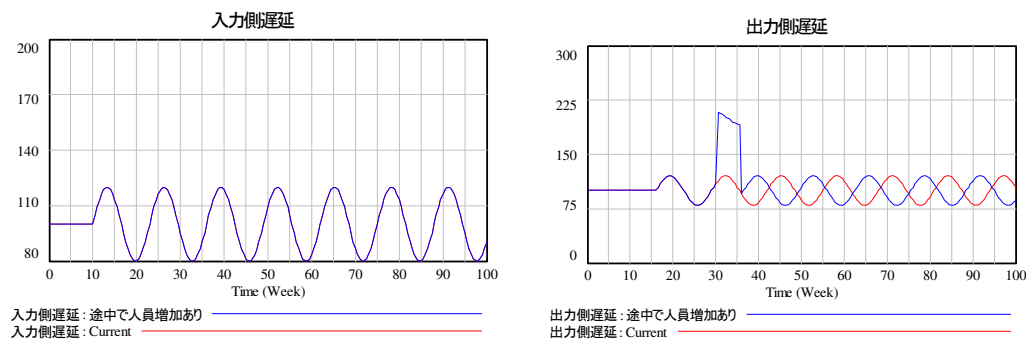


図 7-4 : DELAY MATERIAL のモデルのシミュレーション

(2) 指数遅れ

1) 3 次の指数的遅延 (third order exponential delay)

先のは、単純に遅れるだけでしたが、遅れには、別に指数遅れという遅れのタイプもあります。例えば、仕掛品（あるいは部品）が仕上がっていった商品になっていくといった状況での遅延や、生産指示で生産量を急に増やしても、工場側で急に生産を増やせるわけではなく、ある期間をかけて成長曲線的に増えていくといった遅延のパターンです。あるいは、広告を打って、その反応が、広告を出した直前には大きくありますが、だんだん日がたつにつれて減っていくといった際の遅延のパターンです。増加の場合は成長曲線に、減少の場合は負の指数減少、いわゆる収斂と呼ばれている曲線変化をします。

こういった変化は、DELAY3 関数で表現します。この関数を使う場合は、Vensim PLE は、

遅延までの間は、入力値と変わらないとして、入力値の変化が指数的に処理されますので、先のパイプライン遅延と違い、初期値を設定しません。もし、遅延までの遅延量が違う場合は、DELAY3I を使い、初期値を設定して下さい。

図 7-1 のモデルの 09 式での DELAY FIXED を、DELAY3 に置き換えます。

(01)	FINAL TIME = 40
(02)	INITIAL TIME = 0
(03)	SAVEPER = TIME STEP
(04)	TIME STEP = 0.5
(05)	入力=100+STEP(20, 10)
(06)	入力側ストック= INTEG (-入力側遅延,10000)
(07)	入力側遅延=入力
(08)	出力側ストック= INTEG (出力側遅延,0)
(09)	出力側遅延=DELAY3(入力側遅延, 遅延時間)
(10)	遅延時間=10

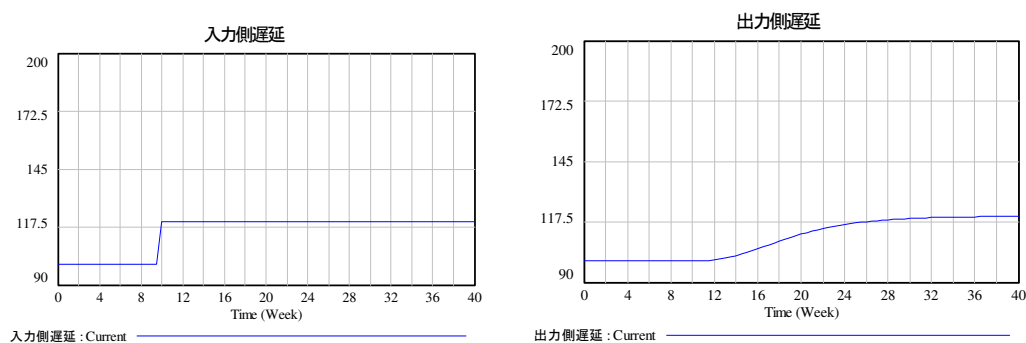


図 7-5 : DELAY3 のモデルのシミュレーション

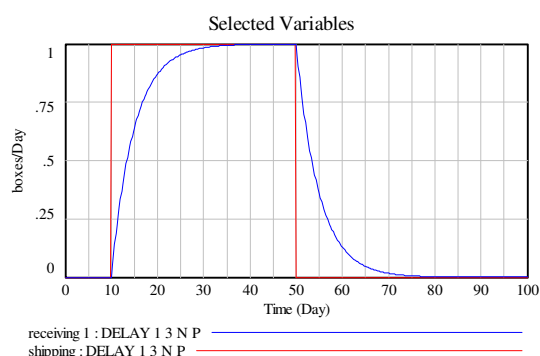
基本的には、図 7-1 と同じで、入力側遅延で 10 週目に 100 から 120 に変化させています。それを出力側遅延で受け、遅延時間を 10 としています。従って、10 週遅れ、20 週目から出力が変わるのではなく、10 週から指数平滑化され、10 週目から徐々に変化し、約 30 週以降 120 に安定します。

Vensim PLE では、一次と 3 次の指数遅れ関数が使えます。3 次の指数遅れでは、一次の指数遅れに比べ、より短い期間で平滑化されます。

指数遅れという関数は何をやっているか興味があると思います。1 次の指数遅れでは、増加の場合は、単に、ストックに値を累積し、それを調整時間で割っているだけであり、減少の場合は、ストックから値を引いていき、ストックを調整時間で割っているだけです。従って、指数変化になるので、指数遅れと呼ばれています。後で説明しますが、実は、平滑化も同じようなことをやっているしますので、場合によっては、指数遅れも平滑化も同じような結果になることがあります。

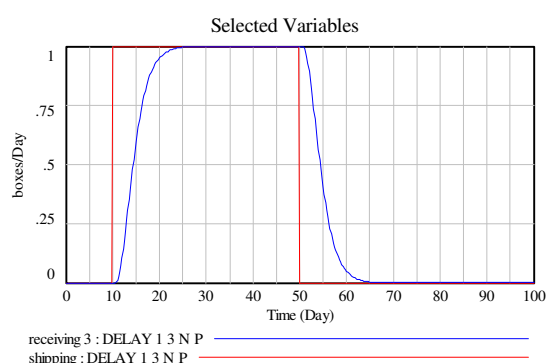
先に、DELAY3 を使うと、初期値は Vensim PLE が勝手に計算して決めてしまうと説明しました。モデルによっては、Vensim PLE で勝手に決めた初期値が使われると困ることがあります。また、モデルの作り方によって、例えば、ストックを挟まないで変数だけのループがあるようなモデルにすると、Vensim PLE は、初期値を自動的に設定できなく、実行時エラーになります。このようなことを避ける必要がある場合には、情報付き指数遅れ関数である、DELAYII や DELAY3I を使って下さい。図 7-8 に例を示していますが、遅れ側で、遅れが発生するまでの間、例えばゼロでは困る場合、ここでは 5 にしていますが、こうい

った指定にします。そして、10 日目から入力値である 10 を指数化して増加させています。



遅延のいろんな関数

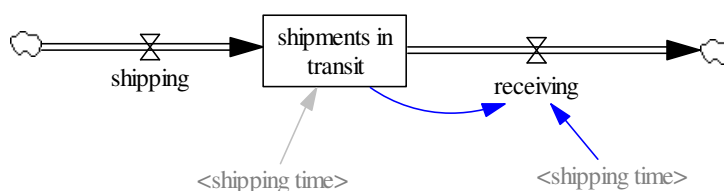
receiving 1=DELAY1(shipping,
shipping time)
一次の指数遅れ



receiving 3=DELAY3(shipping,
shipping time)
3次の指数遅れ

図 7-6 : DELAY1 と DELAY3 の比較

Equivalent 1st order structure



何をやっているかだが、1次の指数遅れでは、増加の場合は、単に、ストックに値を累積し、それを調整時間で割っている。
減少の場合は、ストックから値を引いていき、ストックを調整時間で割っているだけ。従って、積分しているのと同じになる。

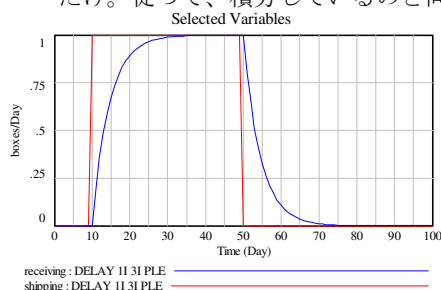


図 2-7 : DELAY1 の原理

遅延のいろんな関数

receiving 1=**DELAY1I**(shipping,
shipping time, shipping)

一次の指数情報遅れ

DELAY 1との違いは、遅延までの間の初期値を指定できるかできないかの違いだけ。このように、遅延までに間の初期値に遅延した後の量を指定すると、全く同じになってしまう。

receiving 3=**DELAY3I**(shipping,
shipping time, shipping)

3次の指数情報遅れ

DELAY 3との違いは、遅延までの間の初期値を指定できるかできないかの違いだけ。

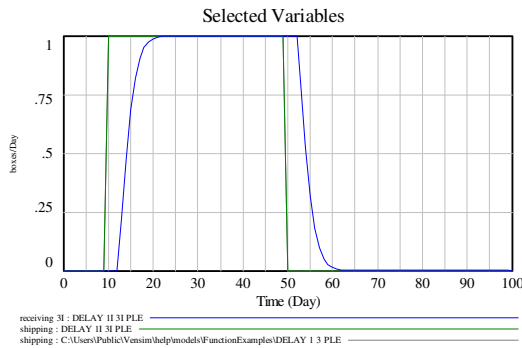


図 7-7 : DELAY1I と DELAY3I の比較

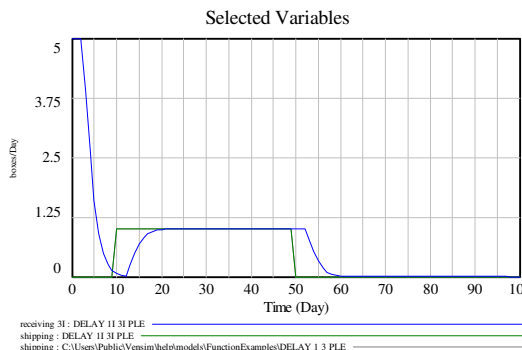
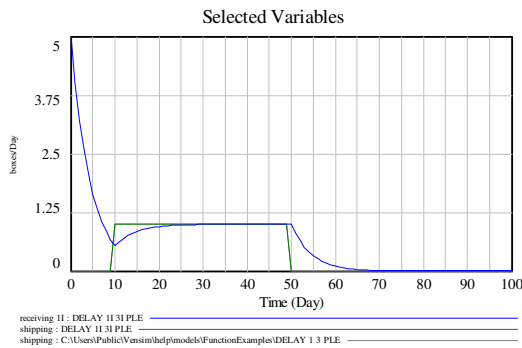


図 7-8 : DELAY1I と DELAY3I の比較

遅延発生までの初期値を指定できるので、ここでは5を指定している。従って、遅延発生までは、5からゼロまで、それぞれ、1次3次で指数変化し、その後、5日目から、変化量として指定されている1になり、50日間継続する。

receiving 1=**DELAY1I**(shipping,
shipping time, 5)

receiving 1=**DELAY3I**(shipping,
shipping time, 5)

(3) 振動

先の例は、一気に増えるというものですが、通常は、一気に増えるだけではなく、振動し、数値が上下に振れることがよくあります。この振動の周期が、指数型遅延との関連で

問題になることがあります。例として、ここでは、入力を4週周期(1ヶ月)、13週周期(季節)、52週周期(1年間)で変動させてみました。周期が大きくなるにつれ、変動幅が大きくなっていきます。また、周期は少しずつ後にずれていきます。これは工学用語ではフィルター(filters)と呼ばれている現象です。振動幅が小さい場合は、振動が取れるような形になるのですが、振動が大きい場合、振動は除去されず、さらに周期変動が起きます。

経営では時として、問題が引き起こす現象がずれて表面化することがありますが、このような場合の原因に、よくこのフィルター現象がからんでいることがあります。例えば、営業からの頻繁な緊急製造指示などが通常のサイクルの製造に紛れ込んで、ある時、原因がすぐに分からないラインの緊急ストップのようなことが起きます。このような事故の原因の一つが、通常であれば起きないのですが、たまたま、遅延によってサイクルの違うものがある所でぶつかってしまい、仕掛品の数がある所で突然不足してしまうなどの異常停滞を引き起こすことがあります。

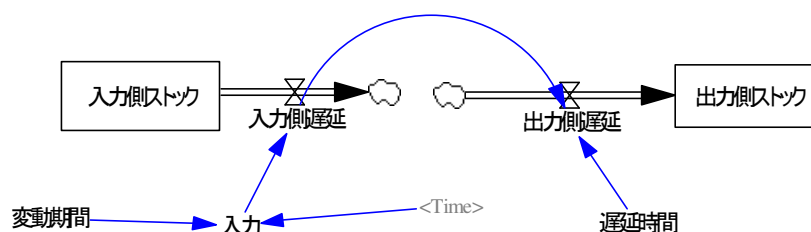


図 7-9：指数遅れと振動のモデル

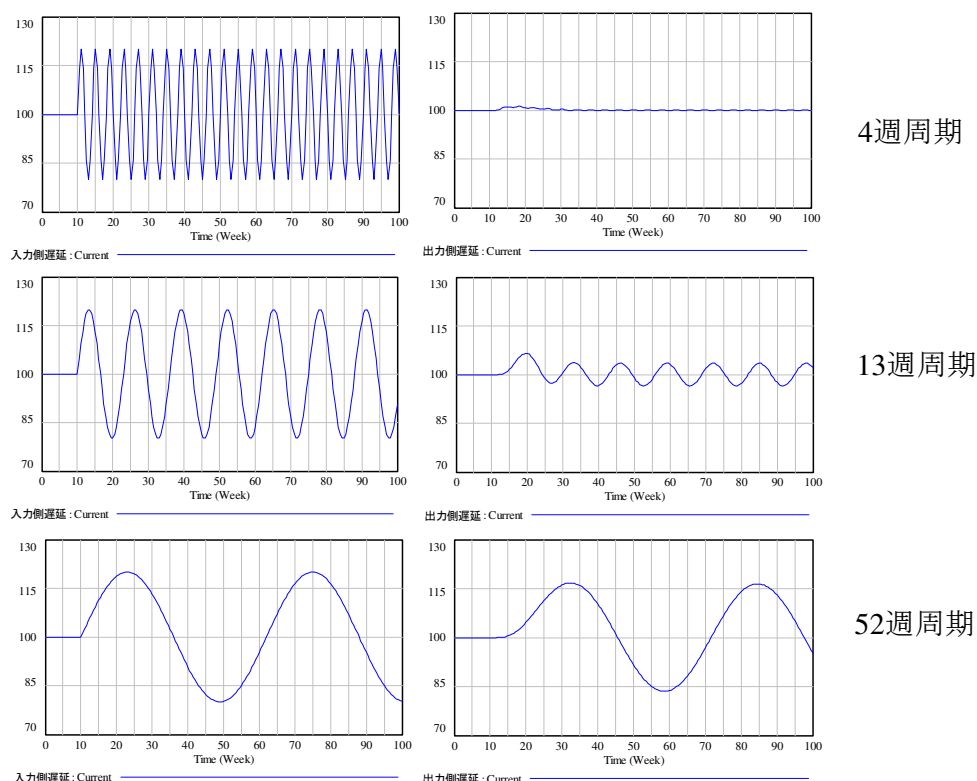


図 7-10：シミュレーション結果

```

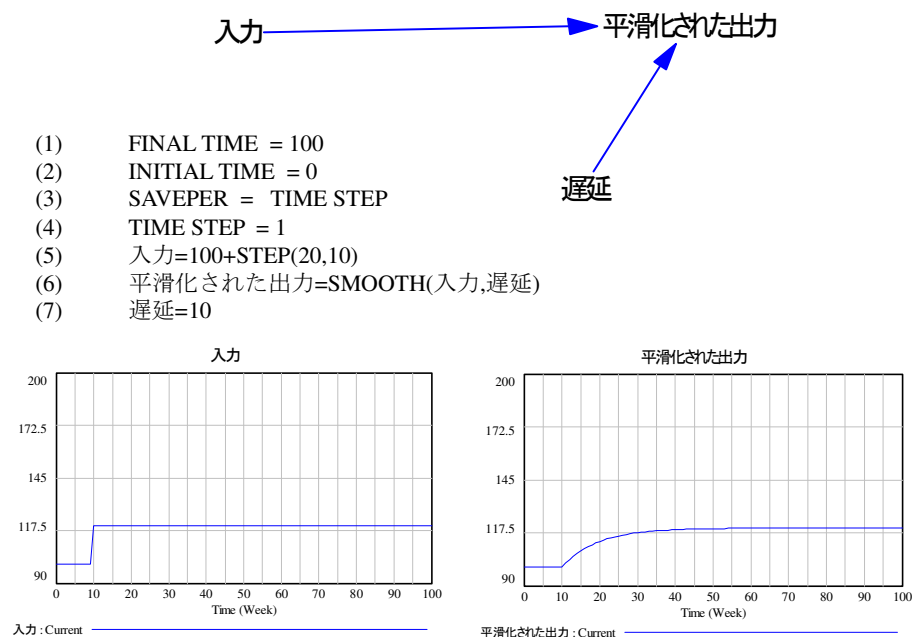
(01) FINAL TIME = 100
(02) INITIAL TIME = 0
(03) SAVEPER = TIME STEP
(04) TIME STEP = 0.5
(05) 入力=100+STEP(20,10)*SIN(2*3.14159*(Time-10)/変動期間)
(06) 入力側ストック= INTEG (-入力側遅延,10000)
(07) 入力側遅延=入力
(08) 出力側ストック= INTEG (出力側遅延,0)
(09) 出力側遅延=DELAY3(入力側遅延 , 遅延時間)
(10) 変動期間=4
(11) 遅延時間=10

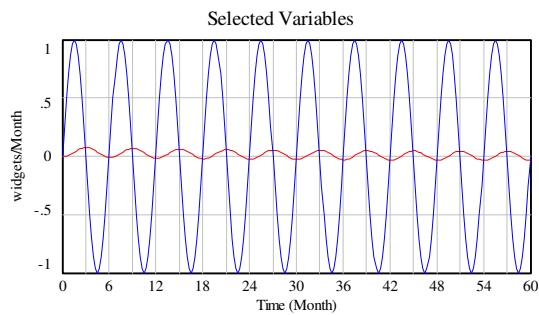
```

(4) 平滑化

変動が乱数的であるとしても、発注や受注、製造などを乱数的に行うことはできません。そうすると、経営資源の投入が振り回され、非効率になります。例えば、製造では、変化があったとしてもゆるやかに変化させ、なるべく一定のペースで製造したいし、実際にもそういったようにしかできないものです。例えば、製造量を一気に 2 倍に増やそうとしても、従業員を確保するのはそう一気にはできなく、人数は一気には増えないし、増やしたとしても、トレーニングなどが必要なので、急に製造を 2 倍にと言われてもすぐには対応できません。そこで、通常、平均あるいは平滑を使うのですが、単純平均だと、変化に必ずしもうまく対応できません。例え平均であっても、最近のものをより重要視すると、急激な変化によりうまく対応できます。このために、こういった場合、移動平均を使います。SD モデルでよく、ストック値をある期間で割るという処理を目にすることがあると思いますが、これは移動平均を行っている事と同じです。

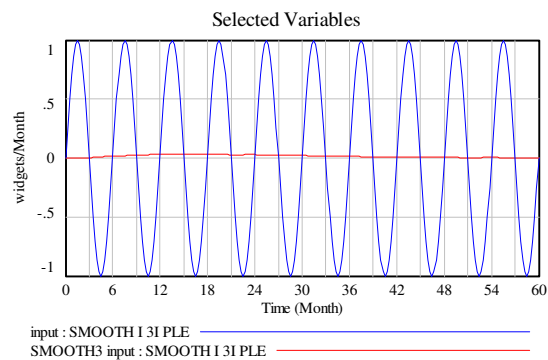
さらには、平滑化を使って指数的に平滑化した方がもっと精度がよくなります。Vensim では、SMOOTH、SMOOTH3、SMOOTH1、SMOOTH3I などの平滑化関数が用意されています。





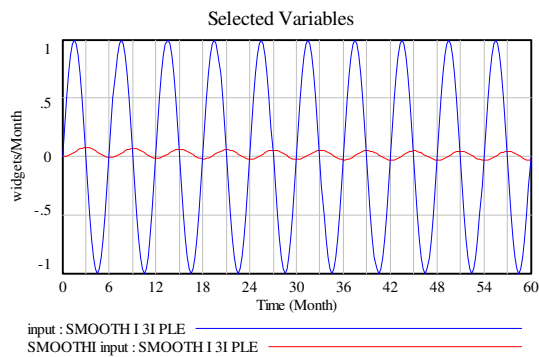
平滑化のいろんな関数

SMOOTH input
=SMOOTH(input,smoothing time)
一次の平滑化



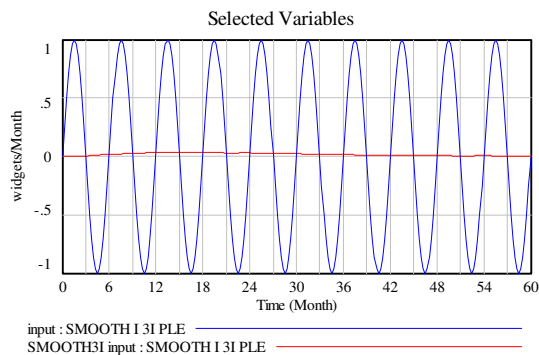
SMOOTH3 input=
SMOOTH3(input,smoothing time)
3次の平滑化

図 7-12 : 平滑化



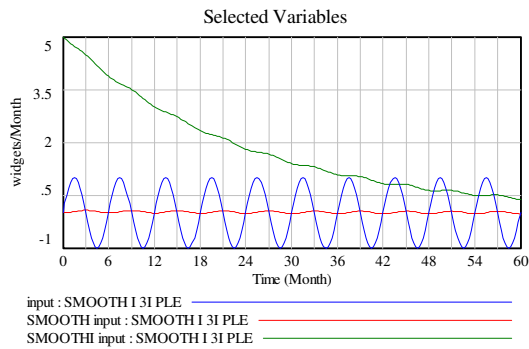
平滑化のいろんな関数

SMOOTH input =
SMOOTH(input, smoothing time, input)
一次の情報平滑化

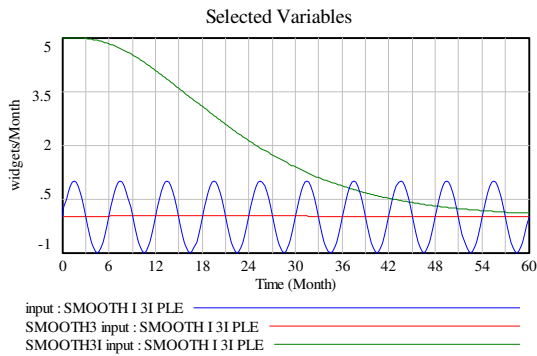


SMOOTH3 input=
SMOOTH3I(input, smoothing time ,
input)
3次の情報平滑化

図 7-13 : 情報付き平滑化

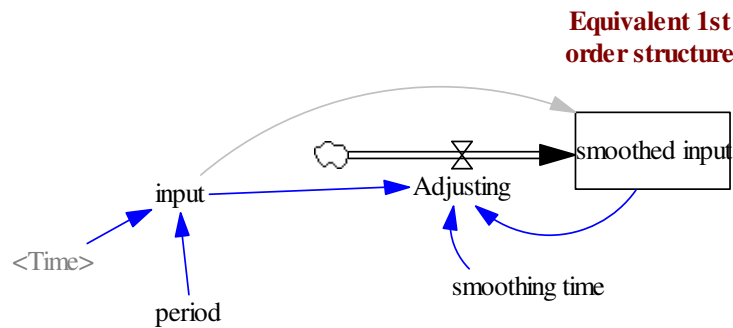


平滑発生までの初期値を指定できるので、ここでは5を指定している。従って、5から徐々に変動を伴いながら減少する。変動は1次で平滑化された振幅と同じ。



こちらは3次で平滑化している。変動がより滑らかになる。

図 7-14：情報付き平滑化



何をやっているかだが、1次の平滑化では、単に、ストックに値を累積し、それを調整時間で割っているだけ。

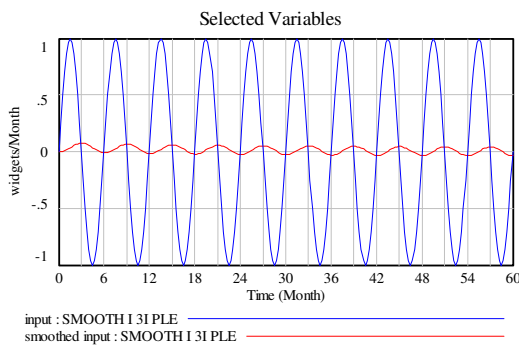


図 7-13：平滑化の原理

SMOOTH は一次の平滑化ですが、SMOOTH3 は3 次の平滑化で、高次になるほど、円滑化はより滑らかになります。

前に指数遅れという関数の原理について説明しましたが、平滑化でも、単に、ストックに値を累積し、それを調整時間で割っているだけなので、実は、遅延の原理で説明したことと同じようなことをやっていることになります。

指数遅れと同じように、平滑化でも SMOOTH では初期値は Vensim PLE が自動計算してくれますので、指定しません。ただ、やはり、DELAY3 などと同じく、モデルの作り方によっては、初期値同時指定の問題が発生し、実行時エラーになることがあります。遅延関数と同じく、情報付平滑化関数が用意されていますので、必要に応じてそちらを使って下さい。図 7-14 に示していますが、初期値がゼロではないものの場合、情報付き平滑化を使い、このように、変化させることができます。

(5) まとめ

遅れには、パイプライン型の単純に遅れるだけのものと、指数遅れの2種類があります。指数遅れは、急激な変化にすぐに追いつけないような変化で用います。

急激な変化を和らげるとか、傾向を知る方法として、平均や平滑化があります。単純な平均では、急激な変化にはうまく対応できないので、最近のものをより重要視する移動平均が用いられます。平滑化はこの移動平均に考え方が近いもので、変化を指数化したものです。使い方次第ですが、Smooth 関数を使った時点で、1 時間単位（日、週、月）の遅れが発生するので、指数遅れの Delay 関数を使ったのと同じような影響があると考えられます。

8. 意思決定

意思決定は、経営では情報と物流を結び付けるものです。2種類の情報があります。一つは、どのような情報を集め、どのように処理するかということに関するもので、この意思決定では、情報ネットワークを通じて物流ネットワークをコントロールします。もう一つは、どのような情報ネットワークのモデルにするかというもので、構築された意思決定モデルに基づき、情報システムによる日常的、定常的業務の自動意思決定を行います。

しかしながら、まだ多くの日常的、定常的な業務における意思決定は人間によって行われています。とは言っても、人間の意志決定をモデルにして検討してみることは重要です。

(1) 専門家と専門化

人間の意志決定をいろいろモデルにして分析することが行われてきて、その結果、人間の意志決定は、そう複雑なものでも、効果的なものでもないことが分かっています。人間の意志決定を比較的単純なモデルで表現できることが分かり、医療などでは、そういったモデル（エキスパート・システム）をサポートとして使うことで、人間の意志決定を改善しようとすることも行われていますし、新人の訓練用に、開発されたエキスパート・システムを使っている企業もあります。

Dawsによれば、人間の意志決定には、心理的な側面と数学的な側面での問題があるそうです。人間の行う意志決定では、心理学的に、特に注目していない要素は単純かつ画一的に見えて、それらが関係する際に、相互に相手をあまり変えない（相互作用がほとんどない）というように見えています。また、要素の変化が将来に大きく影響しない（定型的な一定の変化はするが、複雑な相互作用は及ばさない）というように見がちです。

それに対し、注目している要素は、過大評価してしまいがちです。つまり、ある要素が関連する他の要素に与える影響を、多め、あるいは少な目に見てしまいます。人間には、心理学的に、複数の要素の繋がった変化を取り扱えないという弱点があるそうです。

また、数学的に、直線変化と考えるがちで、指数変化とはまず考えません。効率性などの要素を取り扱う際ですら、直線的な増加として考えてしまい、指数的な増加はイメージされにくいそうです。トレーニングしない限り、人間は上記のようにしか考えられなく、これは専門家であっても、このことは変わらないそうです。（注 8-1）

とは言え、依然として専門家が多くの意志決定に携わっていることも事実です。

意志決定に関する研究がいろいろ行われ、線型型の、移動平均法による統計的な予想と、経験を積んだ専門家の直感的な総合的予測のどちらが優れているかが研究されてきました。Dawsの研究によれば、1954年のMeehlの調査では、20のそういった研究を調べてみて、18の研究は、統計的な意思決定の方が、人間の直感による意思決定よりも優れていると述べていて、2つの研究だけが、どちらも変わらないと結論付けているそうです。

10年後にJack Sawyerが同じようなことを45の研究について調べ、そのどれも、統計的な予測が一番正確で、専門家の直感的な予測の方が優れていたという研究は一つも無かったそうです。だからと言って、単純に、人間の意志決定を機械的な意思決定で置き換えられるわけではなく、また、線型的なモデルを使った判断は専門家や人々にはものすごく不評です。（注 8-1）

状況によっては、エキスパート・システムよりも、既存の知られているモデルの方が有効で、そういったものを活用する方がいいことも多くあります。例えば、まだ存在もしていない、新しい飛行機がどんなものを予想することは、エキスパート・システムも専門家も苦手です。しかし、流体や空気、空気中を移動する物体に関する物理的な法則はよく知られていることであり、そういった物理法則を使った定量モデルが存在します。それを使って、コンピューターの中で、飛行機的设计を行って、飛行したらどうなるかをチェッ

クしながら設計を進めていきます。そういった意味では、新しい飛行機が出現するはるか前から、飛行機の設計に定量モデルを使って設計し、強度実験や負荷実験などを行っていると言えます。

人間の意志決定には、状況的な情報やフィードバックが使えるという利点もあります。これから採用しようと考えている学生の能力を計測できなく、採用していいのかどうか判断できなくとも、先生が推薦してきていて、先生の学績はよく知られているので、その先生の推薦であればと信用して採用するといったことはよくあることです。

線形的なモデルによるエキスパート・システムを使った判断で、ホジキンス病に対する診断の正確度はわずか 18%だったそうです。もっとも同じ条件での有名な専門医師の診断の正確度は 0%でした。従って、現在のエキスパート・システムは、専門家の直感的な判断に対しての代替手段として、こういったものも必要という程度の位置づけで、専門家を代替するものとしては使われていません。さらには、そういった意思決定システム、あるいは意志支援システムの開発には専門家による支援が必要で、専門家の考え方や知識無しには開発はできません。このように、エキスパート・システムがどこまで専門家の仕事を代替できるかという観点でシステムが開発され、研究されてきたのですが、しかし、次第に、専門家と言ってもあてになるわけではないこと、複雑なモデルやシステムを開発する必要はなく、重要な項目に関する単純なモデルやシステムで十分だし、それが必要であることが分かってきました。今では、専門家のパフォーマンスを元に、エキスパート・システムの能力を図ることは無意味であり、それよりも、お互いにどう支援でき、両者でどう意思決定を効果的にできるかが問題にされるようになっていきます。

(2) 意思決定をモデル化する：Thurabond ダムの放水管理

ここでは、ダムの貯水量と放水量の管理という非常に簡単なモデルでの意思決定を考えてみます。モデルが単純なので、ストックとフローの関係が分かりやすく、グラフ化してみると一目瞭然です。また、モデルが単純なので、意思決定がしやすいし、意思決定した結果についても理解しやすいでしょう。しかし、実際の経営では、このような単純なモデルではなく、複数のストックやフローが関係してきます。

このモデルの対象地域である米国西部にある Rappanno Valley は、野菜栽培には非常に適した土地ではあるのですが、問題は、降水量が少ないことです。この地方出身の Thurabond 上院議員の肝いりで、米国連邦政府の開発資金による Rappanno Valley 灌漑施設整備プロジェクトが立ち上がりました。この灌漑施設整備プロジェクトの一環として、Callahali 川の Big Stormy George に Thurabond ダムを建設しました。ダムは建設後 40 年間、プロジェクトで整備された灌漑水路に水を供給し、地域の農業を支えてきました。



図 8-1：Thurabond ダム灌漑モデル

モデルは、図 8-1 に示したようなものです。貯水池への「流入」は、Thurabond ダムの Big Stormy 貯水池に流れ込む川が運ぶ水の量で、これは、このシステムではコントロールできない要素（外部変数）です。

「放出」は、「下流域への放水量」に流れますが、これは、灌漑水路に供給され、農業用に使われます。この「下流域への放水量」に供給された水は、野菜が吸収した量、蒸発した量、余って、下流に流れて行った量に分類できますが、ここではそれらをまとめて、「流

出」としておきます。野菜が吸収する量、蒸発する量、余って、下流に流れて行った量の合計（「放出」）は約 0.5 百万立方エーカーです。従って、これだけの量が灌漑に必要と考えて下さい。

目標として、「貯水池の水量」として、百万立方エーカーの水を保持するようにしたいのですが、「流入」が降水状況で変化すること、灌漑用に「放水」しなければならないことから、常にこの目標水量を貯水できるわけではありません。

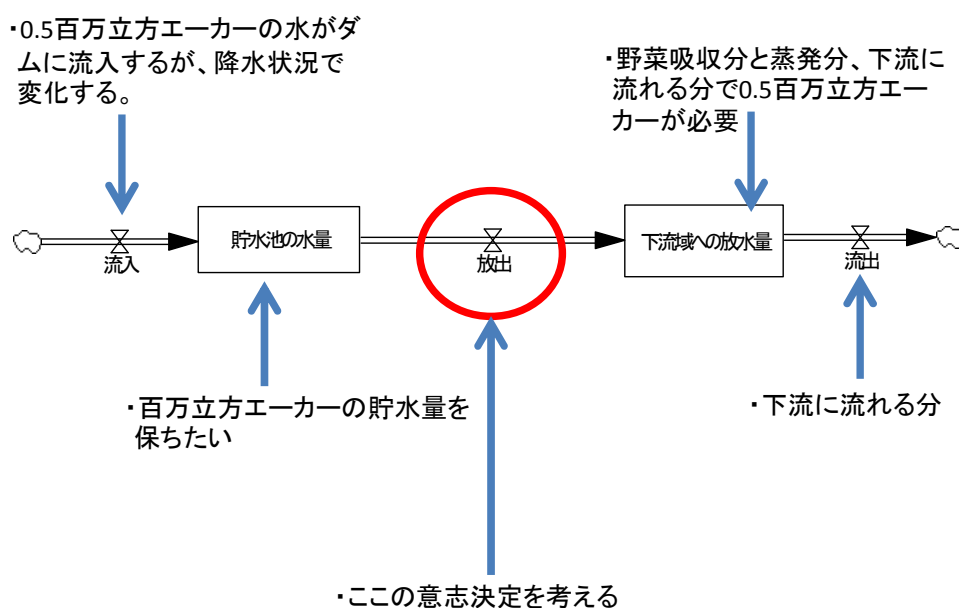


図 8-2 : Thurabond ダム灌漑モデル

1) ダムの目的：

このダムには、灌漑と洪水対策という 2 つの目的があります。

- ・灌漑：雨季は、野菜や穀物の種類によっては、生育時期ではなく、あまり水を必要としませんが、乾季には、そういった野菜や穀物の生育時期に当たり、灌漑を必要とします。
- ・洪水対策：雨季に、そのまま川の水を流すと洪水になる可能性のある地域があるので、雨季に水量調整を行い、流量を抑えることで洪水から守ります。

同じような構造が企業でも存在します。もし、コントロールを間違えると、部品や商品が過剰に溜まってしまったり、不足したりします。適切に、「貯水＝保管」し、「放出」しなければなりません。

2) 意志決定：

意思決定では、2 つのことを考えます。

- ・どのような要素を意思決定モデルに組み込まなければならないか？
- ・どのようにそれらの要素を組み込むか？

ダムの貯水量をあまり多くすると、上流域で雨が降り、突然水がどっと流れてきた時に危険になります。こうなると、緊急放水しなければならず、放水量によっては下流域に洪水を引き起こしますし、ダムも崩壊する危険にさらされます。一方、ダムの貯水量があまりにも少なすぎると、乾季の時の灌漑需要に応えられなく、何のためのダムなのかと非

難を浴びることになります。要約すると、「貯水池の水量」と「下流域への放水量」をある一定の範囲内に収めることがここでの意思決定になります。従って、意思決定のモデルでは：

- ・要素の平均量を決める
 - ・要素の量（生産量）を決める
- ことになります。

(3) 荷重平均法による意思決定モデル

荷重平均法を使って変数をコントロールするというやり方は、直接的で直観的にも受け入れられやすいのではないかと思います。これは：

- 1) ある目標値を維持する。もし上回る、あるいは下回った場合は、ある率で標準値に戻していく。
- 2) 同じく、この調整をある期間内に平滑化して行う。（これを平均化という。）スムーズに平均化することで、突然の変動を避けることができる。
- 3) 変化の総合計（あるいは最終値）が目標になる。
- 4) そうなるように、ファクターに重み付け（係数）を加え、それで変化を調整する。というものです。この方法を、対象としている意志決定モデルに採用することになります。

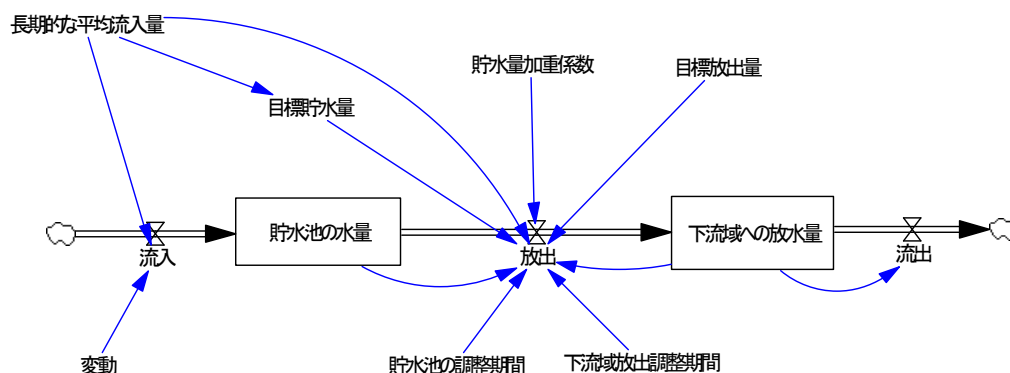


図 8-3：意思決定モデル

図 8-1 のコンセプチュアル・モデルのパラメーター「放出」に、6 つの係数を付け加えました。「長期的な平均流入量」は、ダム上流部からダムに流れ込む川の水量の長期的な平均水量で、0.5 百万立方エーカーと分かっています。この量を「放出」すれば、ダムの目的を十分果たすことができます。

地域の農業は、50%灌漑用水路に、残り 50%は川に流れる水に頼っています。すでに述べたように、野菜吸収量、蒸発量、下流域への流出量で 0.5 百万立方エーカーが必要です。従って、「目的放出量」は、先の 2 倍に設定して、1.0 百万立方エーカーになります。

「貯水池の調整期間」と「下流域放出調整期間」は調整期間で、「貯水池の調整期間」は半年、「下流域放出調整期間」は約 1 週間とします。

「貯水量加重係数」は必要な「下流域への放水量」を確保するための調整係数です。

長期的には、「放出」は「長期的な平均流入量」と同じでなければなりません。そうでないとバランスしなくなり、おかしくなります。そこで：

$$\text{「放出」} = \text{「長期的な平均流入量」} \quad (8.0)$$

となります。

他方、目標貯水量になるように貯水池の水量をコントロールしなければならないので：

$$\text{「放出」} = \text{「貯水池の水量」} - \text{「目標貯水量」} \quad (8.1)$$

となります。

「目標貯水量」よりもダムの貯水量が多ければ、多めに放出し、少なければ、少な目に放出します。ただし、この調整は、調整期間内で行うので、先の式、(8.1)は：

$$\text{「放出」} = \frac{\text{「貯水池の水量」} - \text{「目標貯水量」}}{\text{「下流域放出調整期間」}} \quad (8.2)$$

となります。

$$\text{「貯水池の調整係数」} = \text{「貯水量調整係数」} = \frac{1}{\text{「貯水池の調整期間」}}$$

従って、(8.2)式は：

$$\text{「放出」} = \text{「貯水量調整係数」} \times (\text{「貯水池の水量」} - \text{「目標貯水量」})$$

「貯水量加重係数」を使って、(8.2)式は：

$$\text{「貯水量加重係数」} \times \frac{\text{「貯水池の水量」} - \text{「目標貯水量」}}{\text{「貯水池の調整期間」}} \quad (8.3)$$

また、灌漑のために、「下流域への放水量」を確保する必要があるので：

$$(1 - \text{「貯水量加重係数」}) \times \frac{\text{「目的放出量」} - \text{「下流域への放水量」}}{\text{「下流域放出調整期間」}} \quad (8.4)$$

「放出」は、これまで述べてきた3つの式(8.0)、(8.3)、(8.4)を合計したものになり：

$$\begin{aligned} \text{「放出」} &= \text{「長期的な平均流入量」} \\ &+ \text{「貯水量加重係数」} \times \frac{\text{「貯水池の水量」} - \text{「目標貯水量」}}{\text{「貯水池の調整期間」}} \\ &+ (1 - \text{「貯水量加重係数」}) \times \frac{\text{「目的放出量」} - \text{「下流域への放水量」}}{\text{「下流域放出調整期間」}} \end{aligned} \quad (8.5)$$

となります。

(8.0) は長期的には「放出」は「長期的な平均流入量」でなければならないという式です。(8.3) は、ダムの水量を標準量に保つようにする式で、(8.4) は、灌漑に必要な量をダムから放出するという式です。

ここまで決めれば、あとは、「貯水量加重係数」にふさわしい値を設定するだけです。ふさわしい値を設定するために、0、0.5、1.0 の3つの場合についてシミュレーションで結果を確かめてみます。

「貯水量加重係数」を0にすると、灌漑に必要な量をダムから放出することだけを重要視することになります。「貯水量加重係数」を0.5にすると、ダムの水量を標準量に保つことに50%注意しながら、50%分は灌漑に必要な量をダムから放出することになり、「貯水量加重係数」を1.0にすると、ダムの水量を標準量に保つことだけに専念することになります。

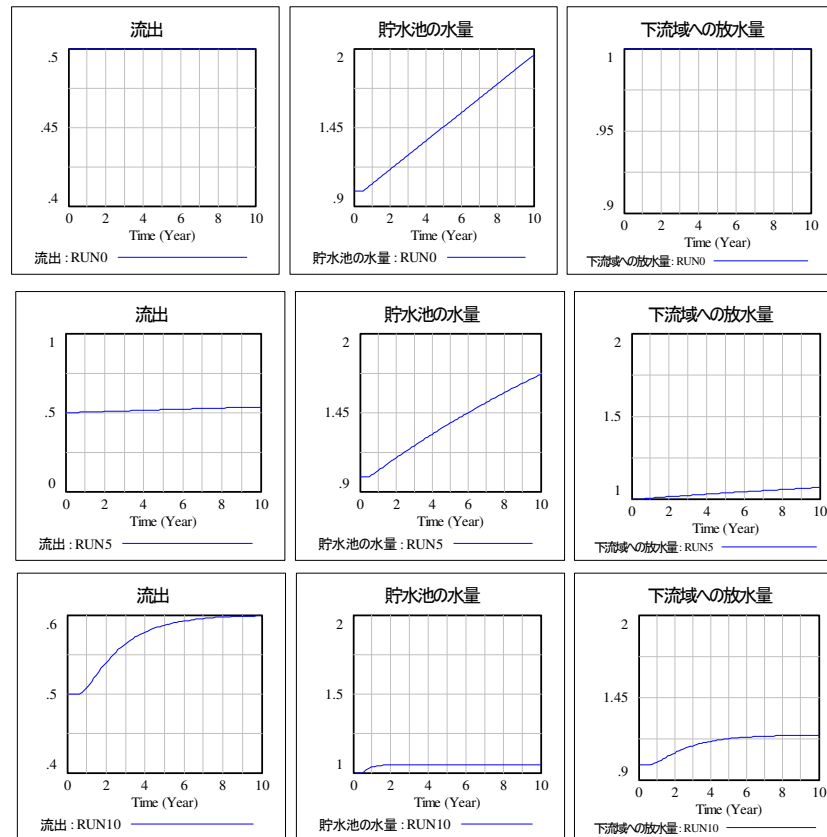
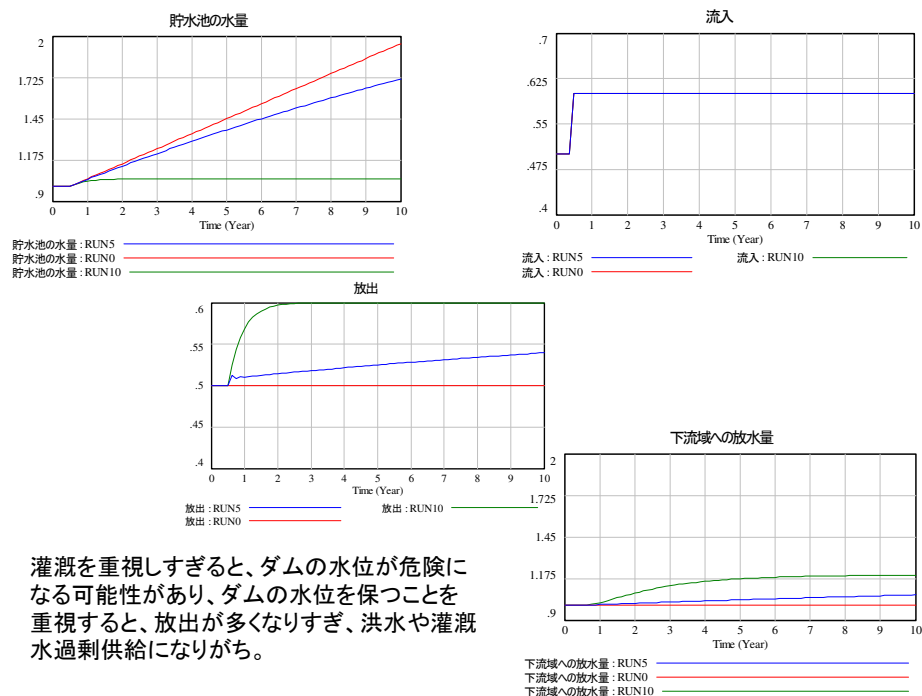


図 8-4：シミュレーション結果



灌漑を重視しすぎると、ダムの水位が危険になる可能性があり、ダムの水位を保つことを重視すると、放出が多くなりすぎ、洪水や灌漑水過剰供給になりがち。

図 8-5：シミュレーション結果

(4) 目標が変動する場合

先の議論では、「長期的な平均流入量」が分かっている、固定だという前提で行ってききました。これは一種の目標ですが、しかし、もし、この値を目標に使えない場合はどうするかを考えます。一つの解決法として、ダムに流れ込む「流入」の状態から、「短期的な平均流入量」を求め、これを目標にして、ダムの水位と灌漑を調整するやり方があるので、それを検討します。ここでは、「流入」を「平均流入時間」で平滑し、「短期的な平均流入量」として設定し、これを調整目標値に使います。平滑化した値を目標に使うというものです。

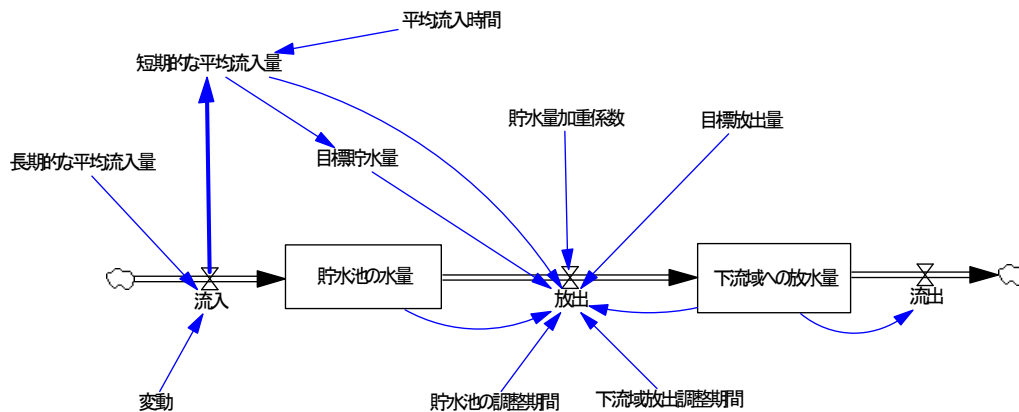


図 8-6：短期目標をセットした意思決定モデル

図 8-6 に長期目標から短期目標に変えたモデルを、図 1-7 にそのシミュレーション結果を示していますが、目標が変化するので、計数を 1.0 にして、ダムの貯水量を標準値に保つことを優先する意志決定では、「放出」が一時的（約 1 年間）に落ち込む現象が見られます。

このモデルの等式は以下の通りです。

(01)	FINAL TIME = 10
(02)	INITIAL TIME = 0
(03)	SAVEPER = TIME STEP
(04)	TIME STEP = 0.125
(05)	下流域への放水量= INTEG (放出-流出,目標放出量)
(06)	下流域放出調整期間=0.05
(07)	変動=STEP(0.1,0.5)
(08)	放出=長期的な平均流入量+貯水量加重係数*(貯水池の水量-目標貯水量)/貯水池の調整期間+(1-貯水量加重係数)*(目標放出量-下流域への放水量)/下流域放出調整期間
(09)	流入=長期的な平均流入量+変動
(10)	流出=0.5*下流域への放水量
(11)	目標放出量=1
(12)	目標貯水量=2*長期的な平均流入量
(13)	貯水池の水量= INTEG (流入-放出,目標貯水量)
(14)	貯水池の調整期間=0.5
(15)	貯水量加重係数=1
(16)	長期的な平均流入量=0.5

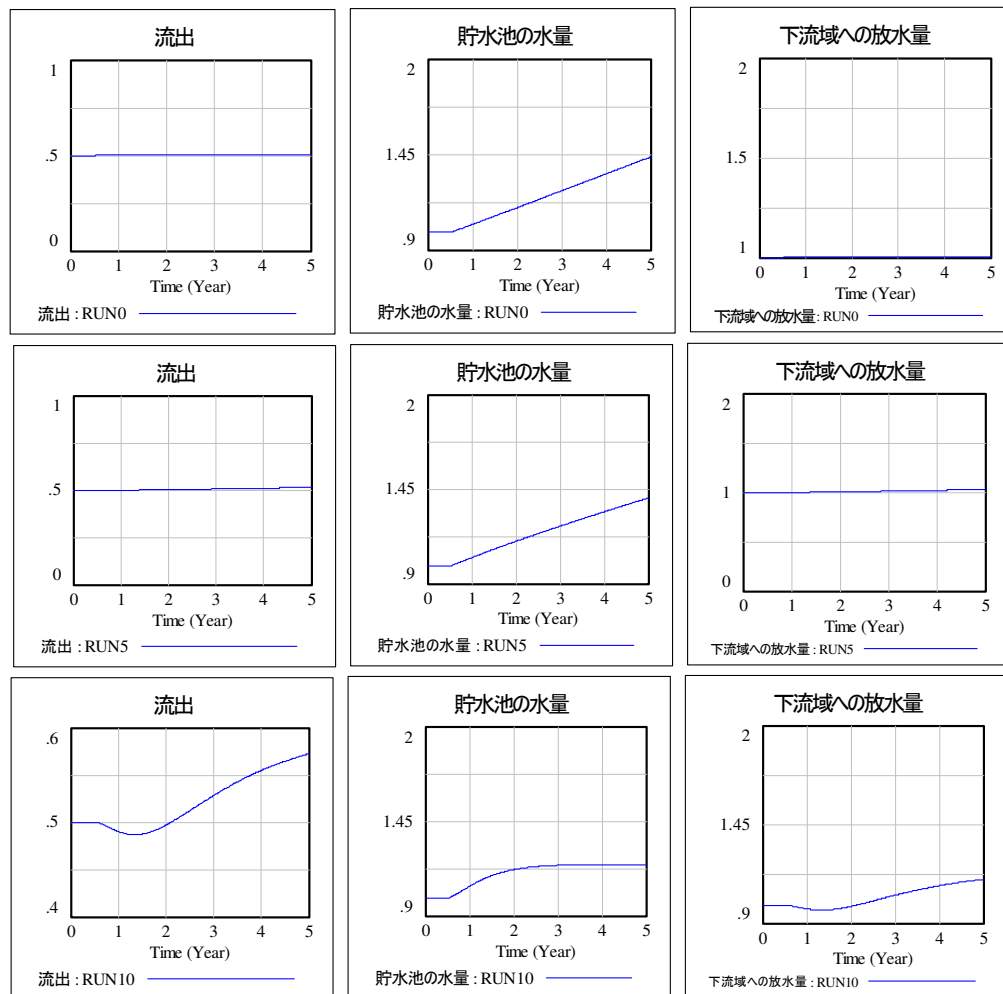


図 8-7：シミュレーション結果

(5) 勾配係数による意志決定

先のような単純な重み付けのやり方とは別に、勾配係数を重み付け係数に使うという意志決定の方法があります。これは、勾配が強くなればなるほど、重み付け係数が大きくなるというしくみです。

図 8-8 にそのモデルを、図 8-9 にその原理を、図 8-10 にそのシミュレーション結果を示しています。この方法では、調整期間で平滑化するわけではないので、平滑化期間を指定するパラメーターが無くなり、単純に、貯水池の水量をコントロールする勾配係数と、灌漑のために下流域に方室する量をコントロールする勾配係数のみが追加されます。

こうしてみると、こういったシステムでの意思決定の結果もそう悪くないことが分かります。何よりも、意思決定の結果をシミュレーション結果で確かめてみるができることが強みです。

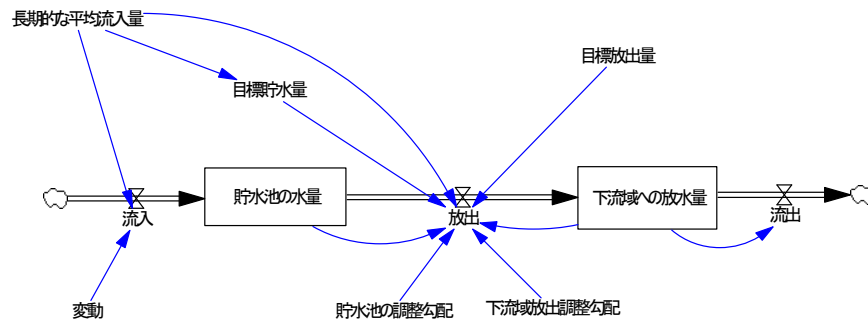
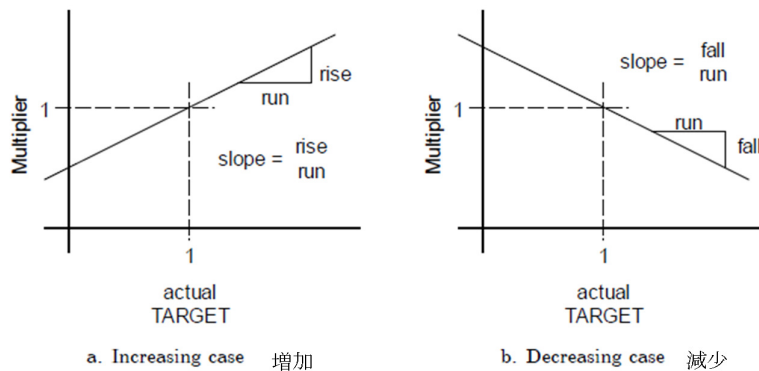


図 8-8：勾配係数による意思決定モデル



$$\text{重付係数} = \text{勾配} \times \frac{\text{実際の値} - \text{目標値}}{\text{目標値}} + (1 - \text{勾配}) \quad \text{重付係数} = 1 + \text{勾配} - \text{勾配} \times \frac{\text{実際の値} - \text{目標値}}{\text{目標値}}$$

図 8-9：勾配係数の原理

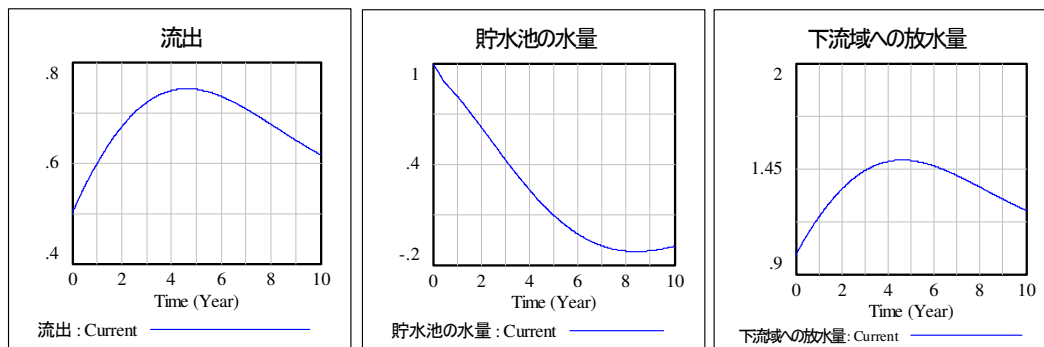


図 8-10：勾配係数による意思決定モデルのシミュレーション結果

このモデルの等式は以下の通りです。

- | | |
|------|---------------------------------|
| (01) | FINAL TIME = 10 |
| (02) | INITIAL TIME = 0 |
| (03) | SAVEPER = TIME STEP |
| (04) | TIME STEP = 0.015625 |
| (05) | 下流域への放水量 = INTEG (放出-流出, 目標放出量) |
| (06) | 下流域放出調整勾配 = 0.05 |

(07)	変動=STEP(0.1,0.5)
(08)	放出=長期的な平均流入量*(貯水池の調整勾配*(貯水池の水量/目標貯水量)+(1-下流域放出調整勾配)*(1+下流域放出調整勾配-下流域放出調整勾配)*(下流域への放水量/目標放出量))
(09)	流入=長期的な平均流入量+変動
(10)	流出=0.5*下流域への放水量
(11)	目標放出量=1
(12)	目標貯水量=2*長期的な平均流入量
(13)	貯水池の水量= INTEG (流入-放出,目標貯水量)
(14)	貯水池の調整勾配=0.5
(15)	長期的な平均流入量=0.5

注 8-1)

Dawes, R. M. 1979. The Robust Beauty of Improper Linear Models in Decision Making. American Psychologist 34, 571-582.

Dawes, R. M. 1988. Rational Choice in an Uncertain World. Harcourt Brace Jovanovich, San Diego.

9. 非線型性

線型と考える、あるいは線型と考え処理していることは実は多くあります。預金額を 2 倍にすれば、利子が 2 倍になって戻ってきますし、1 割仕事をする時間を増やせば、1 割余分に仕事をこなせます。線型であれば、処理しやすいし、コンピューターを使わなくとも解けることもあるので便利です。従って、世の中の多くのことは、線型のモデルとして開発され、線型モデルで条件を与えて、解いています。

しかし、ビジネスの世界では非線型のことが多くあります。特に極端にした場合、非線型性が顕著になってきます。1 割仕事をする時間を増やせば、1 割余分に仕事をこなせるかも知れませんが、仕事時間を 2 倍にしたら、くたびれて仕事の効率が下がり、2 倍の仕事はできないでしょう。同様に、製造能力には限界があり、製品の販売量にも上限があり、顧客の需要にも限界があるのが普通です。税金や預金のシステムでも、累進課税になっているし、預金高によって利子が違って来るようなシステムになっています。問題は、非線型であると、手計算できないことが多いことです。

(1) 非線型的反応

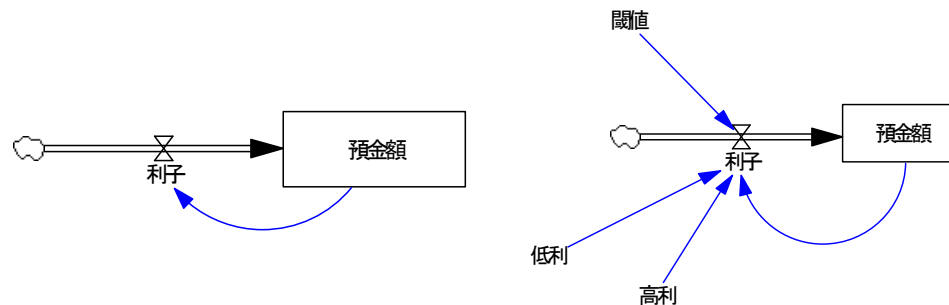


図 9-1：線型モデルと非線型モデル

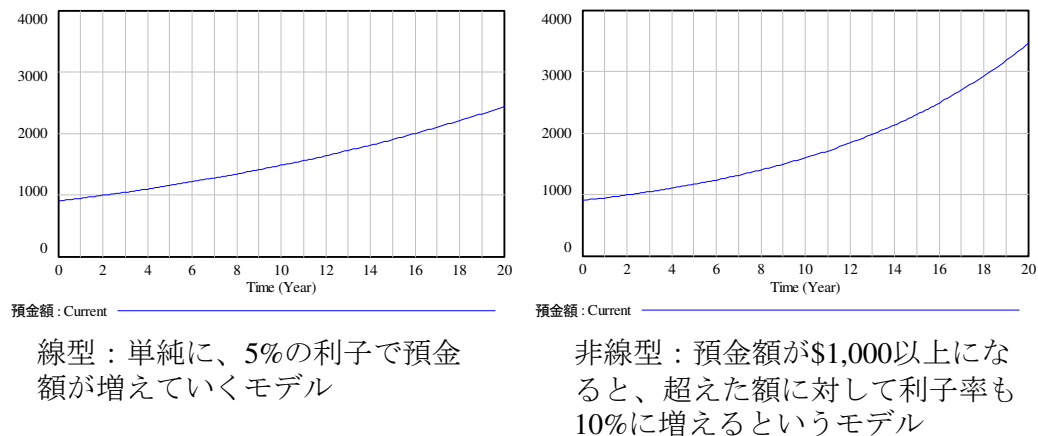


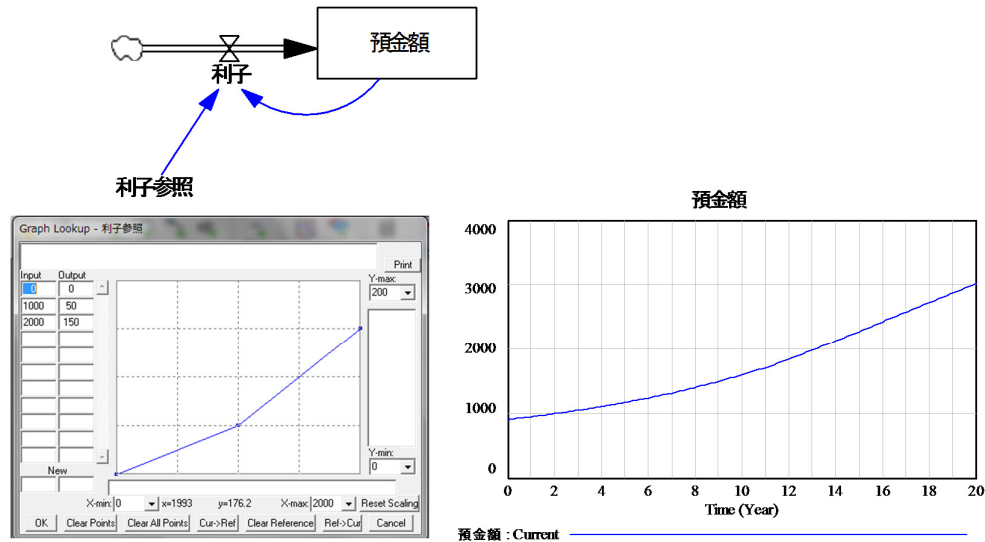
図 9-2：線型モデルと非線型モデルのシミュレーション結果

ここでは、2つの簡単なモデルを考えます。図 9-1 の左は、単純に、5%の利子で預金額が増えていくモデルで、それに対し、右は、\$1,000 以下だと利子は 5%だが、預金額が\$1,000 以上になると、超えた額に対して利子率も 10%に増えるというモデルです。シミュレーション結果を図 9-2 に示していますが、\$1,000 以降の預金額の伸びが違い、左よりも右は勾配

が大きくなっています。

ここでは非線型性を表現するために IF THEN ELSE を使っていて、定義は下のようになっています。

利子=IF THEN ELSE(預金額<閾値, 低利*預金額, 低利*閾値+高利*(預金額-閾値))



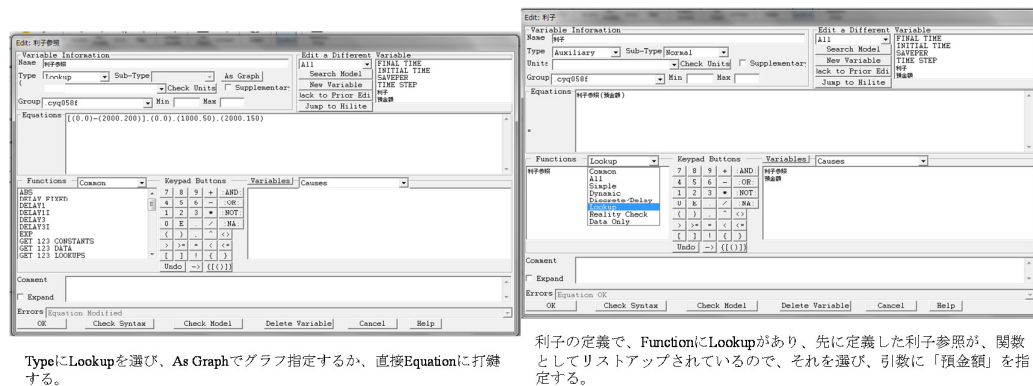
LOOKUPを使って。

利子=利子参照(預金額)

利子参照([(0,0)-(2000,200)],(0,0),(1000,50),(2000,150))

Vensimでは、このように利子参照を関数にできる。

図 9-3 : LOOKUP を使った定義



TypeにLookupを選び、As Graphでグラフ指定するか、直接Equationに打鍵する。

利子の定義で、FunctionにLookupがあり、先に定義した利子参照が、関数としてリストアップされているので、それを選び、引数に「預金額」を指定する。

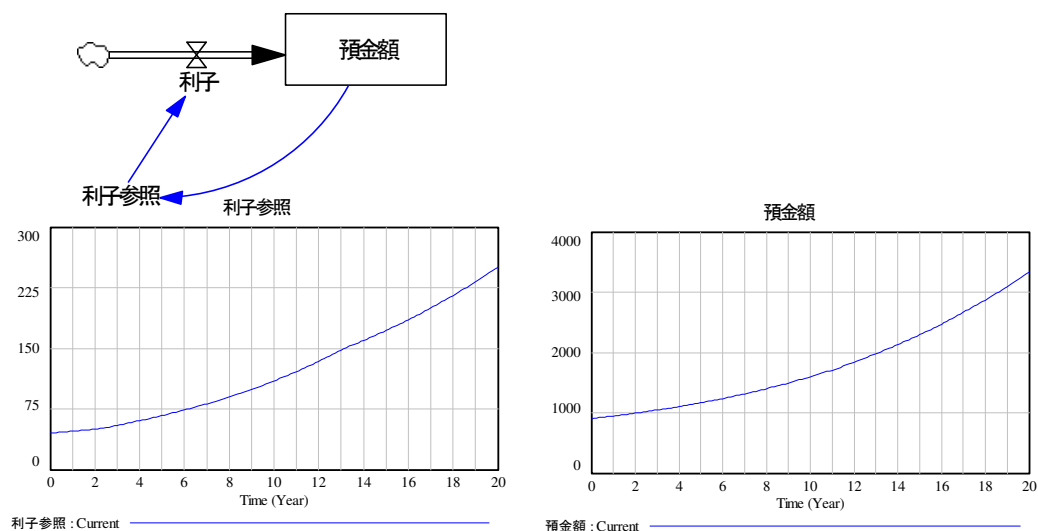
図 9-4 : LOOKUP を使った定義

非線型性は、IF THEN ELSE を使うのではなく LOOKUP を使って、表から参照させるというやり方もあります。等式定義で、Type に Lookup を選び、As Graph でグラフ指定するか、直接 Equation に打鍵します。すると、Vensim PLE は、この「利子参照」というパラメーターを一種の関数にします。「利子」では、この「利子参照」という一種の関数を使って計算できます。

「利子」の定義の際に、定義画面の左にある Functions の中に Lookup があり、それを開

くと、先に定義した「利子参照」が、関数としてリストアップされていますので、それを選び、引数に「預金額」を指定します。

もう一つのやり方は、with Lookup を使うやり方です。この場合は、「利子」ではなく、「利子参照」の方で引数に「預金額」を指定します。



先の代わりに、with LOOKUPを使ってもいい。

利子参照 = WITH LOOKUP (預金額,([(0,0)-(4000,300)],(0,0),(1000,50),(2000,150),(4000,300)))

図 9-5 : with Lookup を使った定義

IF THEN ELSE 文は直接的で分かりやすいし、単純ですが、ネスト状態のものをこれで記述すると面倒になります。そういった場合は、LOOKUP や with LOOKUP を使った方が、記述が簡単になります。

(2) 経営資源の有限性

経営資源が有限で、経営資源が少なくなって、経営活動が減退することはよくある話です。図 9-6 の左のモデルの例では、営業がいくら販売しようとも、製品在庫が無くなれば、それ以上は販売できないのですが、しかし、そんなことを考えないと、どんどん商品を売っていきけるというように考えてしまい、在庫がマイナスになっていることに気が付かないことがよくあります。線型では架空のこと（在庫がマイナスで受注残とみなすといったこと）もありうるのですが、非線型では架空のことはあり得ないので、これは成立しません。在庫はゼロより少なくはなりません。

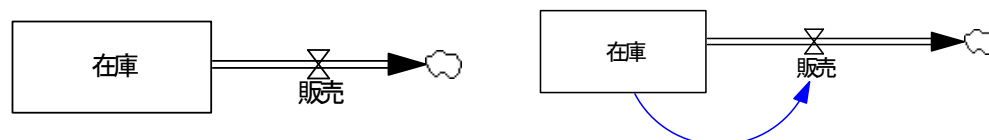


図 9-6 : 在庫の販売の線型、非線型モデル

図 9-7 の上に、図 9-6 の左のモデルでのシミュレーション結果を、下に、右のモデルのシミュレーション結果を示していますが、線型モデルのままにしておくと、現実的ではないこのようなことが起きますので、注意が必要です。SD モデルでは、現実的なものにするために、非線型性を組み入れることが重要です。

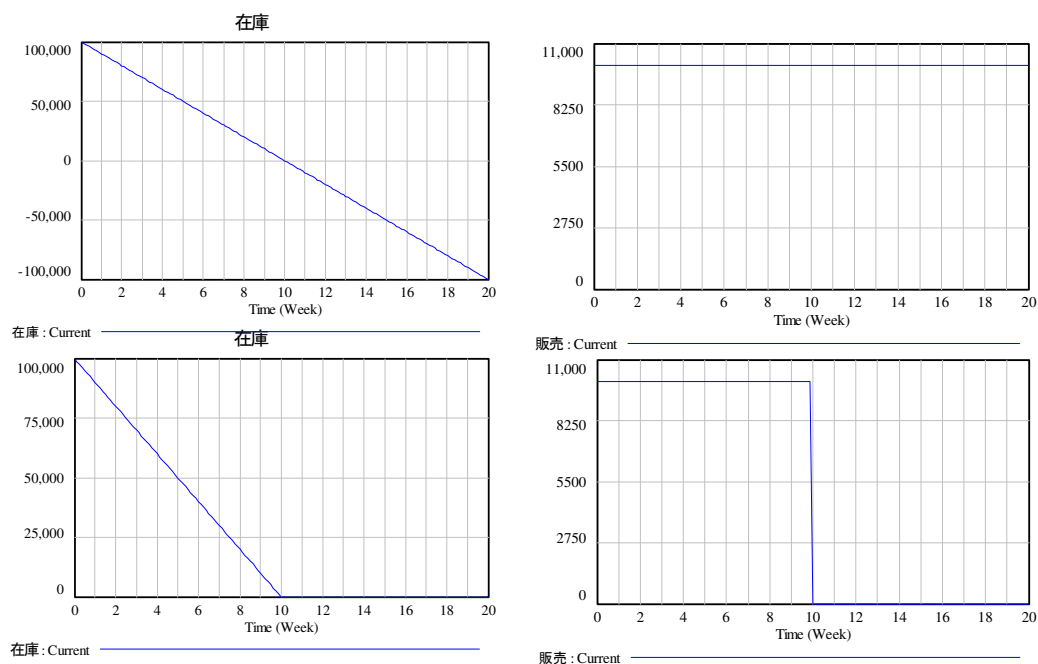


図 9-7：在庫の販売の線型、非線型モデルのシミュレーション結果

社会科学、例えば経営とか社会開発では、ある仮説を実際に実験で確かめることがなかなかできません。もし失敗したら実際に被害が発生し、会社が倒産してしまうとか、社会開発が失敗して、国家が負債を抱える、あるいは開発の失敗で政治不安が起きるなど、リスクがあまりにも大きすぎるからです。そこで、何かを数理モデルにして、コンピューター・シミュレーションで仮想実験し、その仮説を確かめるというやり方が行われます。ただ、多くの定量モデルは線型モデルであり、非線型を取り扱える定量モデルはそう多くありません。SD は、非線型を取り扱うものとして、その中でも一番有名で多く使われているものです。

10. 初期値設定

モデルを作成する際に、ストックに初期値を設定しなければならないのですが、その、初期設定に関する事項を議論します。

(1) 等式定義における初期値設定

多くのモデルは均衡として考えられています。つまり、変数（係数、初期値）の値は通常は変化しないように定義します。しかし、方針の変化がどれだけの影響を及ぼすかを見たいので、モデルを作って、ショック（変動）を与え、その影響を計算しようとしています。そういった際の初期値の設定が問題です。

ここでは、研修と雇用の関係のモデルを取り上げています。社員を採用して一人前になるまでに平均3ヶ月の研修が必要です。社員は、研修後、熟練工として平均3年間（36月）勤務します。社員が退職するので、その穴埋分を新規採用します。この新規採用者の採用数は、社員の直近6ヶ月間（26週）での退職者数を指数平滑して決めます。

現在、研修を終えている社員数は1,000人、現在研修中の社員は250人であるとしています。さて、この条件でシミュレーションしてみると妙なことが起きます。この会社は、継続して操業しているので、多少の変動はあっても、研修中の社員の数や研修が終了した社員の数が大幅にぶれるとおかしいはずですが。

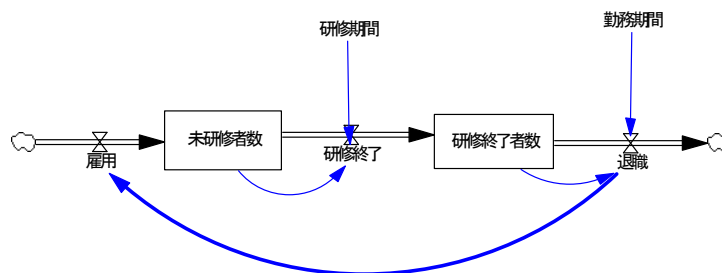


図 10-1：研修と新規採用のモデル

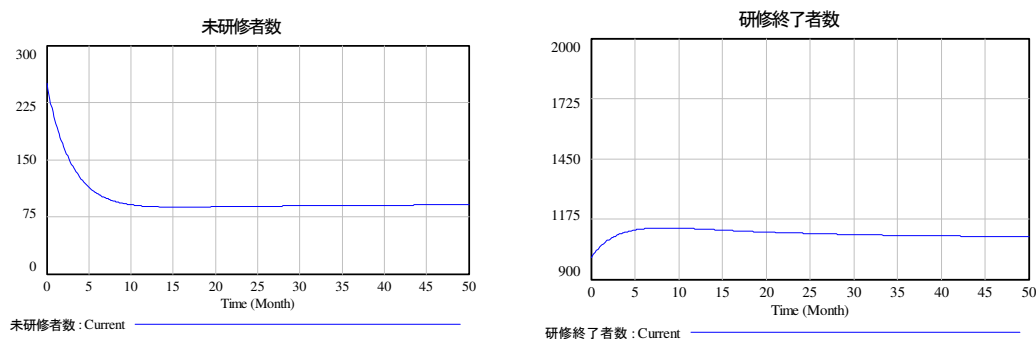


図 10-2：研修と新規採用のモデルのシミュレーション結果

つまり、それぞれのストックの入出力のフローは一定で、かつ、同じ値でないとおかしいはずですが。従って、このモデルでは、「雇用」と「退職者」の数が同じ（均衡している）でないとおかしいことになります。つまり、

$$\frac{\text{未研修者}}{\text{研修期間}} = \frac{\text{研修終了者}}{\text{雇用期間}}$$

とならないといけないはずで。とすれば、
 未研修者 = $1,000 \times (3/36) = 83.3$ (人)
 となるはずで。しかし未研修者の初期値を 250 人と設定しています。これが、図 10-2 のシミュレーション結果で、最初に未研修者がどんと落ち込む理由になっています。初期値は 83.3 (人) にしなければなりません。そう修正して、再度シミュレーションしてみます。今度は、こんな大幅な落ち込み現象は見られなくなります。

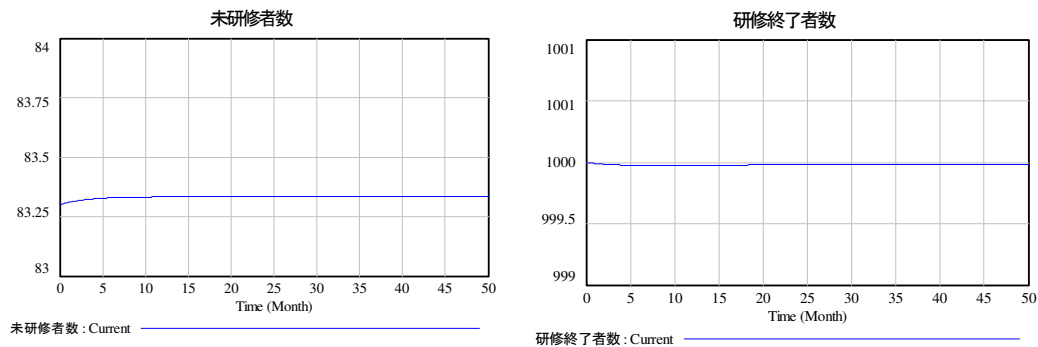


図 10-2：初期値修正後の研修と新規採用のモデルのシミュレーション結果

もっとも、私も、あまり初期値に注意を払ってモデルを作っていないので、こういったエラーはよくやります。初期値設定が不適切でも、モデルが均衡であれば、振動収束しますが、増強であれば、初期値のせいで、振動が拡大することになってしまいます。いずれにしろ、不適切な初期値設定のせいで、本来であれば起きるはずがない変動をシステムに与えているので、今後、気を付けなくてははいけません。

(2) 自動同時初期値設定

自動同時初期値設定が起きると、SD のソフトはこれを解決してくれなく、エラーになってしまいます。これは、フィードバック・ループにストックが介入しない、変数だけでモデルを作り、そこで SMOOTH 関数、あるいは指数遅れの DELAY 関数を使うと、この現象が発生します。

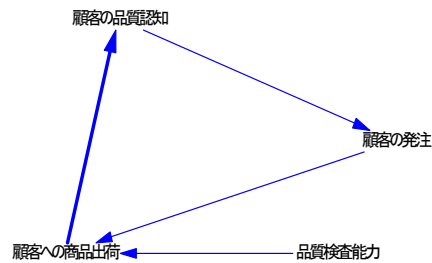
図 10-3 のモデルは、「品質検査能力」によって「顧客への商品出荷」の数が決まり、その数の変動を「顧客の品質認知」が認知し、発注した数量がきちんと納品されないといったクレームの状態を勘案し、あのメーカーはいつも注文した数量の 9 割しか納品してこないから 10%カットした発注量にするといった変更を行い発注するというものです。ここでは、「品質検査能力」を月間 10,000 個に設定しています。つまり、顧客の発注数が月間 10,000 個までであれば納期通りに納入できますが、それを超えると、超えた分は、能力の比で納品数が減っていきます。

このメーカーは、顧客から受けた発注に対し、この品質検査能力で検査し、その率で顧客に出荷しています。顧客は、メーカーから納品された商品を SMOOTH 関数で 6 ヶ月で平滑化し、出荷状況も含め品質を判断し、発注を、10,000 個に対してどのくらいの割合で増減するかを率で決めています。

ここで、SMOOTH 関数を使うと、SMOOTH 関数は遅れ関数と同じなので、遅れまでの間の初期値を要求し、もし、SMOOTH 関数がストックに対するフローで使われるのであれば問題がないのですが、このモデルでは変数で使われているので、ソフトでは自動的に初期値を設定してくれなく、実行時エラーになります。問題は、ストックがないとか、変なメッセージで、原因がエラー・メッセージだけでは特定できないことです。

このフィードバック・ループにもしストックが存在すれば、ストックには初期値を設定するので、その初期値を使って計算してくれ、このことは起きません。しかし、このフィ

ードバック・ループではそれが不可能になっています。



このフィードバック・ループにはストックが存在しない。通常、ストック値を計算する際に、初期値を計算してくれるので、そこでの計算が初期値として設定されるが、このフィードバック・ループではそれが不可能。

- (5)品質検査能力= 10000
- (6)顧客の品質認知=**SMOOTH**(顧客への商品出荷,6) ←
- (7)顧客の発注=10000*顧客の品質認知
- (8)顧客への商品出荷=品質検査能力/顧客の発注

このせいでこのモデルは実行時にエラーになる
これを防ぐには、
SMOOTH(顧客への商品出荷,6,1)にする。

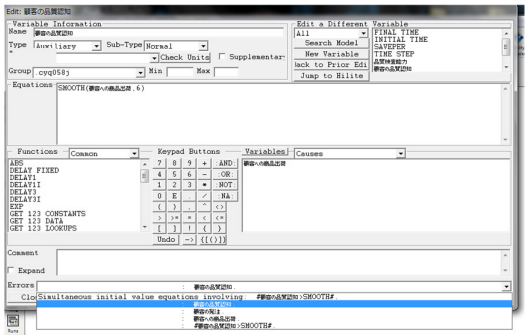
図 10-3：自動同時初期値設定のモデル

- (5)品質検査能力= 10000
- (6)顧客の品質認知=**SMOOTH**(顧客への商品出荷,6)
- (7)顧客の発注=10000*顧客の品質認知
- (8)顧客への商品出荷=品質検査能力/顧客の発注

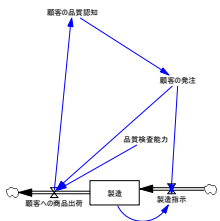
の式(6)のせいでこのモデルは実行時にエラーになります。これを防ぐには、フィードバック・ループのどこかで初期値を与えればいいのですから、(6)式を、初期値設定平滑関数である SMOOTHI を使い、

SMOOTHI(顧客への商品出荷,6,1)

とすれば解決します。



こんなエラーになる。「顧客の品質認知」に入力する前に、何か遅れ期間中の情報を与えればいいのだが、変数だけだと難しい。



- (05)品質検査能力=10000
- (06)製造= INTEG (製造指示-顧客への商品出荷,10000)
- (07)製造指示=IF THEN ELSE(製造>10000,0,顧客の発注/10000)
- (08)顧客の品質認知=SMOOTH(顧客への商品出荷,6)
- (09)顧客の発注=**DELAY FIXED**(顧客の発注/品質検査能力, 0.2, 10000)
- (10)顧客への商品出荷=顧客の発注/品質検査能力のように、DELAY FIXED を挟んで、初期値を与えてもこの問題を解決できる。

図 10-4：自動同時初期値設定の回避

あるいは、図 10-4 のように、DELAY FIXED を挟んで、初期値を与えてもこの問題を解決できます。

(05)品質検査能力=10000
(06)製造= INTEG (製造指示-顧客への商品出荷,10000)
(07)製造指示=IF THEN ELSE(製造>10000,0,顧客の発注/10000)
(08)顧客の品質認知=SMOOTH(顧客への商品出荷,6)
(09)顧客の発注=DELAY FIXED(顧客の発注/品質検査能力, 0.2, 10000)
(10)顧客への商品出荷=顧客の発注/品質検査能力

以下の4章は、Business Process Analysis Workshops: System Dynamics Models, by Craig W. Kirkwood, 1998 の内容で、これは、今までやってきた、System Dynamics Method: A Quick Introduction の応用のような位置づけのものです。今まで取り上げてきた話題やモデル、技法を使って、経営問題の原因を特定化し、解決するやり方を示しています。

11. 品質

ここでは、ある中小製造業の品質検査プロセスの業務改善について取り上げています。

(1) 品質に関する印象

ここで対象としている企業は、中小で電子部品を作っているメーカー、FEC: Future Electronics Company です。この会社の経営者が、ある顧客から、「貴社の製品の品質や納品状態には基本的には満足してはいるが、時として、納品の中には不満足な品質のものが含まれ、納期遅も見られる。そういったことが発生すると、わが社の製品の納期にも関係するので、貴社よりも大手の企業であっても、わが社としてはチェックさせていただいて、発注を決めている。」と言われた。この経営者は、この会社からは大量の返品を受けることもあれば、すっきり納品を認めることもあることを思い出しました。製品品質は FEC にとって重点に置いている事項でもあるので、この FEC の経営者は、品質工程にもっと従業員をかけて、もっと品質検査を重要視しなければならないと思いました。

ここでの経営者の意志決定は、品質検査員の雇用を増やし検査率を上げるか、品質検査プロセスを改善し、不良品の見逃しを減らすことになります。検査技術の問題もあるでしょうが、すでに技術的には解決され、検査技術の問題はないと思われます。

(品質検査の業務プロセス)

この会社の品質検査工程では、熟練工が、マンツーマンで新規採用者に教育し、一人前に検査できるように訓練するというシステムを採択しています。このため、熟練工は、約半分の能力をこの非熟練工のトレーニングに割いています。非熟練工には、早く検査技術を習得し一人前になる人もいれば、なかなか習得しない人もいてばらつきがありますが、平均3ヶ月のOJTで一人前(熟練工)になります。

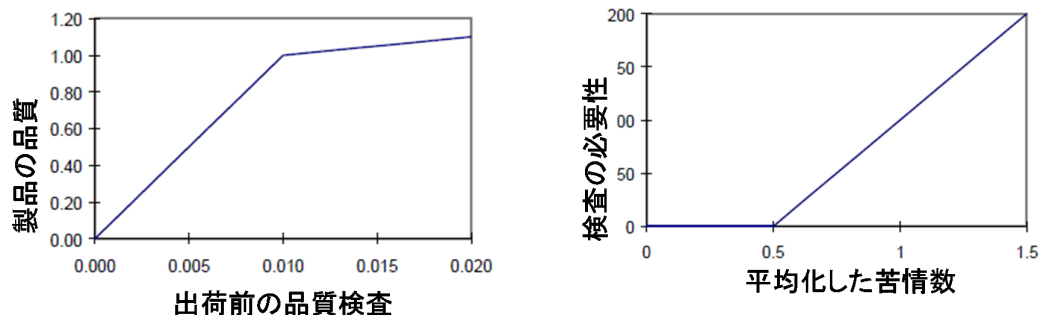


図 11-1 : 品質と苦情数

出荷前の品質検査では、品質検査能力が 0.01 の時は 100 個当たりに対して、それ以上であれば 10 個当たりに対して行っています。

品質検査の必要性は、平均苦情数が 0.5 以下であれば無いと考え、しかし、0.5 以上になったら、製品 200 個に対して一人の検査工が確保できるような検査工の数が揃っているか見て、不足すれば、必要な人員の雇用を考えます。

(2) 最初のモデル

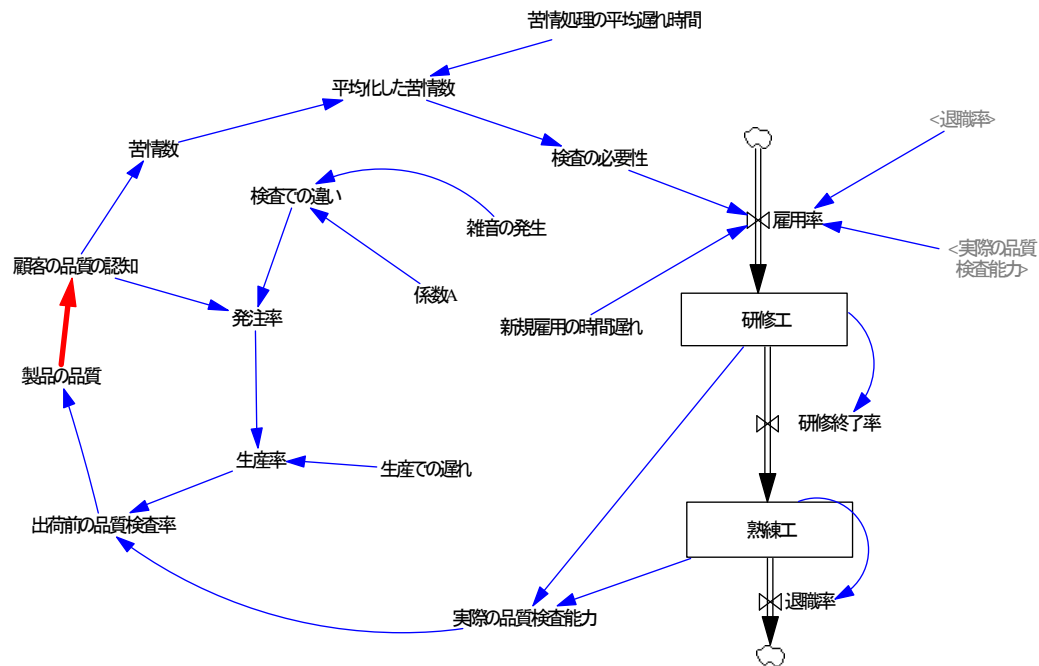
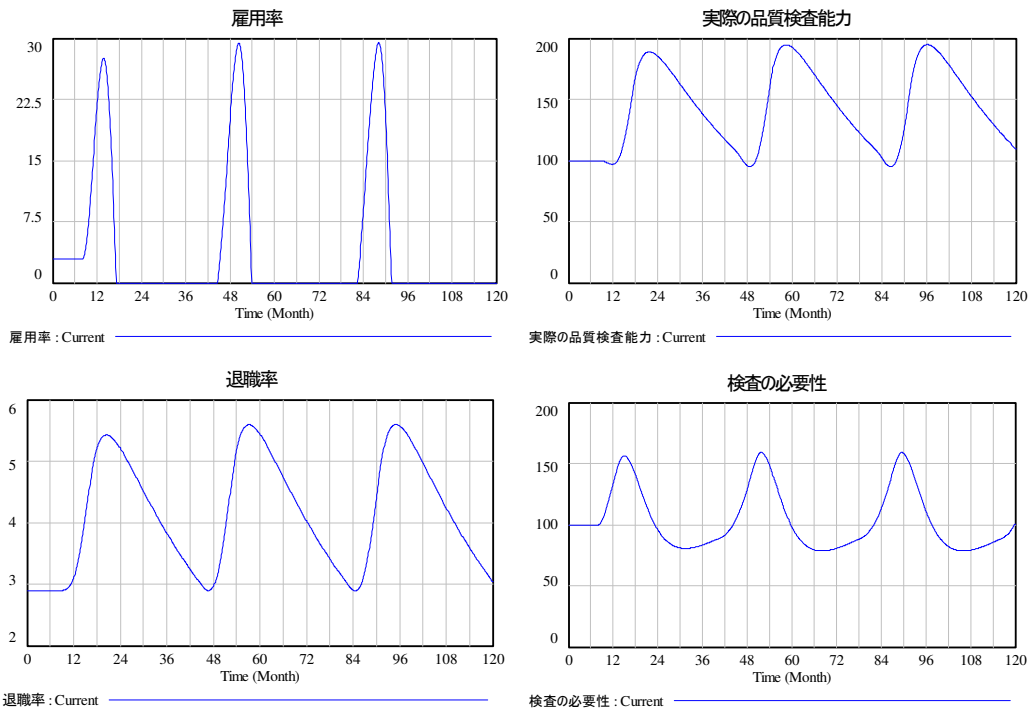


図 11-2：最初のモデル

- (05) 係数 A=0
 (06) 出荷前の品質検査率=実際の品質検査能力/生産率
 (07) 実際の品質検査能力=熟練工-0.5*研修工
 (08) 平均化した苦情数=SMOOTH(苦情数,苦情処理の平均遅れ時間)
 (09) 新規雇用の時間遅れ=2
 (10) 検査での違い=STEP(0.2,5)*((1-係数 A)+係数 A*RANDOM UNIFORM(-0.5,0.5,雑音の発生))
 (11) 検査の必要性=IF THEN ELSE(平均化した苦情数<0.5,0,200*(平均化した苦情数-0.5))
 (12) 熟練工= INTEG (研修終了率-退職率,100*24/23)
 (13) 生産での遅れ=3
 (14) 生産率=DELAY FIXED(発注率,生産での遅れ,発注率)
 (15) 発注率=10000*顧客の品質の認知*(1+検査での違い)
 (16) 研修工= INTEG (雇用率-研修終了率,(3/36)*(100*24/23))
 (17) 研修終了率=研修工/3
 (18) 苦情処理の平均遅れ時間=2
 (19) 苦情数=(3/顧客の品質の認知)-2
 (20) 製品の品質=IF THEN ELSE(出荷前の品質検査率<0.01,100*出荷前の品質検査率,1+10*(出荷前の品質検査率-0.01))
 (21) 退職率=熟練工/36
 (22) 雇用率=MAX(0,退職率+(検査の必要性-実際の品質検査能力)/新規雇用の時間遅れ)
 (23) 雑音の発生=1013
 (24) 顧客の品質の認知=SMOOTH(製品の品質,6,1)



シミュレーションしてみると、品質にむらが発生していることが分かる。

図 11-3：最初のモデルでのシミュレーション結果

図 11-2 に、最初のこの品質検査業務プロセスの SD モデルを、図 4-3 にそのシミュレーション結果を示しています。シミュレーションしてみると、検査の必要性の変動から、確かに品質にむらが発生していることが分かります。

(3) 最初のモデルに関する事項 (初期条件設定)

時間単位はここでは 1 ケ月にしています。1 ケ月間に苦情数が 1 件であれば、「実際の品質検査能力」=1 となります。もし「発注率」に変動が無ければ、変数は全て一定になるはずですが、従って、「研修工」も「熟練工」も一定になり、その検査数である「実際の品質検査能力」も一定になるはずですが。

とすれば、「発注率」も、「製品の品質」も一定になるはずですが。従って、1 ケ月間の苦情数は 1 件になるはずですが。つまり、これは、均衡状態とよばれている状態で、全ての変数は変動しないはずですが。

ただ、それでは問題が特定できないので、このモデルでは、この均衡状態に変化を与えるために、「検査での違い」というパラメーターを入れて、ステップ関数で 5 ケ月目に変化するようにしています。ただ、最初の 5 ケ月までは「検査での違い」はゼロであるべきです。そして、5 ケ月後からは、-0.1 から+0.1 まで、乱数変化させて、変化を与え、システムの反応を見ています。

このモデルが均衡モデルであることを確認するために、「検査での違い」で変動を無くし、均衡状態になることを確かめます。上の 10 式を変更し、変動をゼロにします。

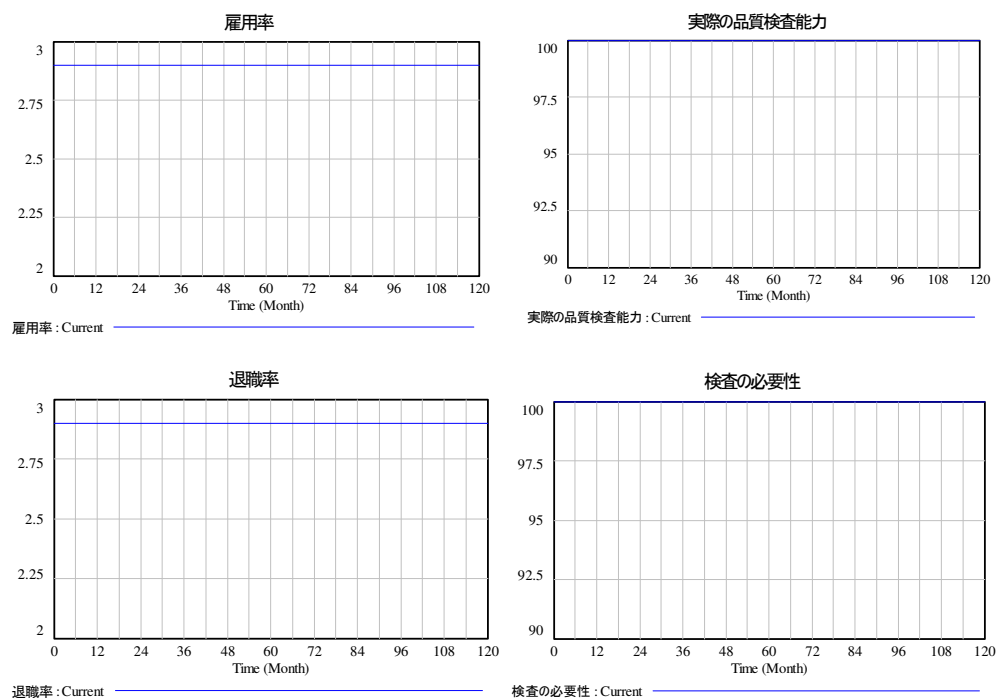


図 11-4：検査での変動が発生しないようにしたモデルでのシミュレーション結果

図 11-4 にその結果を示していますが、「検査での違い」で変化を発生させないようにすると、確かに均衡状態になっています。このことから、システムが均衡であることは確認できました。それでは、初期値を大きくしてみます。もし、均衡になっていれば、均衡ループが機能して、収斂するはずです。そこで、ストックである、「研修工」の初期値を、例えば、150 に変えてみます。(変える前の初期値はほぼ 8.7。)

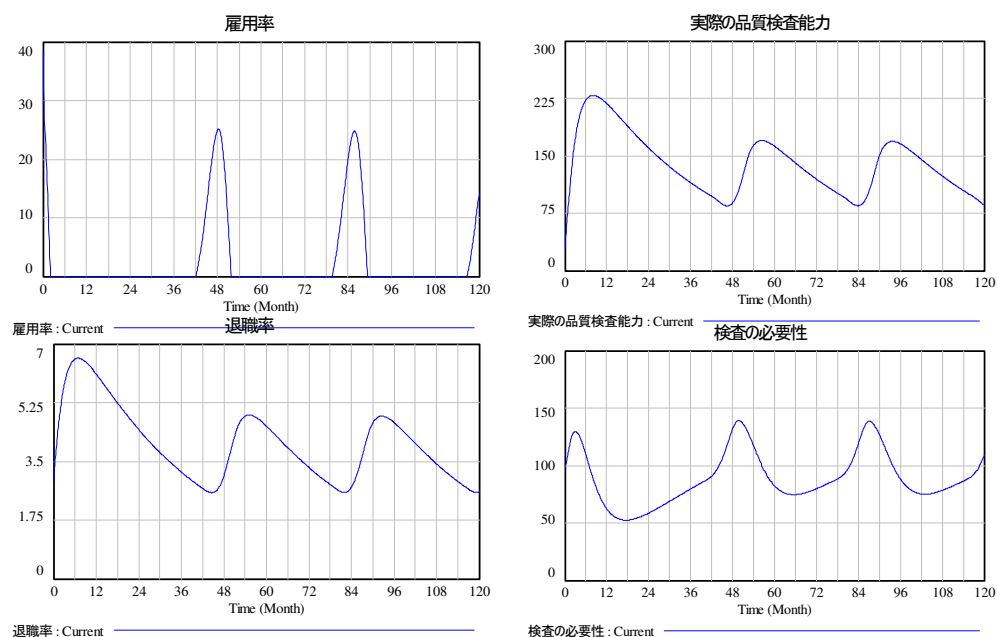


図 11-5：初期値を大きくはずしてのシミュレーション結果

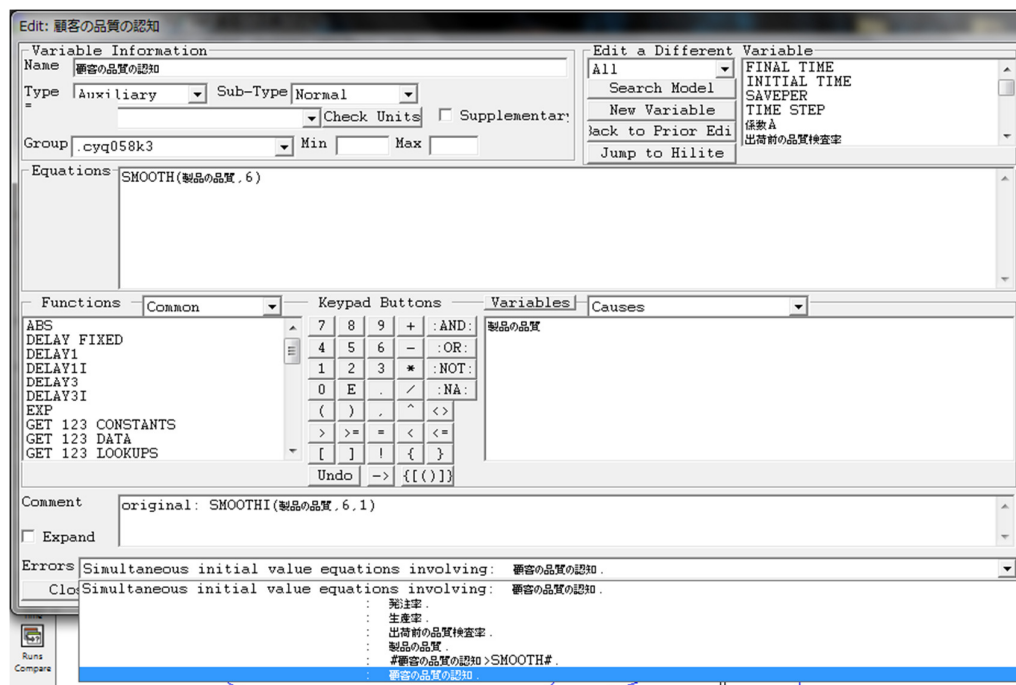
図 11-5 にその結果を示していますが、初期値を変えると、とたんに振動が発生していることが分かります。しかも、収斂するのではなく、永久に振動しているように見えます。つまり、
 仮説 1) 構造の問題：システムのフィードバック構造が、変動を産むような均衡ループになっているか、
 あるいは、
 仮説 2) パラメーターの問題：遅延などの変動要因を引き起こすパラメーターがあり、それがからんで、変動を引き起こすようになっている。
 のではないかということが推測されます。
 そこで、何が品質変動を引き起こしているのかを、SD を使って解明することとします。

(自動同時初期値設定)

「製品の品質」のループで、変数が相互依存しているので、初期値を、モデルを作成した人が決めてやらないとエラーになります。変数が相互依存しない構造のモデル（ループしない構造）であれば、初期値は Vensim の機能を使って自動初期設定することも可能です。例えば、「平均化した苦情数」に SMOOTH 関数を使うと、このエラーが起きます。

SMOOTH 関数は、遅れ関数と同じなので、遅れの間の値（初期値）がシステムで自動的に与えられると想定しています。しかし、相互依存でループになっていると、初期値がどこからも与えられないので、実行時エラーになります。これを防止するために SMOOTH ではなく、SMOOTHI を使い、初期値に 1 を与えています。

ただし、このループ構造でも、どこか 1 ケ所で初期値が与えられればいいので、1 ケ所 SMOOTHI が使われれば、その他は SMOOTH でいいし、DELAY FIXED を介在させてもいいでしょう。



こんな実行時エラーになる。これを防ぐためにSMOOTHIにする。

図 11-6：エラー

(非線型性に関する事項)

非線型にするために IF THEN ELSE を使っています。「製品の品質」は「出荷前の品質検査率」に依存するようになっていました。つまり、「出荷前の品質検査率」が 0.01 (人/月) であるか、それ以上であるかで「製品の品質」が違ってきます。最適な「出荷前の品質検査率」は果たしてどれだけなのでしょう。0.01 以上なのでしょう、それとも 0.01 以下なのでしょう。

同じく、「検査の必要性」も、「平均化した苦情数」で決めています。これも適切なのでしょうか。

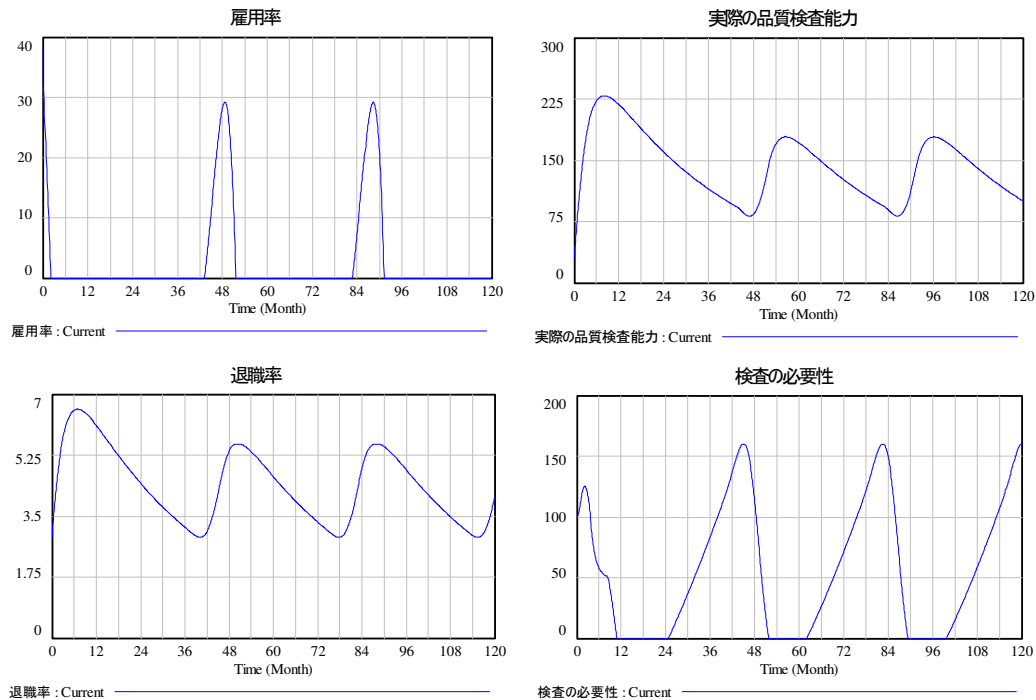


図 11-7 : 「出荷前の品質検査率」を 0.01 から 0.02 にしたもの

「出荷前の品質検査率」を 0.01 から 0.02 にして、シミュレーションを実施してみます。検査の必要性はのこぎり型になり、ゼロになることもあるのですが、しかし、変動が無くなるわけではないことが分かります。「出荷前の品質検査率」は、0.01 よりも 0.02 の方が、検査の必要性がゼロになることもあるので、経営コストを考えなければいいに決まっているが、0.01 で悪いわけではなさそうです。

(遅延関数に関する事項)

遅延には、DELAY FIXED のように、そのまま遅延するものと、DELAY3 のように指数遅れのものとがあります。この遅延の違いが影響している可能性はないのでしょうか？仮に DELAY3 のような指数遅れにしてみたらどうなるのでしょうか？

図 4-8 に遅延のタイプを DELAY3 に変えたものを示しています。シミュレーション結果はほとんど変わりません。図 4-9 には、両者を比較したものが示されています。上が DELAY FIXED、下が、DELAY3 です。DELAY3 にすると、周期が若干早まること以外はあまり変化がないことが分かります。このことから、遅延のタイプの違いが品質のムラの問題を発生させているわけではないことが分かります。

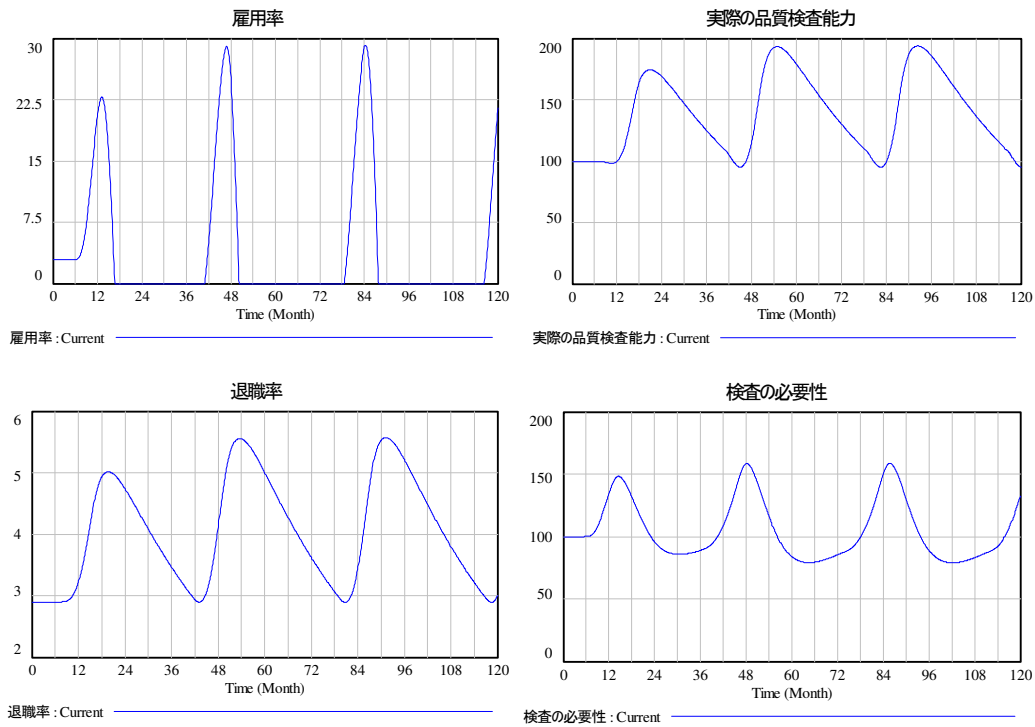


図 11-8：遅延のタイプを DELAY3 にしたもの

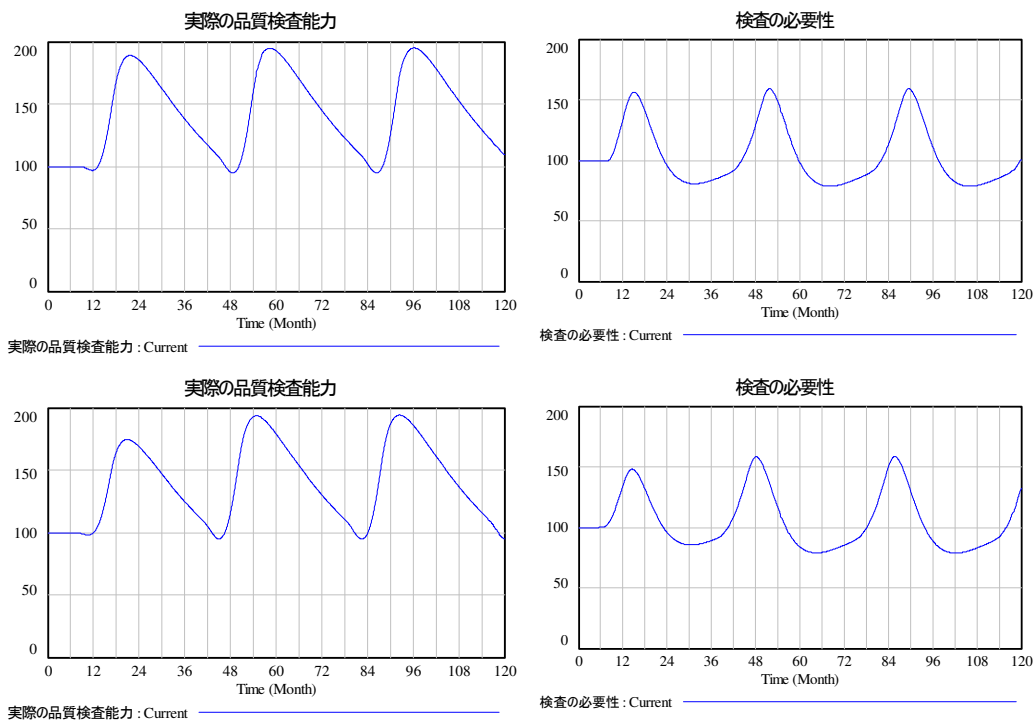


図 11-9：遅延のタイプを DELAY3 にしたものとの比較

これまでで、IF THEN ELSE による非線型性で振動が起きているわけではなく、また、DELAY のタイプの違いで振動が起きているわけではないことが分かりました。

「実際の品質検査能力」が「発注率」、「雇用率」、「製品の品質」（ここでは「検査の必要性」で見る）にどれだけの影響をもたらすのでしょうか？

遅れは DELAY 関数と SMOOTH 関数で与えています。グラフで見るとこれらは変動しています。仮に、発注率を 5 倍にした場合、品質検査への負荷がどれだけののでしょうか？（どれだけ余分に品質検査を実施する必要があるのでしょうか？）

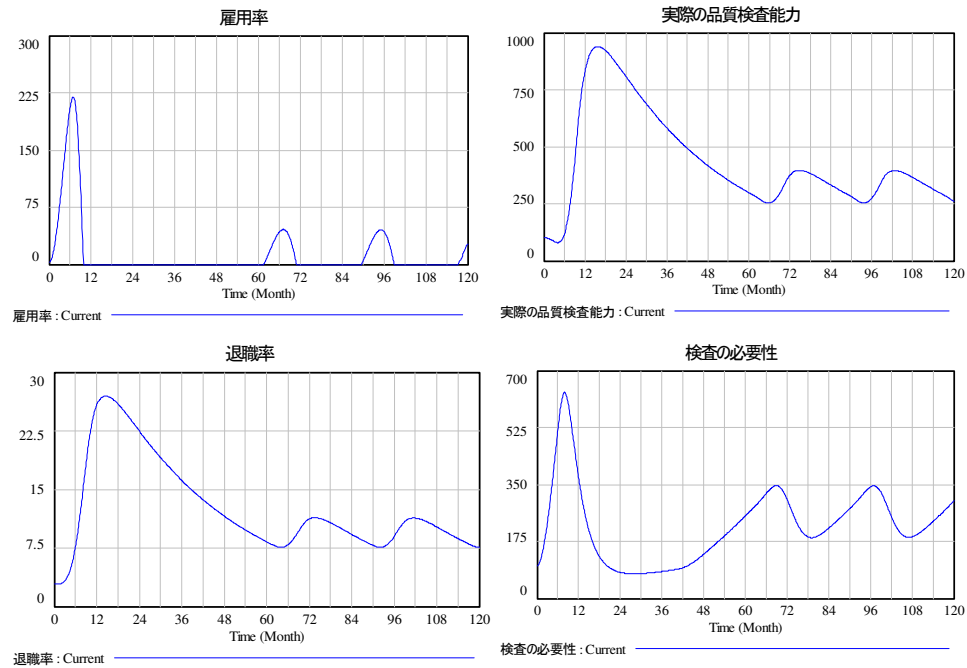


図 11-10：発注率を 5 倍にしたもの。乱数変動無し。

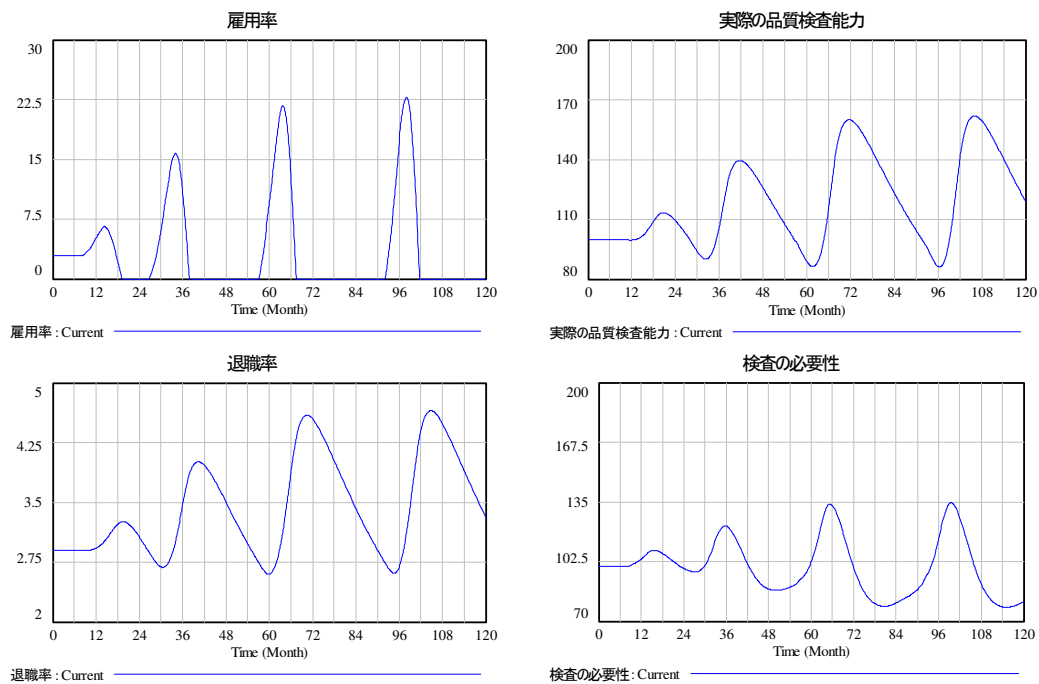


図 11-11：乱数変動ありにしたもの。

図 11-10 に発注率を 5 倍にしたものを示しています。ここでは乱数変動無しにしています。検査の必要性はピーク値で 150 から 600 にまで高くなりますが、変動パターンは同じです。従って、あまり変わらないことが分かります。

それでは、乱数変化を加えるために、係数 A を 0 から 1 に変えてみます。図 11-10 に乱数変化を加えたものを示しています。やはり、あまり変わらないことが分かります。

つまり、顧客側のテストに乱数効果を加え、顧客のテストのやり方が少しいいかげんでばらつきがあり、それが原因でこんな変化が生まれたのかどうかをチェックしてみました。ここでは、乱数変化のみにしてみました。検査の必要性の振れ幅は、乱数のみなので小さくなっていますが、変動パターンは変わりません。このことから、顧客側の発注変動や、テストがいいかげんで、ムラがあって、そのせいで変動が発生しているわけではないことが分かります。

それでは、「新規雇用の時間遅れ」「生産での遅れ」「苦情処理の平均時間遅れ」を、それぞれを半分、及び 2 倍にしてみます。図 11-12 から図 11-15 までにその結果を示していますが、雇用の遅れとか、製造の遅れ、クレーム処理の遅れのせいで変動が発生しているわけではないことが分かります。

以上のことから：

- ・「苦情数」に対応して、「雇用率」を変えるという政策は、苦情数の変動に影響されやすい。
- ・また、「雇用」に遅れがあるので、変動が振動化してしまう。
- ・このことが、「出荷前の品質検査率」に影響し、「製品の品質」に影響している。
- ・「製品の品質」に影響するので、まわりまわって、「検査の必要性」に影響している。

このことから、何のことはなく、システム構造で変動を引き起こし、製品の品質の変動を作り出していると言えます。さらに、

- ・遅延のパターン（DELAY FIXED、DELAY3）はあまり影響しない。
- ・新規採用の遅れは、周期には影響するが、振動には影響しない。
- ・生産の遅れは、あまり影響しない。
- ・苦情処理の遅れはほとんど影響しない。
- ・「苦情数」に対応して、「雇用率」を変えるという政策は、苦情数の変動に影響されやすい。
- ・また、「雇用」に遅れがあるので、変動が振動化してしまう。
- ・このことが、「出荷前の品質検査率」に影響し、「製品の品質」に影響している。
- ・「製品の品質」に影響するので、まわりまわって、「検査の必要性」に影響している。

このことから、システム構造で変動を引き起こし、製品の品質の波を作り出していることが分かります。業務プロセスを変えて、顧客からの苦情ではなく、顧客からの発注量に応じて、検査の必要性を決め、それで雇用する検査工の数を確保するようにすれば、こんなバカな構造変動を抑えられると思われます。このような、業務プロセスを改善するやり方を、リ・エンジニアリングと呼んでいます。

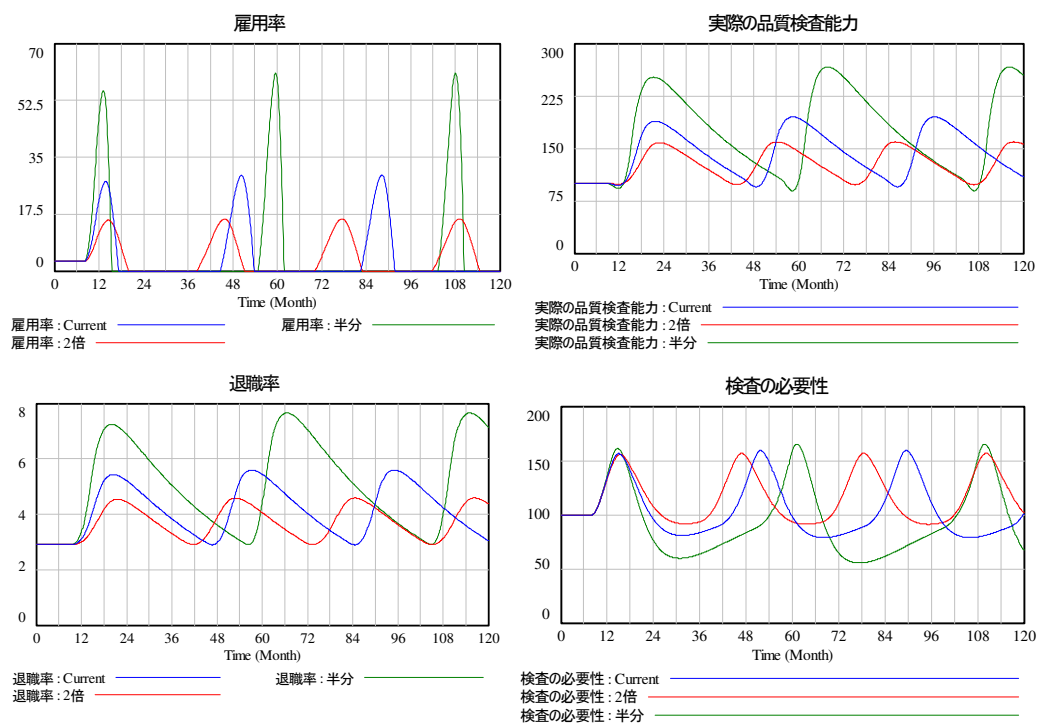


図 11-12 : 「新規雇用の時間遅れ」を半分と 2 倍と変えたもの。

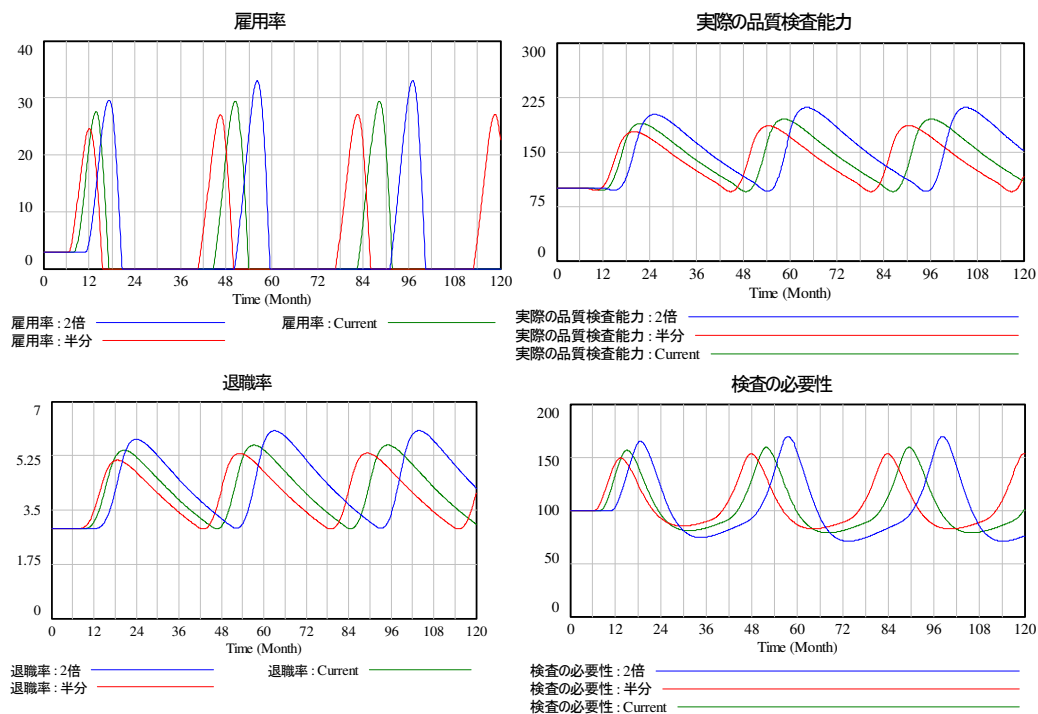


図 11-13 : 「生産での遅れ」を半分と 2 倍と変えたもの。

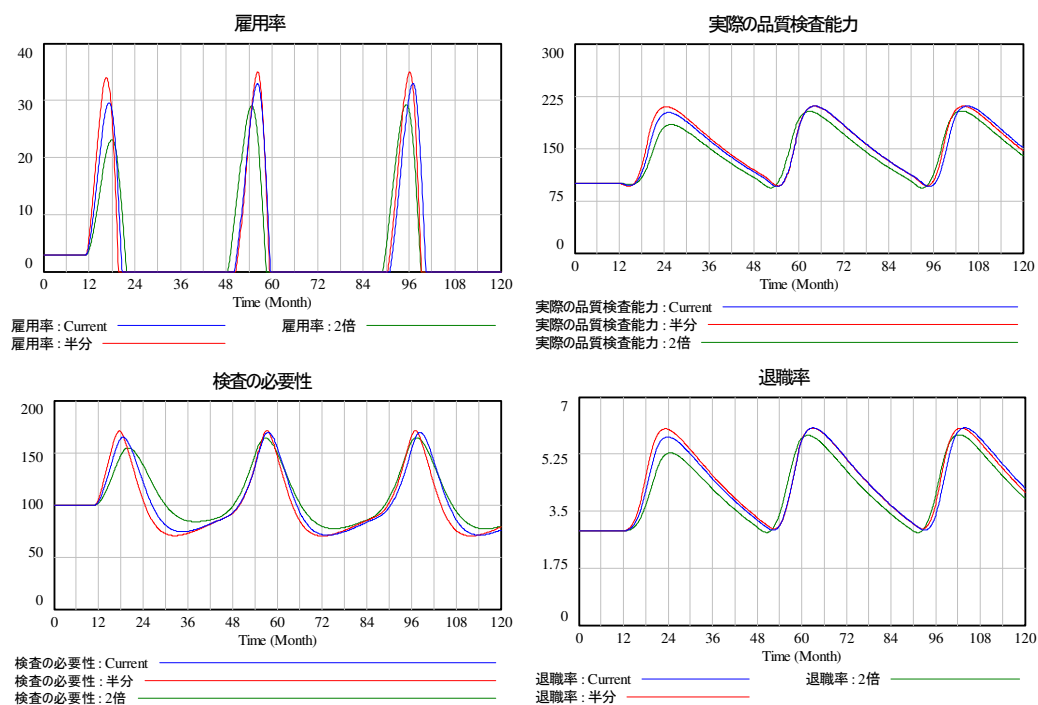


図 11-14 : 「苦情処理の平均時間遅れ」を半分と 2 倍と変えたもの。

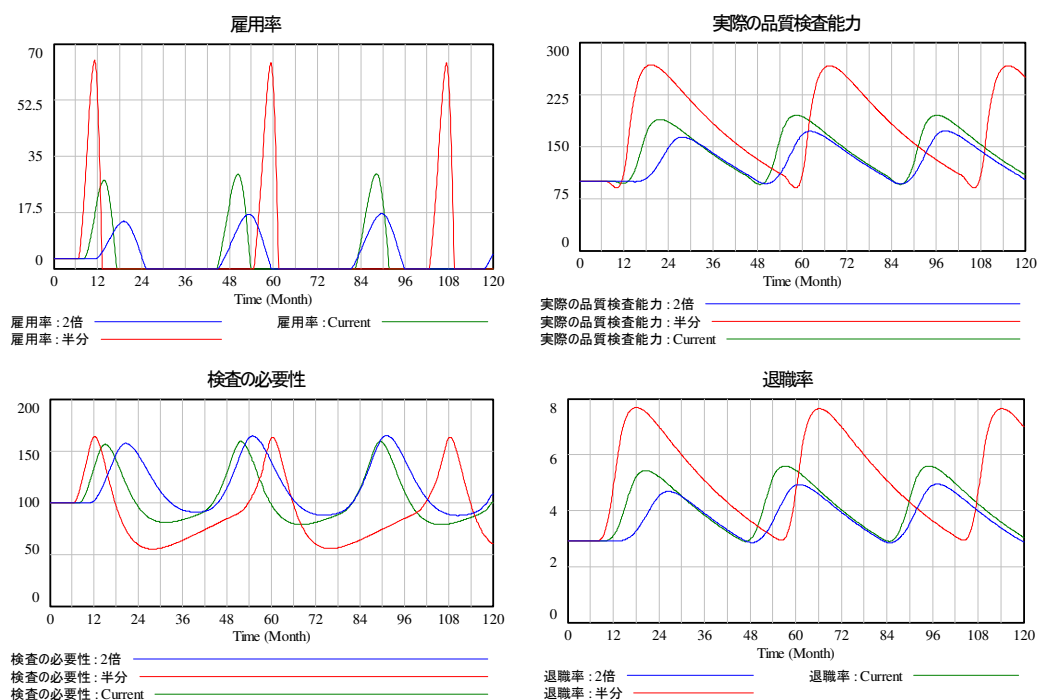


図 11-15 : 3 係数を半分と 2 倍と変えたもの。

(4) 2 番目のモデル

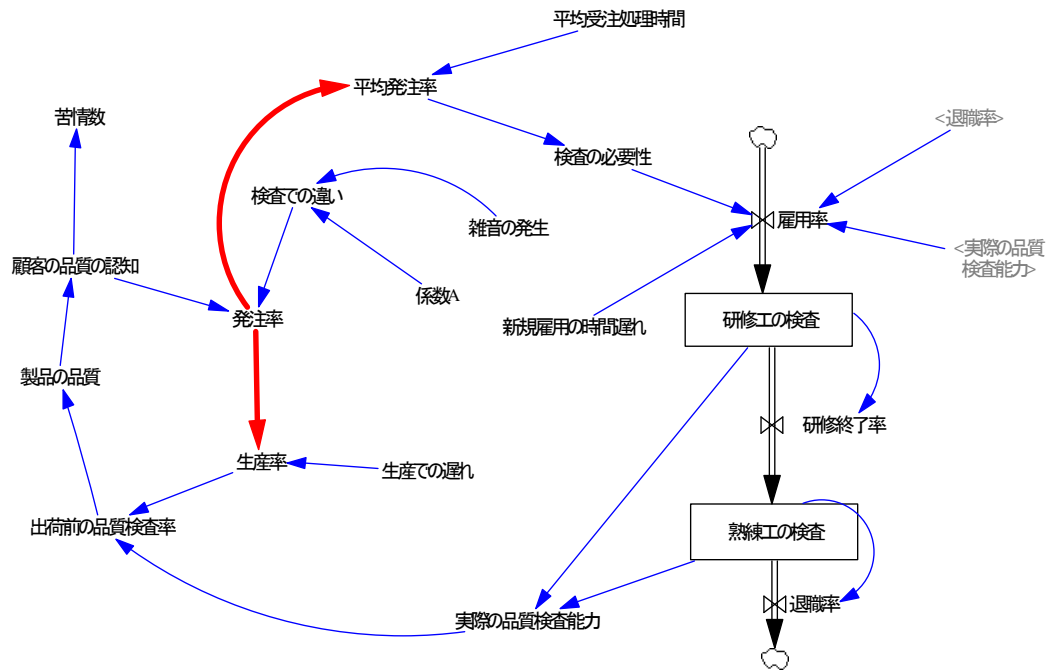


図 11-16 : 2 番目のモデル

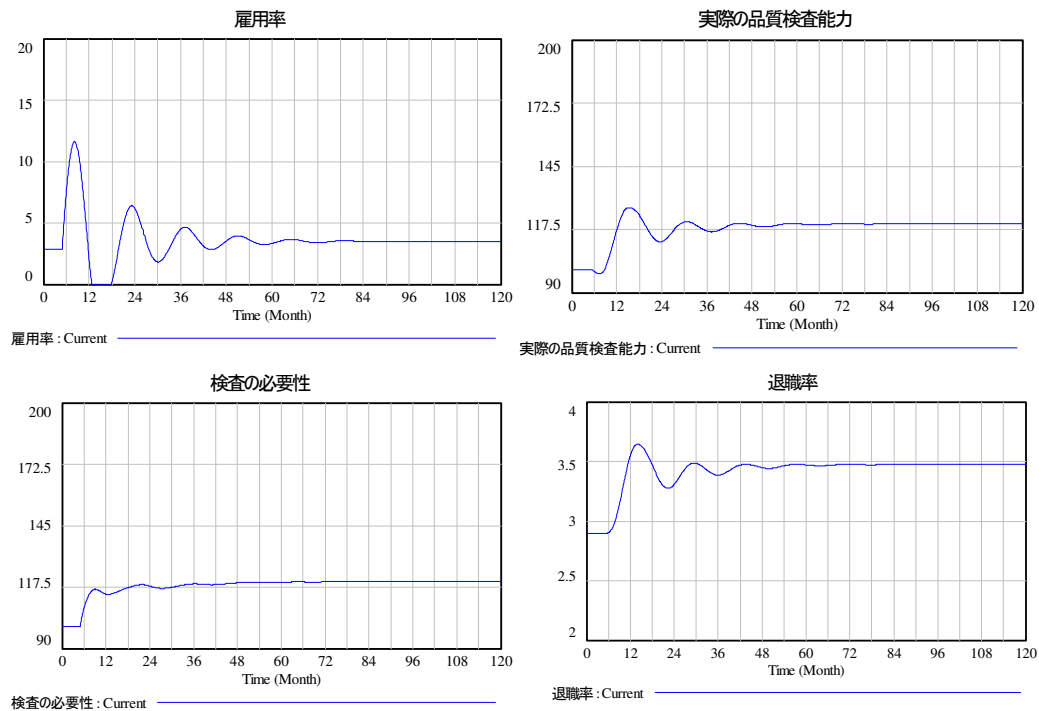


図 11-17 : 2 番目のモデルでのシミュレーション結果

業務プロセスを変更し、顧客の発注を平滑化し、その量でどこまで検査をするかを決め、

それで検査業務プロセスにかけるようにしました。平滑化した顧客の発注量の1%を検査できる人員がいるかどうかで検査工数を決め、不足すれば雇用するようにします。そのシミュレーション結果を図 11-17 に示しています。今度は変動が無くなっている（まま）ことが分かります。乱数でシステムに変動を与えても、収斂することがシミュレーション結果で確かめられました。

(5) 2 番目のモデルに関する事項

ここでは、プロセス変更を行ってはいませんが、最初のモデルにあった要素の中で不要のものも残しています。

この品質検査業務プロセスの変更で、「生産での遅れ」、「新規雇用の遅れ」、「平均受注処理時間」の遅れや低下が「製品の品質」にもっとすばやく反応するようになっています。特に、「新規雇用の遅れ」の影響が影響します。このような業務プロセス変更は、製造部門内だけではなく、人事部にも関係してくるので、業務横断的な改善が必要となります。人事部門での採用情報が適切にタイムリーに製造部門にもたらされ、OJT 体制をすばやく整える必要があります。（テキストには、どのような業務上問題があるか議論しろとなっている）

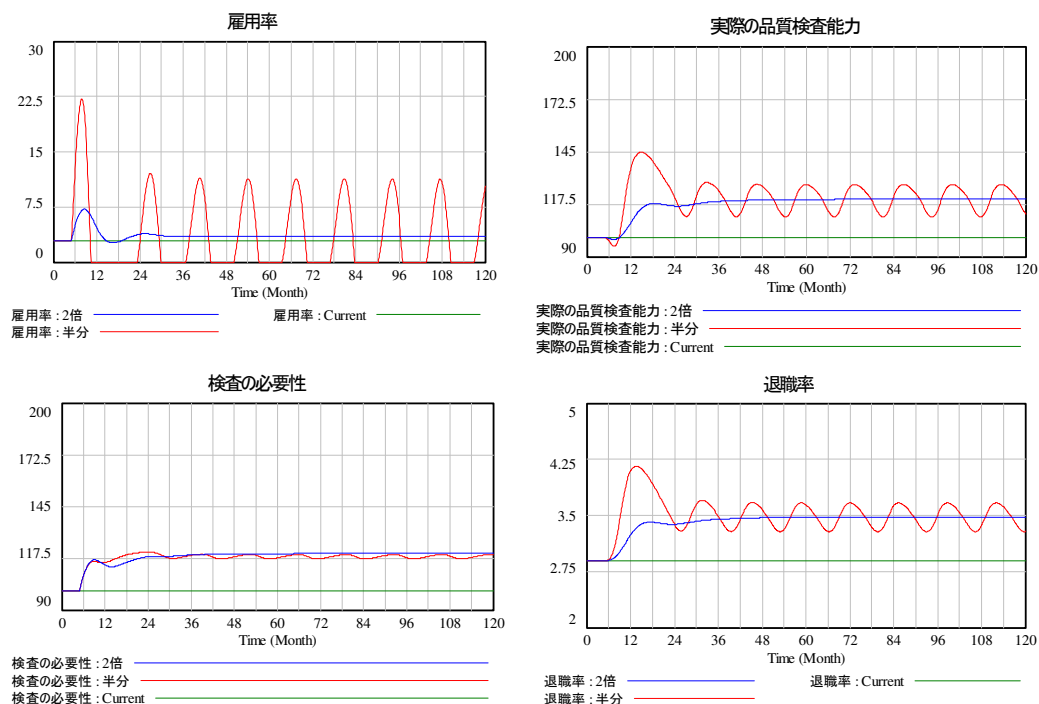


図 11-18：2 番目のモデルでのシミュレーション結果

以下は、この 2 番目のモデルの等式です。

- | | |
|------|---|
| (05) | 係数 $A=0$ |
| (06) | 出荷前の品質検査率=実際の品質検査能力/生産率 |
| (07) | 実際の品質検査能力=熟練工の検査-0.5*研修工の検査 |
| (08) | 平均受注処理時間=2 |
| (09) | 平均発注率=SMOOTH(発注率,平均受注処理時間) |
| (10) | 新規雇用の時間遅れ=2 |
| (11) | 検査での違い=STEP(0.2,5)*((1-係数 A)+係数 A*RANDOM UNIFORM(-0.5,0.5,雑音の発生)) |

(12)	検査の必要性=0.01*平均発注率
(13)	熟練工の検査= INTEG (研修終了率-退職率,100*24/23)
(14)	生産での遅れ=3
(15)	生産率=DELAY FIXED(発注率,生産での遅れ,発注率)
(16)	発注率=10000*顧客の品質の認知*(1+検査での違い)
(17)	研修工の検査= INTEG (雇用率-研修終了率,(3/36)*(100*24/23))
(18)	研修終了率=研修工の検査/3
(19)	苦情数=(3/顧客の品質の認知)-2
(20)	製品の品質=IF THEN ELSE(出荷前の品質検査率<0.01,100*出荷前の品質検査率,1+10*(出荷前の品質検査率-0.01))
(21)	退職率=熟練工の検査/36
(22)	雇用率=MAX(0,退職率+(検査の必要性-実際の品質検査能力)/新規雇用の時間遅れ)
(23)	雑音の発生=1013
(24)	顧客の品質の認知=SMOOTH(製品の品質,6,1)

12. 販売

ここでは、販売プロセスでの納期の問題の改善について取り上げます。

(1) 市場の拡大と製造能力強化

新製品を打ち出して、ものすごく売れたと思うと、とたんに売れなくなってしまうことがあります。SD 的に言えば、増強ループがあって売上が伸びて、何か均衡ループが働いて、売上が伸び悩んでいると説明できるのですが、このような状況に陥った原因はいろいろあるのですが、どのような製品市場でも、こういったことが起きると思います。競合状況が高まり、競合相手と価格競争になって、品質での差が小さければ、顧客は納期で発注を決めるでしょう。納期で発注が決まるというのは、小売業や銀行、外食産業ではよくある話です。製造業でも、消費財市場、例えば、テレビ、オーディオ製品、カメラなどで見られる現象ですし、また、OEM 製品市場でもよく見られる現象です。もし、OEM 製品がかなり安く、販売が伸びているような状態では、納期が営業の成績を左右するようになります。このことは、部品製造であっても、完成品の PC の製造であっても当てはまることです。

(ここで OEM 製品市場と言っているのは、ブランド品ではない、性能的にはあまり変わらず、かつ価格の安い他社製品が出回っている市場といった意味です。OEM メーカーが、ブランド品メーカーに製品を供給している場合もあれば、同時に、自社製品として販売していることもあります。)

(2) モデル

図 12-1 にモデルを示しています。モデルは、営業、流通、製造の 3 セクターから構成されています。

(営業セクター)

営業マンの人数は、「受注」から「収入」が決まり、その収入に対する「営業費用配比率」で営業マンの人員数目標（望ましい営業人員数）を決めています。ただし、雇用までの時間遅れがあります。一応、利益の中から、どのくらい営業費用を割り当てられるかで営業マンの人数が決まるようになっています。

しかし、売上に関しては、顧客の「納期の認知」が、顧客の発注を決め、それが「営業効率」になり、「受注」になるようになっています。つまり、受注効率が、顧客が感じている納期の長さに対する印象で決まるようになっています。従って、営業マンの数とこの「営業効率」で受注が決まるわけです。

製造能力の割り当ても、納期（リードタイム）に影響されるようになっています。納期が短い受注に対しては、製造能力の割当が小さくなり、長ければ、あまり問題なく、99%まで割り当てられます。従って、営業マンは、納期が長いものであれば受注できますが、納期が短い受注はあまり取れません。ここは非線型です。

(流通セクター)

営業マンの受注は、受注残というストックに蓄積され、製造が終わったものから逐次出荷されます。直接的には、出荷状況と受注残の状況によって「納期」が決まるのですが、これは、受注と製造能力によって「納期」が決まるようなものです。

(製造セクター)

設備投資の必要性を感じて設備投資を行うのですが、そこに遅れが存在します。「設備投資の必要性」=1 の時、設備投資が行われ、0 の時は行われないようにしています。また、設備投資は「設備投資量」分行われます。

設備投資とここでは言っていますが、内容を厳密に分けていなく、新しい工場を建設することであったり、既存の製造会社を買収することであったり、あるいは単に製造担当の人員を増強するだけであることもあります。ここでは、単純に、製造能力が増強されるこ

としか考えていません。

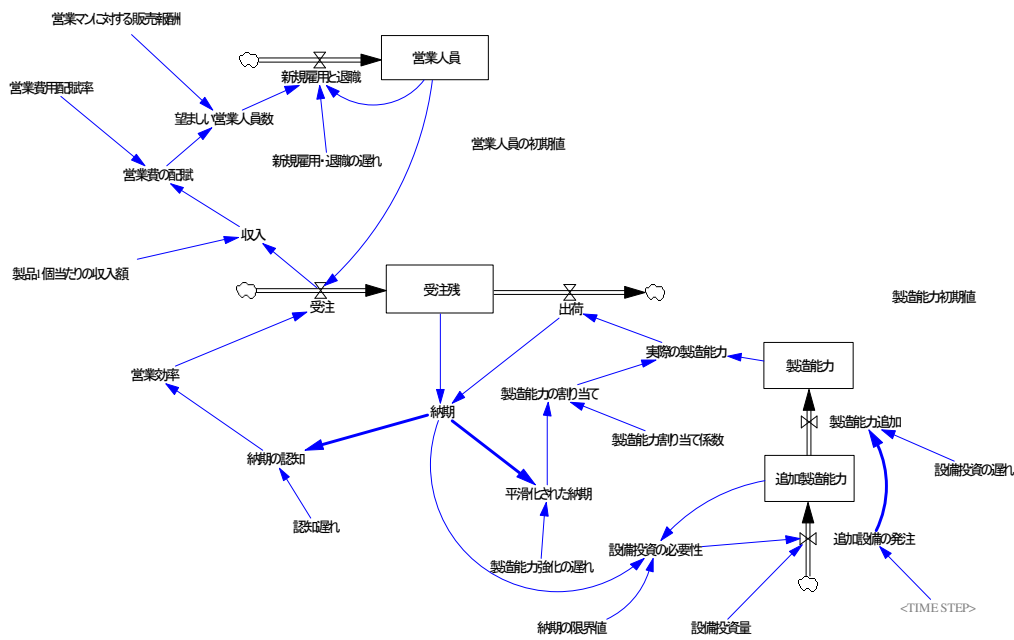


図 12-1：販売モデル

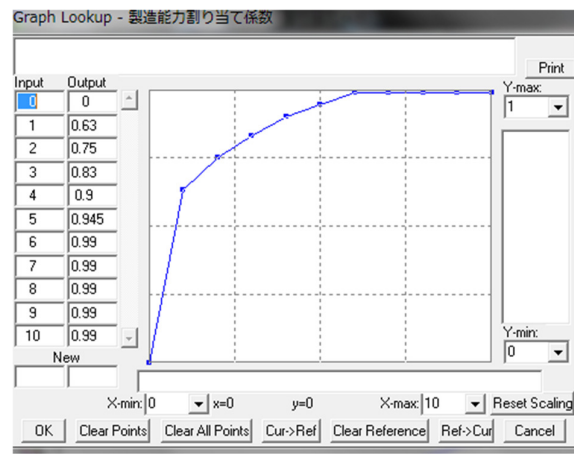


図 12-2：設備能力の割り当て：製造能力の割当。納期が短いほど、製造能力はあまり割り当てられない。

設備投資は、オイラー法を使って、TIME STEP で割った量だけ増えていくようにしています。つまり、「設備投資量」÷「TIME STEP」分の四角形分が「追加設備能力」になり、それが累積された分が、「設備投資の遅れ」分だけ遅れて、「製造能力」に加わり、「製造能力」が増えるようにしています。ここは、こういった風にしないで、「設備投資量」が、「設備投資の遅れ」を伴うものの、「設備投資期間」に「設備能力」に加わるように変えることも可能ですが、ただ、これだと、製造能力がすぐに増えてしまいます。また、前の方法では、計算方法で、ルンゲ・クッター法を選択すると、エラーになってしまう可能性があります。

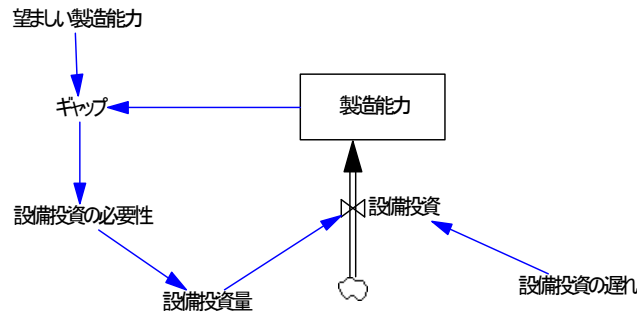


図 12-3：設備投資の別のやり方

設備投資では、図 12-3 のようにすることも可能です。もし、製造機器を導入するというのであれば、機械などは一気に購入し、据え付け、あとは受注に応じて稼働を増やしていくだけなので、線型で、こちらに近いのですが、もし、人員を増やすとか、工場建設で、第 1 工場建設、第 2 工場建設のように逐次建設していくのであれば、期や年で捻出できる資金に上限があることや、離散的な増加になるので、図 12-1 のモデルのようになるでしょう。

製造能力が強化されれば、製造が早く行われ、その結果、「出荷」が早まるので、「納期」は短縮されます。

(3) モデルを使った意志決定

Vensim PLE は、自動的に初期値を計算し、シミュレーションを進めてくれるのですが、フィードバック・ループにストックが含まれない場合、初期値を計算できないので、エラーになります。このモデルでは、「出荷」、「納期」、「平準化された納期」、「製造能力割り当て係数」、「製造能力の割り当て」、「実際の製造能力」のループには、ストックが介在しないので、エラーとなってしまいます。それを避けるためには、「平準化された納期」で SMOOTH 関数ではなく、情報指定の SMOOTHI を使い、初期値を与えるようにします。

(4) シミュレーション

72 ケ月（6 年間）のシミュレーション計算を図 12-4 から 12-6 に示しています。収入と受注を見ると、18 ケ月周期の変動が見られます。この変動の原因を考えてみましょう。

すぐに目につくのは、「営業効率」が落ちているので、これが原因ということは分かりません。「営業効率」は「納期」に影響されます。「営業効率」と「納期」を比べてみると、確かに、「納期」と反転しています。

ただ、問題は、営業マンは納期を意識しても、経営者が納期をどこまで意識して、納期を短縮するための設備投資を行ってくれるかが鍵となるでしょう。ただ、これは、経営者の意識として難しい問題です。

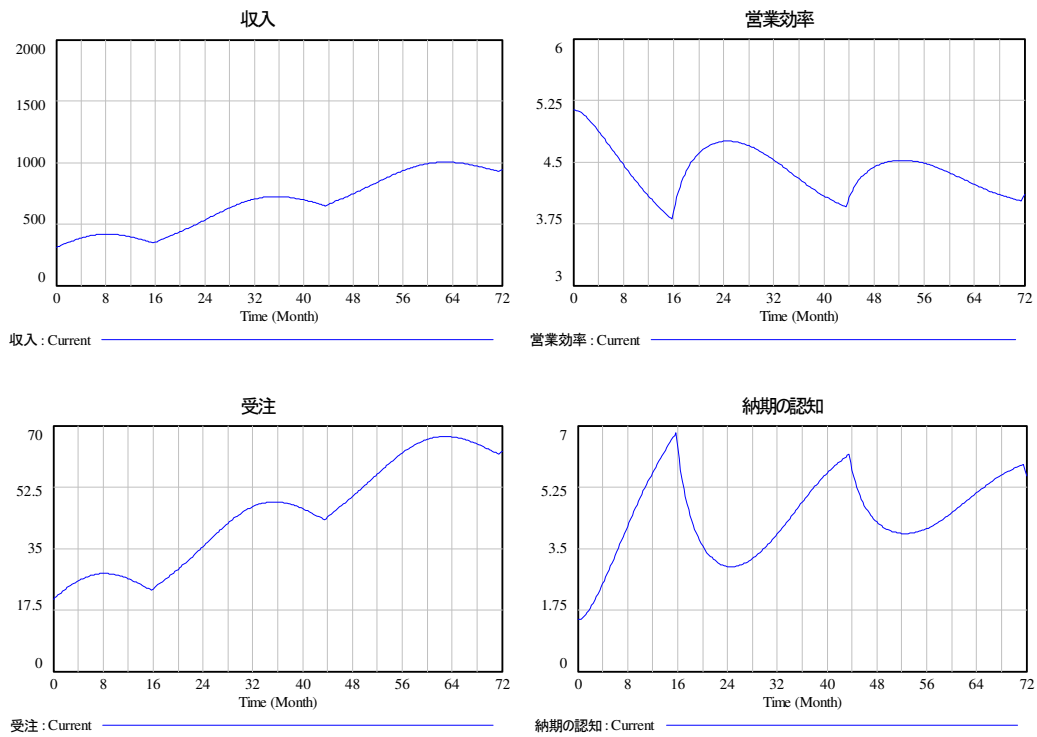


図 12-4：シミュレーション結果：収入、受注に 1 年半の波があり、それは納期の問題から来ている

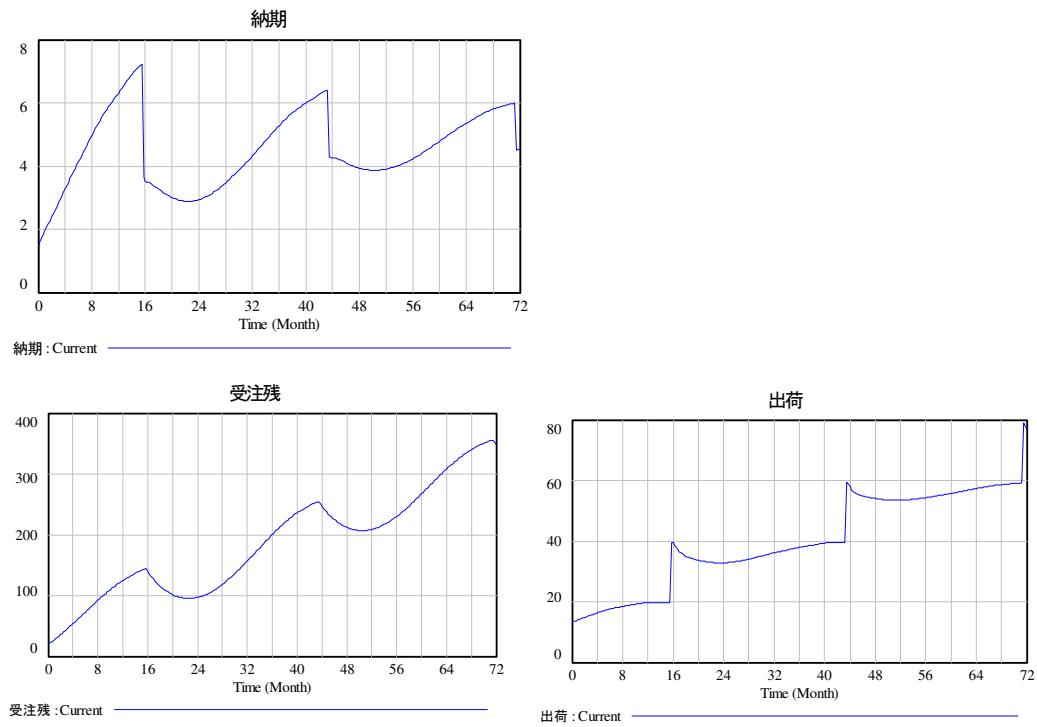


図 12-5：シミュレーション結果：納期のせいで、出荷や受注在庫が変動している。

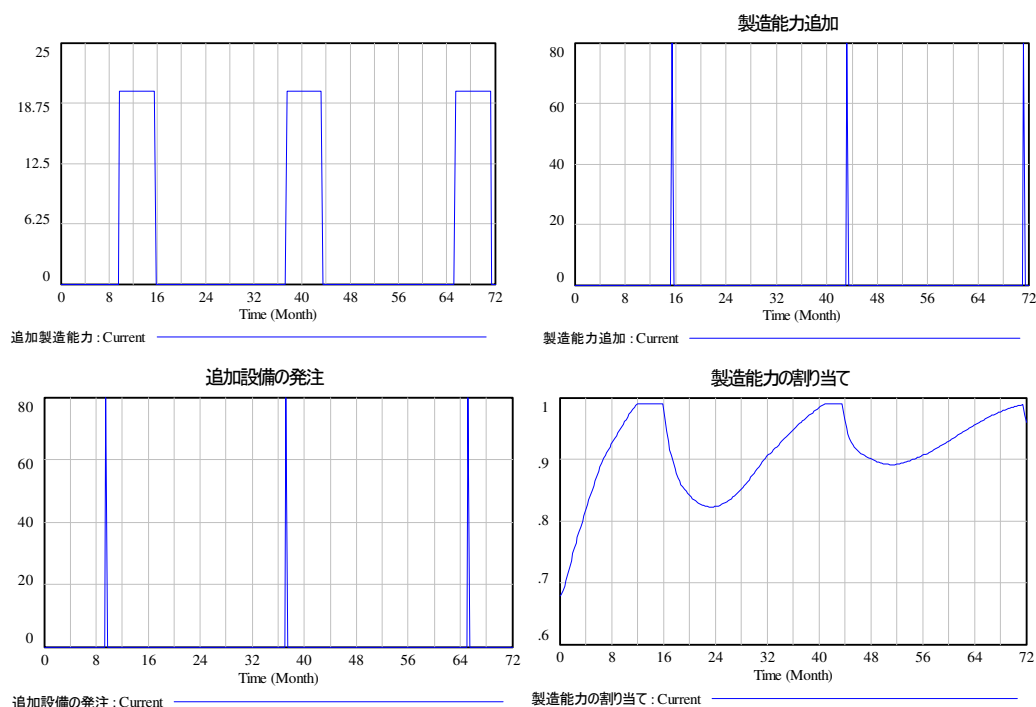


図 12-6：シミュレーション結果：製造能力の割り当ての変動で 0.99 で頭打になることが納期の変動を作り出している。

経営者は、図 12-5 の受注と収入のグラフを重視し、製品が売れているのだから、もっとどんどん積極的に販売しろと営業マンにはっぱをかけてしまいがちです。デスクワークとはかくそういったものとなりがちなもの。そうすると、納期が長くなって、製品は売れなくなります。そして、どうして製品が売れないのだと不思議に思うでしょう。営業マンが無能だという結論に達し、そして、今いる営業マンを首にして、もっと優秀な営業マンを雇おうとするでしょう。首になった営業マンには可愛そうであり、新しく雇われた営業マンにはラッキーなのですが、営業不振の間に製造能力が回復し、納期も短くなるので、製品の売れ行きはまた好調になります。経営者は、やっぱり前の営業マンは無能で、今度の営業マンは優秀だったと確信してしまうことになります。

（設備投資）

図 12-5 で、納期と受注、出荷の関係を見ると、前の受注で見られたような振動パターンが見られます。納期の変化にシンクロナイズして受注の波が生まれ、それに応じて、出荷の波も生まれていることが分かります。ただ、これを見て、製造部門の責任者が、納期の問題を意識するかどうかは疑問です。製造部門の責任者には、納期の問題よりも、受注変動の方が大問題です。受注が増えると、製造能力を超えるので、納期を長くすることで製造を調整しようとします。顧客と接しているので、営業マンには納期が長くなることは大問題なので、営業マンは伸びた納期に対し文句を言いますが、製造担当者には納期のことはピンとこないのが普通です。（営業マンにとってみれば、1 ヶ月で納品できると受注し、顧客にも 1 ヶ月で納入できますと確約し、工場に廻してみたら、納期が半年先だなどと言われたら、困るし、顧客も怒って、次からは、半年先に納品できるような量しか発注しなくなります。）

製造担当は、「営業マンは製造のことが分かっていない」ということになりがちです。

これは、設備投資を認めるかどうかを意志決定する財務担当にとっても同じで、売上が

変動するのは財務的にも問題ですが、製造能力の問題というようには見えません。財務担当者も図 12-5 の受注と収入のグラフを重視し、18ヶ月周期で売上が変動していることは認知しますが、変動しているだけにしか見えないでしょう。設備投資は中長期（短くともせいぜい3-4年）を睨んで行うので、1年半の変動は長期とは言えないので、一応、受注が伸び、収入も増えていて、その受注増加に応じた設備投資は必要とは思いますが、納期を短縮するために設備投資が必要というようには見えません。また、設備投資には比較的多額の費用がすぐにかかるので、こちらの方は、財務担当としては大問題です。（このモデルには含めていません。）こんな状況では、おいそれと、設備投資の意志決定はできないのが普通で、受注の伸びや売上の伸び、特に売上の伸びがそう急激ではないので、今の設備でもいいのではないかと、となってしまうがちです。

(5) 業績の改善（システム的な見方）

業績（売上）の伸び悩みと好調との波があるということの原因を、営業マンの能力のせいだとするのは簡単だが、そうではないことをこれまで見てきました。このことを確認し、どうすればいいかを考えることにします。

収入からの営業費用への配賦を増やし、営業マンを増やせば、売上は上がるでしょうか？例えば、「営業費用配賦率」を25%増やし、0.4から0.5に増やしてみたらどうでしょうか。

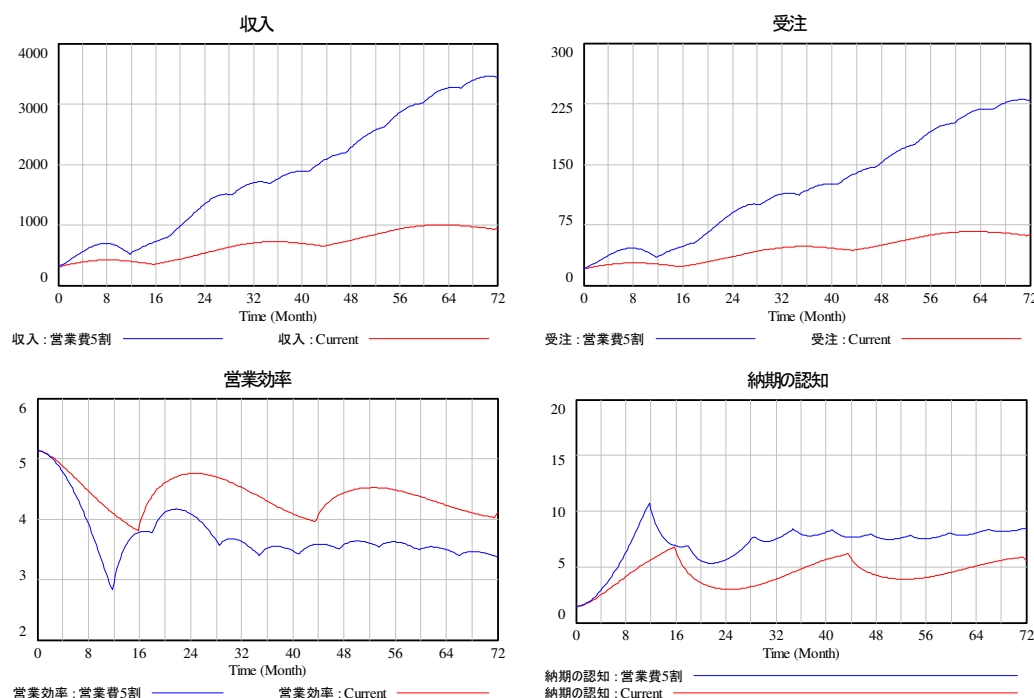


図 12-7：「営業費用配賦率」25%増のシミュレーション結果：売上は伸びるが、営業効率は下がる。

営業マンが増えたので、売上は上がりますが、売上の変動が無くなるわけではないし、納期が改善されるわけでもありません。というか、納期が延び、製造能力割り当てがフラットになります。つまり、納期が長い受注しか受けられなくなります。実際にも、営業マンを増やしても、顧客は迷惑なだけでしょ。何人も営業マンがやってくるが、サービスが改善されるわけではありません。それに営業マンを増やすと、その分営業費用が増え、会社の収益を圧迫します。そんなことは長く続けられないでしょう。また、サービスが悪いうのでは、競争相手に負けてしまうでしょう。顧客の注文は、納期が短い会社流れて

しまいます。

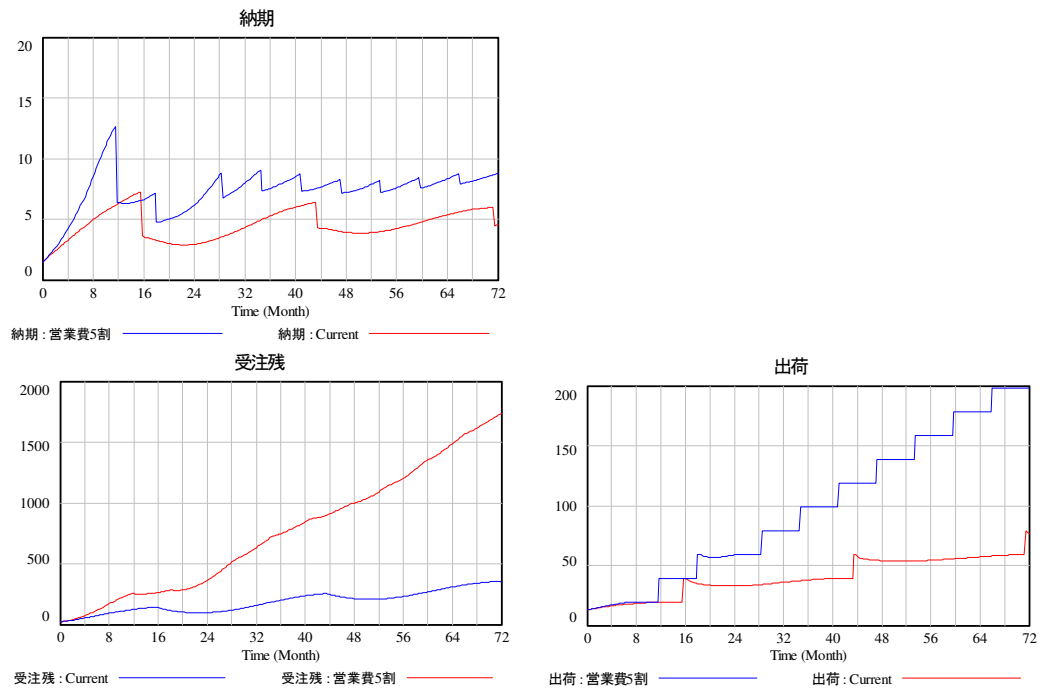


図 12-8 : 「営業費用配賦率」 25%増のシミュレーション結果：納期変動が激しくなる。

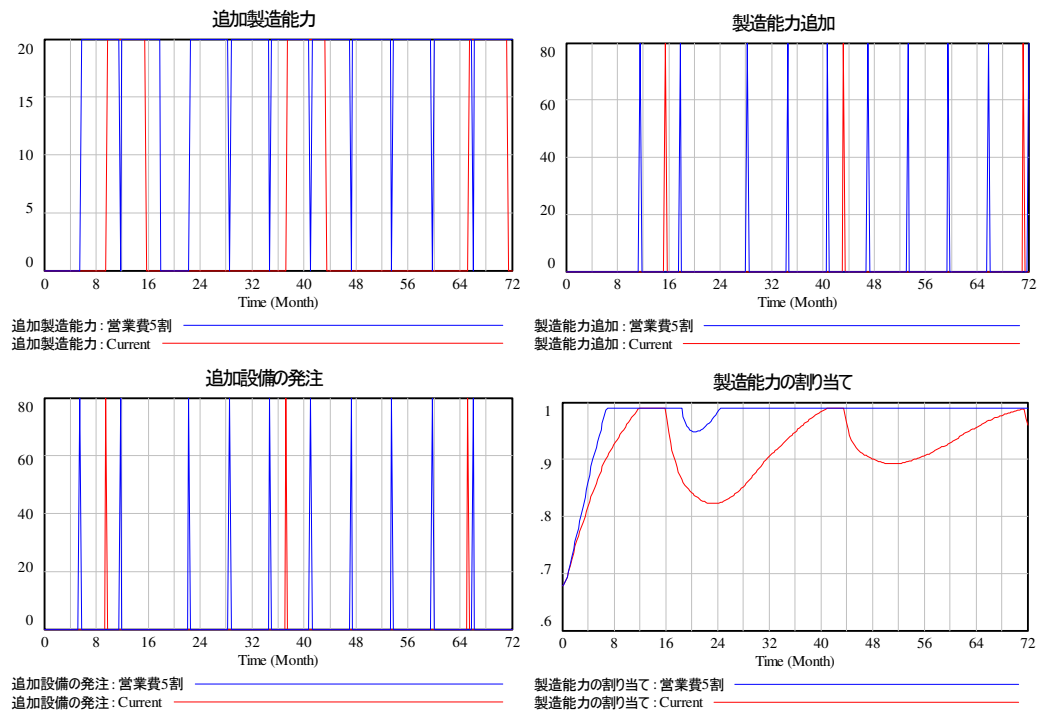


図 12-9 : 「営業費用配賦率」 25%増のシミュレーション結果

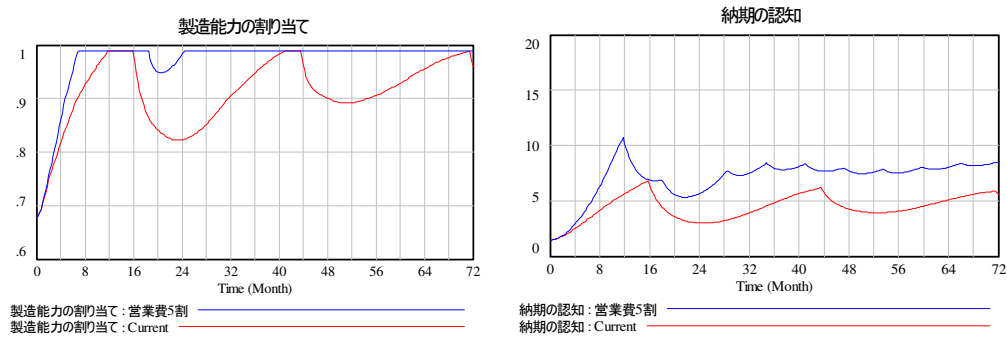


図 12-10 : 「営業費用配賦率」 25%増のシミュレーション結果

図 5-10 に示していますが、「営業費用配賦率」を 25%増やしたせいで、製造能力割り当てがほとんど 0.99 で、つまり納期が長い受注しか受け付けなく、納期が短いものは受注できないという状態になっています。納期はそれまで平均 4 ケ月だったものが、8 ケ月と 2 倍に伸びています。図 12-9 で、設備投資が増えています、これは受注が増えたことによるもので、業務プロセスが改善されたことによるものではありません。

設備投資の部分を改善してみましょう。「設備投資量」を半部の 10 にします。代わりに「設備投資の遅れ」を 6 ケ月から 3 ケ月に短縮します。これであれば、財務負担は基本的には変わりませんので、財務的には問題はありません。シミュレーションをしてみます。結果は、「収入」が減少してしまいます。

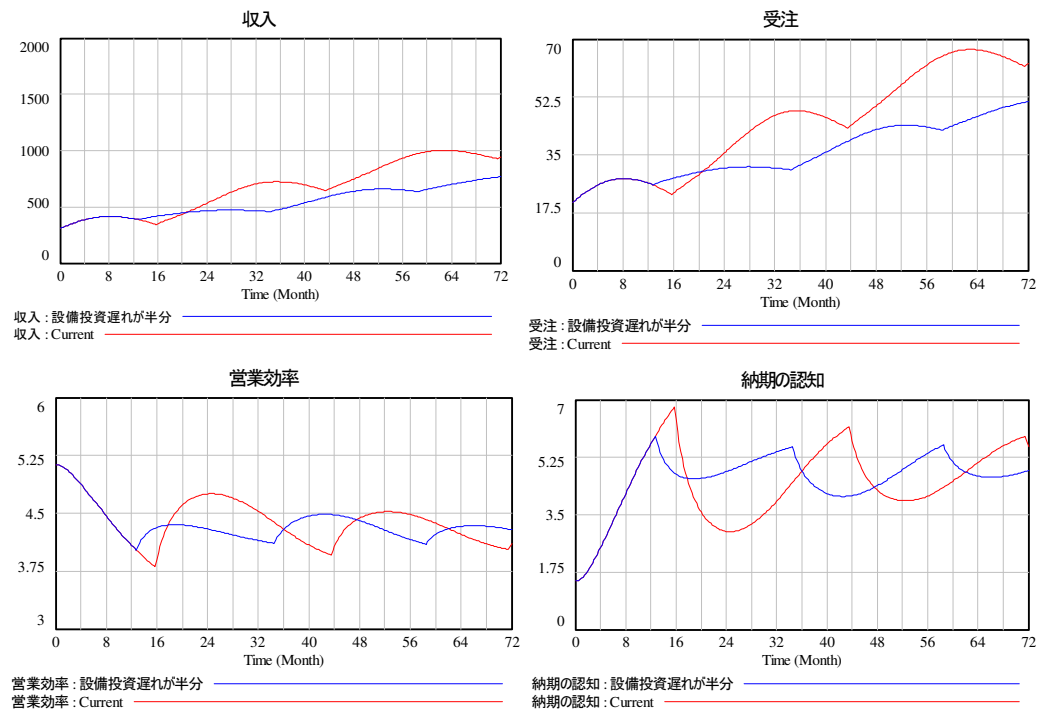


図 12-11 : 「設備投資量」 半減、「設備投資の遅れ」を半分に短縮したシミュレーション結果

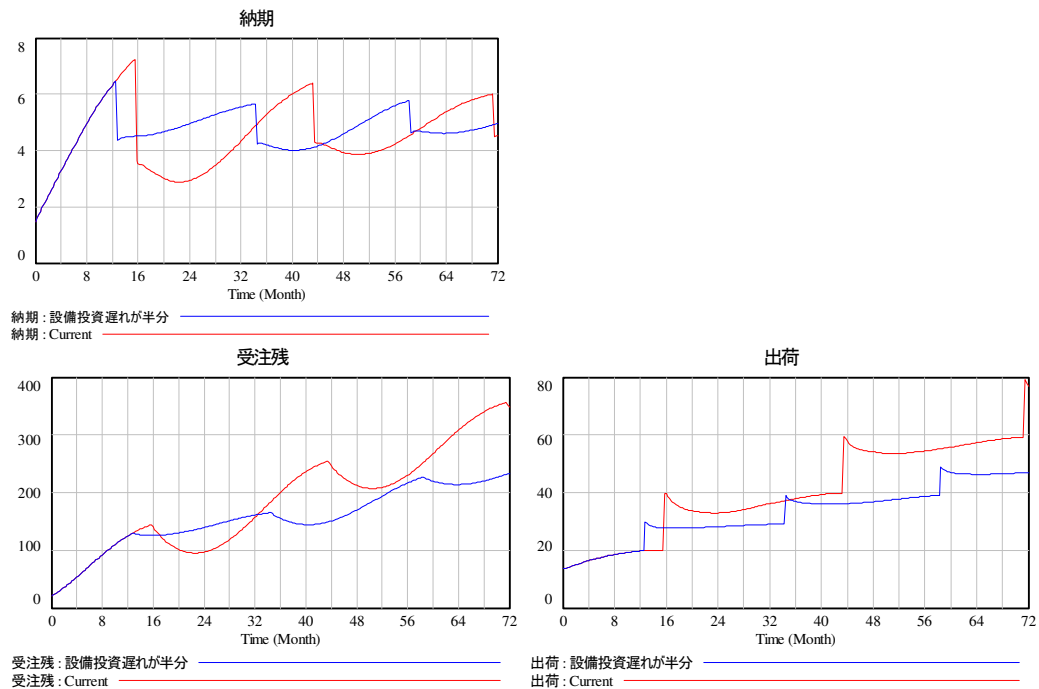


図 12-12：「設備投資量」半減、「設備投資の遅れ」を半分に短縮したシミュレーション結果

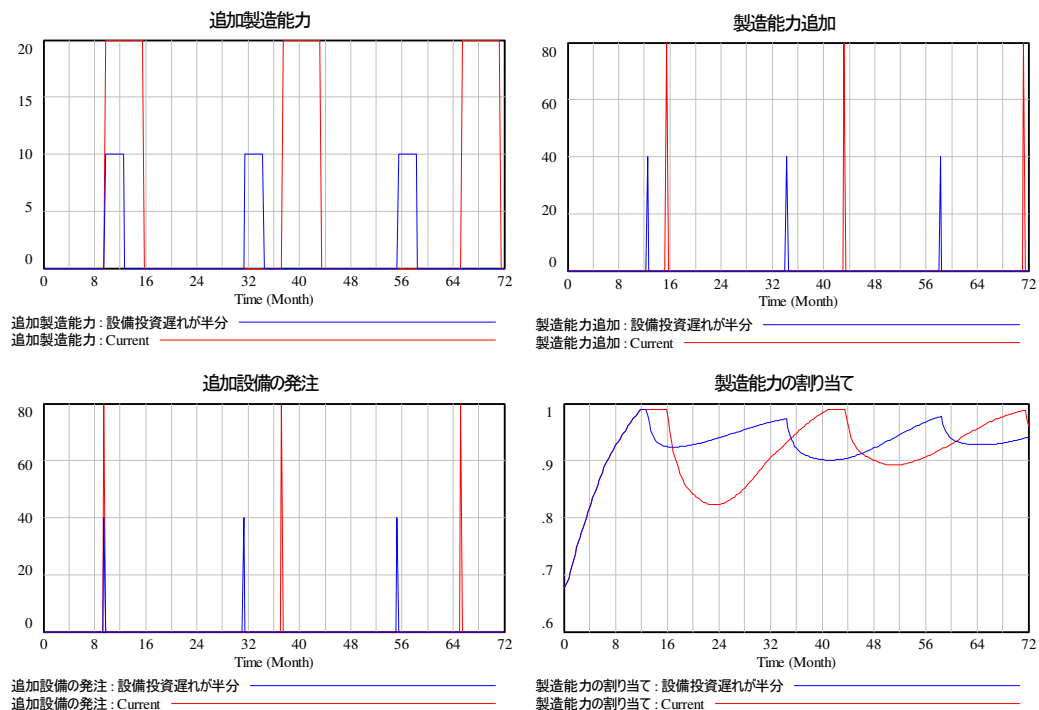


図 12-13：「設備投資量」半減、「設備投資の遅れ」を半分に短縮したシミュレーション結果

この原因として、「納期」を短くしていないからではないかという仮説が成り立ちます。納期の部分を改善してみます。「納期の限界値」を半分の 2.25 ケ月に短縮してみます。

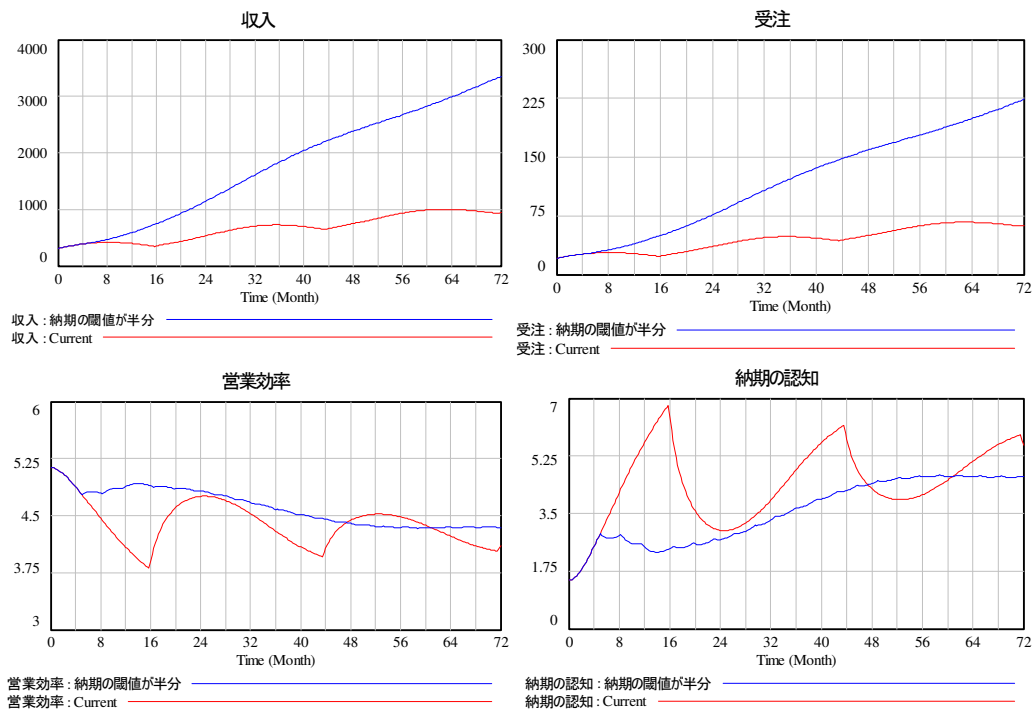


図 12-14 : 「納期の限界値」を半分に短縮したシミュレーション結果

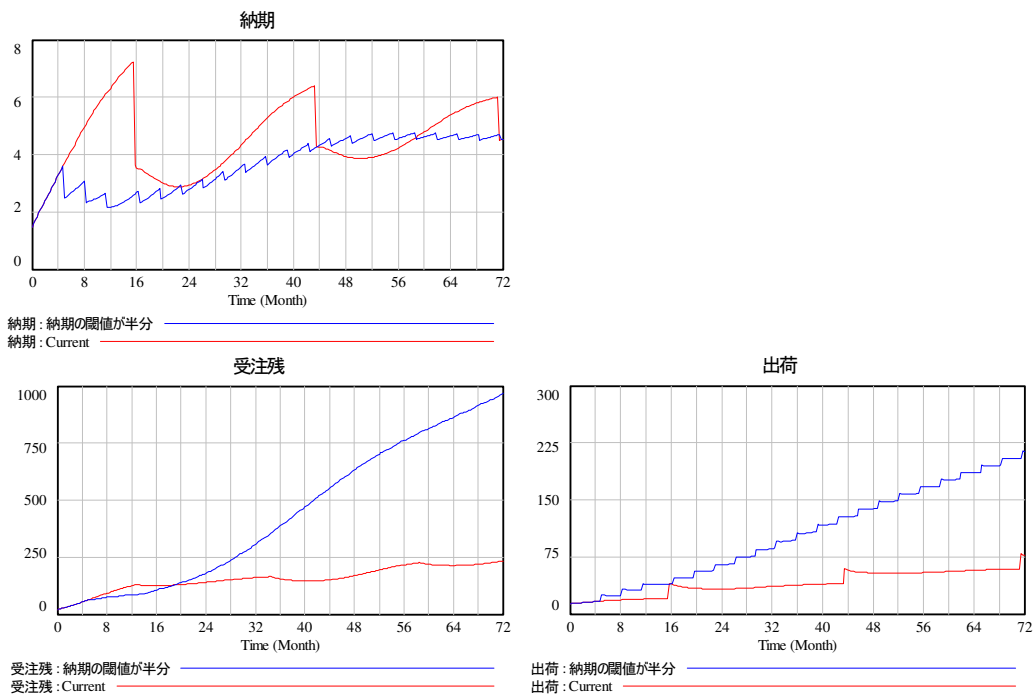


図 12-15 : 「納期の限界値」を半分に短縮したシミュレーション結果

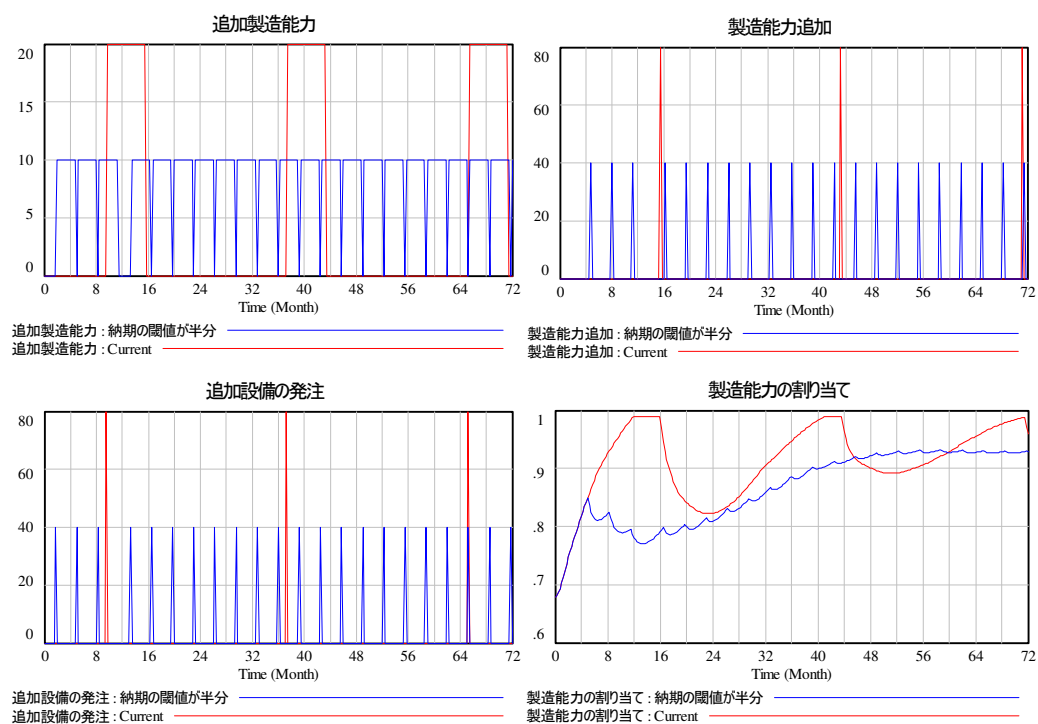


図 12-14 : 「納期の限界値」を半分に短縮したシミュレーション結果

結果は、極めて良好で、納期が大幅に改善され、収入も順調に伸びていることが分かります。念のため、納期の閾値 5.5 ケ月、4.4 ケ月、3.85 ケ月、2.25 ケ月の 4 つのケースを比べてみます。

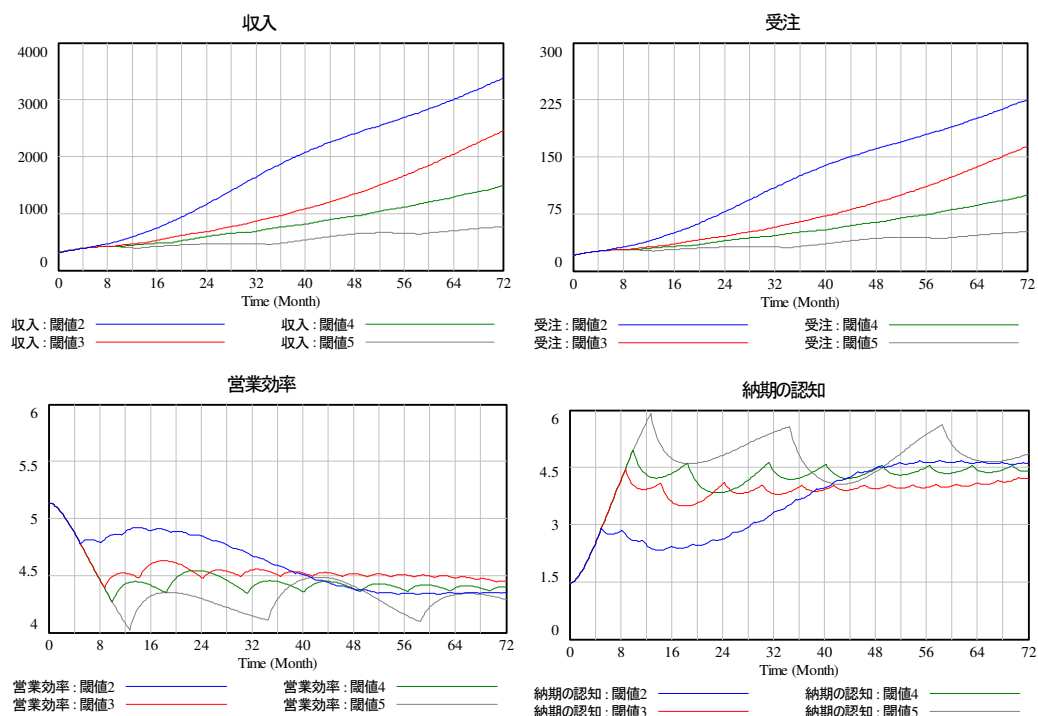


図 12-15 : 「納期の限界値」を納期閾値 5.5 ケ月、4.4 ケ月、3.85 ケ月、2.25 ケ月にしたシミュレーション結果

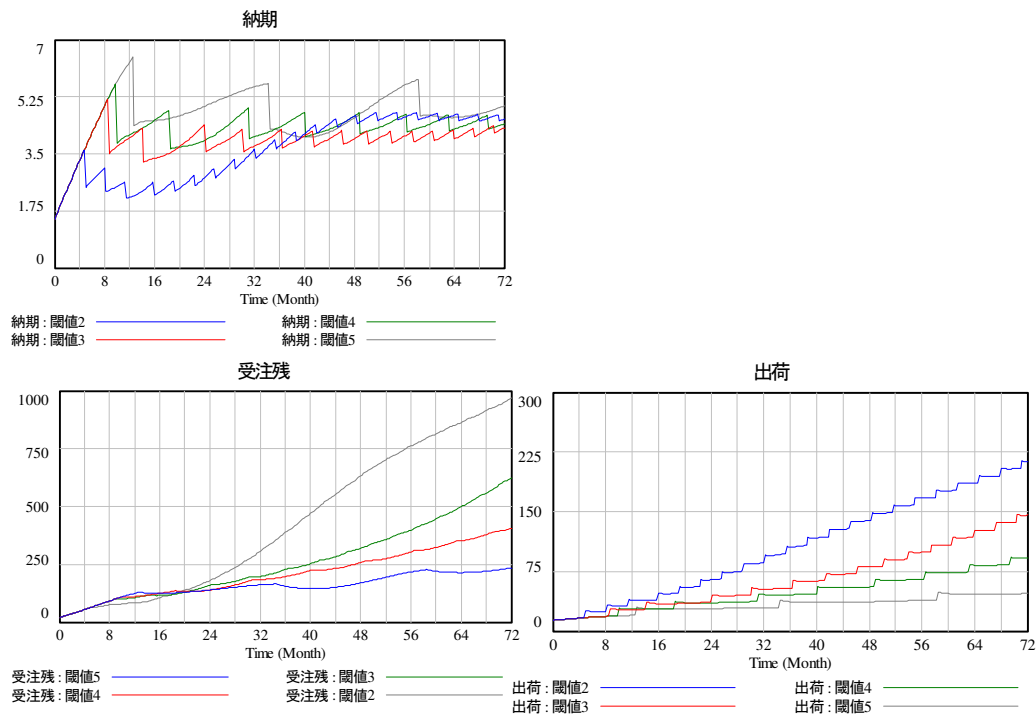


図 12-15：「納期の限界値」を納期閾値 5.5 ケ月、4.4 ケ月、3.85 ケ月、2.25 ケ月にしたシミュレーション結果

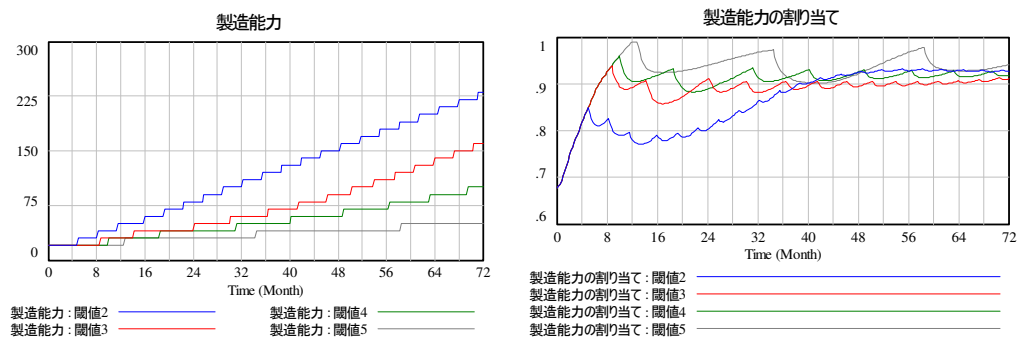


図 12-15：「納期の限界値」を納期閾値 5.5 ケ月、4.4 ケ月、3.85 ケ月、2.25 ケ月にしたシミュレーション結果

4 つのケースの中では、納期の閾値を 2.2 ケ月にしたものが一番いいことが分かります。

このモデルの最初の等式を下に示しておきます。赤が、これまでの議論で変更した部分です。最初に(14)式を 0.5 に変更しました。次に、(31)式を 3 に、(32)式を 10 に変更しました。最後の議論で、(22)式を、4.4、3.85、2.25 に変えています。

- | | |
|------|---------------------|
| (01) | FINAL TIME = 72 |
| (02) | INITIAL TIME = 0 |
| (03) | SAVEPER = TIME STEP |
| (04) | TIME STEP = 0.25 |
| (05) | 出荷=実際の製造能力 |
| (06) | 収入=受注*製品 1 個当たりの収入額 |
| (07) | 受注=営業人員*営業効率 |

(08)	受注在庫= INTEG (受注-出荷,5*営業人員の初期値)
(09)	営業マンに対する販売報酬=25
(10)	営業人員= INTEG (新規雇用と退職, 営業人員の初期値)
(11)	営業人員の初期値=4
(12)	営業効率=5-0.25*(納期の認知-2)
(13)	営業費の配賦=収入*営業費用配賦率
(14)	営業費用配賦率=0.4
(15)	実際の製造能力=製造能力*製造能力の割り当て
(16)	平滑化された納期=SMOOTH(納期,製造能力強化の遅れ,1.4)
(17)	新規雇用と退職=(望ましい営業人員数-営業人員)/新規雇用・退職の遅れ
(18)	新規雇用・退職の遅れ=3
(19)	望ましい営業人員数=営業費の配賦/営業マンに対する販売報酬
(20)	納期=受注在庫/出荷
(21)	納期の認知=SMOOTH(納期,認知遅れ)
(22)	納期の限界値=5.5
(23)	製品 1 個当たりの収入額=15
(24)	製造能力= INTEG (製造能力追加,製造能力初期値)
(25)	製造能力の割り当て=製造能力割り当て係数(平滑化された納期)
(26)	製造能力初期値=20
(27)	製造能力割り当て係数 ([(0,0)-(10,1)],(0,0),(1,0.63),(2,0.75),(3,0.83),(4,0.9),(5,0.945),(6,0.99),(7,0.99),(8,0.99),(9,0.99),(10,0.99))
(28)	製造能力強化の遅れ=1
(29)	製造能力追加=DELAY FIXED(追加設備の発注,設備投資の遅れ,0)
(30)	設備投資の必要性=IF THEN ELSE(納期>納期の限界値:AND: 追加製造能力=0,1,0)
(31)	設備投資の遅れ= 6
(32)	設備投資量=20
(33)	認知遅れ=2
(34)	追加製造能力= INTEG (追加設備の発注-製造能力追加,0)
(35)	追加設備の発注=(設備投資の必要性*設備投資量)/TIME STEP

モデルは、製造、流通、消費市場の3つのセクターから構成されていて、価格で、この3つのセクターの中の動きが決まるようになっています。

流通セクターでは、流通業者が、自社が抱える在庫を最適な量になるように価格設定して販売します。在庫が多ければ価格を下げ、在庫が少なくなれば、価格を上げます。消費者は価格に反応して購買量を変えます。安ければ多く買い、高くなれば買う量を減らします。

製造セクターでは製造者が、市場価格の変動を見ながら、製造能力を考え、設備投資を行い、製造します。製造セクターには、3ヶ所で遅れがあります。まず、製造で12ヶ月の遅れがあります。また、価格変動を見ながら、期待価格を決める遅れ(5ヶ月)、それに基づき設備投資を行うのですが、実際に設備投資が行われるまでに遅れ(4ヶ月)があります。

価格は、卸が卸の在庫量を見て決めます。ここは、表になっていて、表 13-1 上段のような非線型です。同じく、一人当たりの購買量も、購買者が、価格を見て購買量を決めます。ここも表 13-1 中段のような非線型になっています。

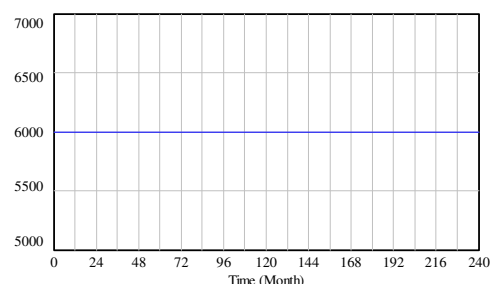
設備投資も、期待価格(一種の価格予想。平滑化して決める)を見て、製造者が、設備投資量を決めます。ここも表 13-1 下段のような非線型になっています。

製造された製品が「生産・入庫」に入力になりますが、ここは指数遅れにしています。

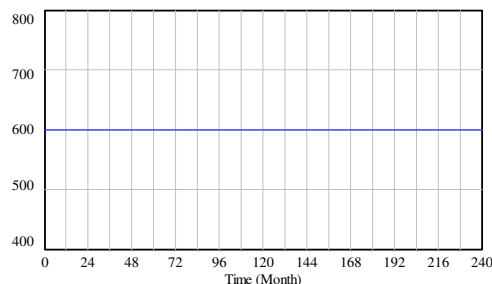
「製造能力」と「卸の在庫」を中心に変動を見ることにします。基本的には均衡モデルなので、変化はないはずなので、その他のパラメーターは見ないことにします。図 6-2 に示していますが、確かに均衡していて、変化が見られません。

表 13-1：在庫と価格、購買量、製造能力

流通業者の価格設定						
卸の在庫	0	2,000	4,000	6,000	8,000	10,000
価格	100	94	80	50	20	10
購買者の購買量						
価格	0	20	40	60	80	100
一人当たりの購買量	7	6	4	2	1	0
製造業者の設備投資						
価格	0	20	40	60	80	100
望ましい製造能力	0	40	200	1,000	1,200	1,280



卸の在庫 : Current



製造能力 : Current

図 13-2：シミュレーション結果

(2) 製品市場モデル：購買変動を加える

このモデルに、上下 25%の購買変動を加えてみます。変動はあるものの、基本的には変

わらない均衡状態であることが分かります。

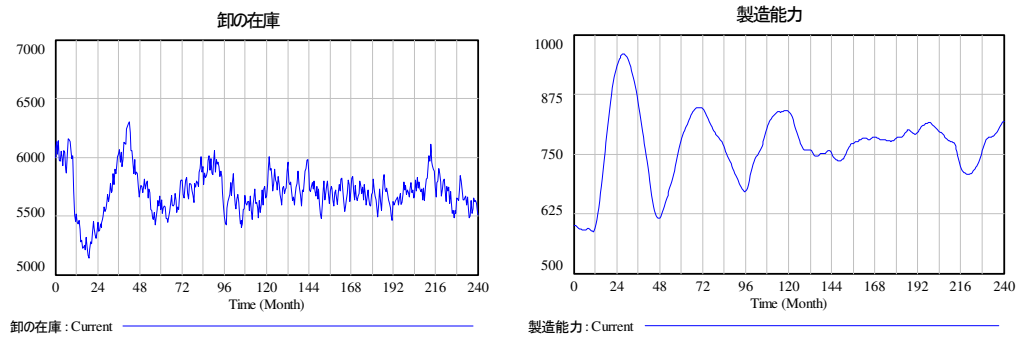


図 13-3：購買変動を加えたモデルでのシミュレーション

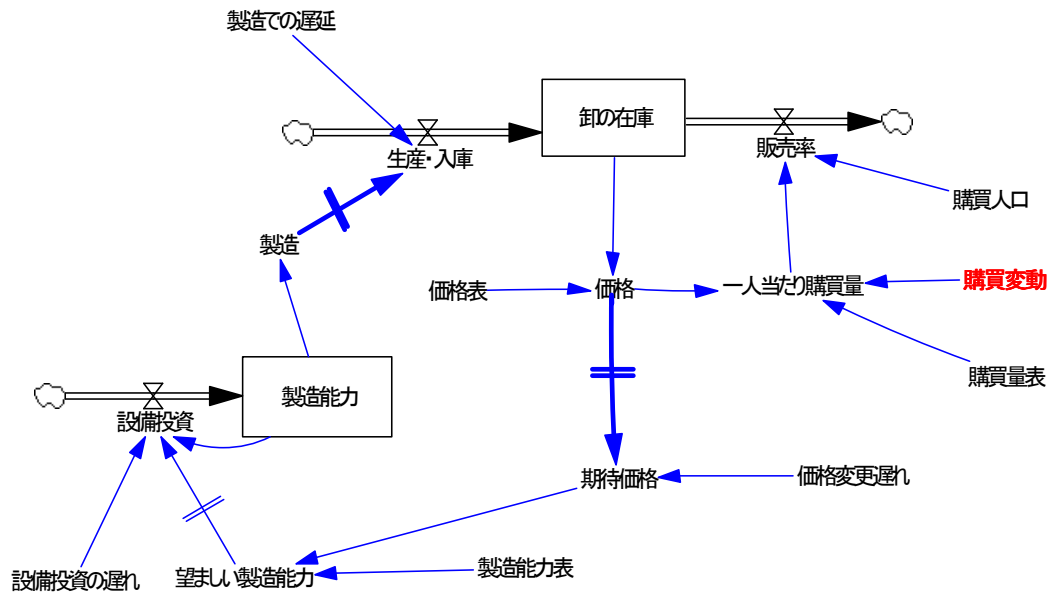


図 13-4：購買変動を加えたモデル

(3) 製品市場モデル：購買者人口増を加える

購買者人口が月5%で増加するような変化を加えてみます。卸の在庫が減っていく一方で、製造能力が増えていくことが分かります。

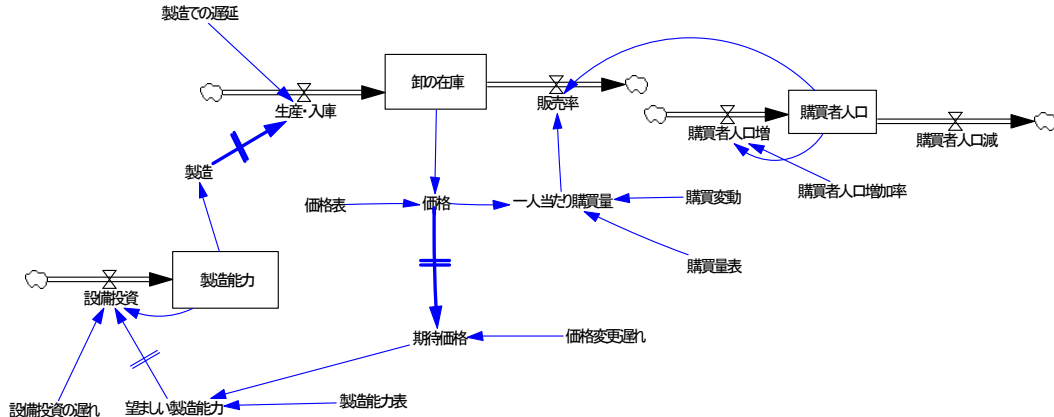


図 13-5：購買者人口増を加えたモデル

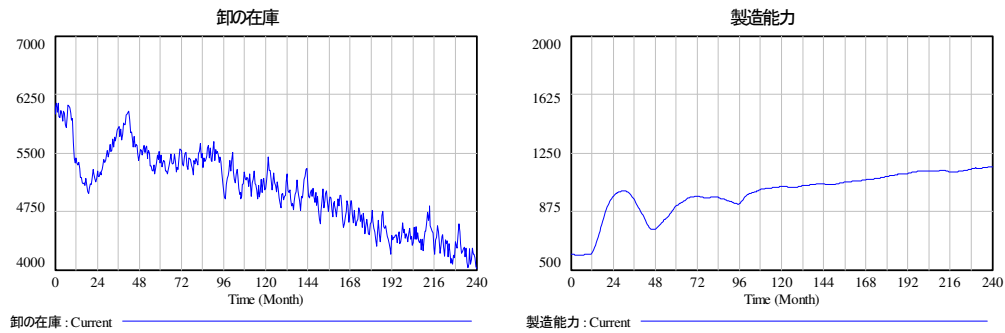


図 13-6：購買者人口増を加えたモデルのシミュレーション結果

(4) 製品市場モデル：購買者人口は変えなく製造能力減耗を加える

購買者人口は変えなく製造能力減耗を加えてみましょう。製造能力が 10 年（120 ケ月）で減耗するようにします。製造能力の減耗は、製造能力の変化には影響を及ぼすものの、卸の在庫にはあまり大きな影響を与えないことが分かります。

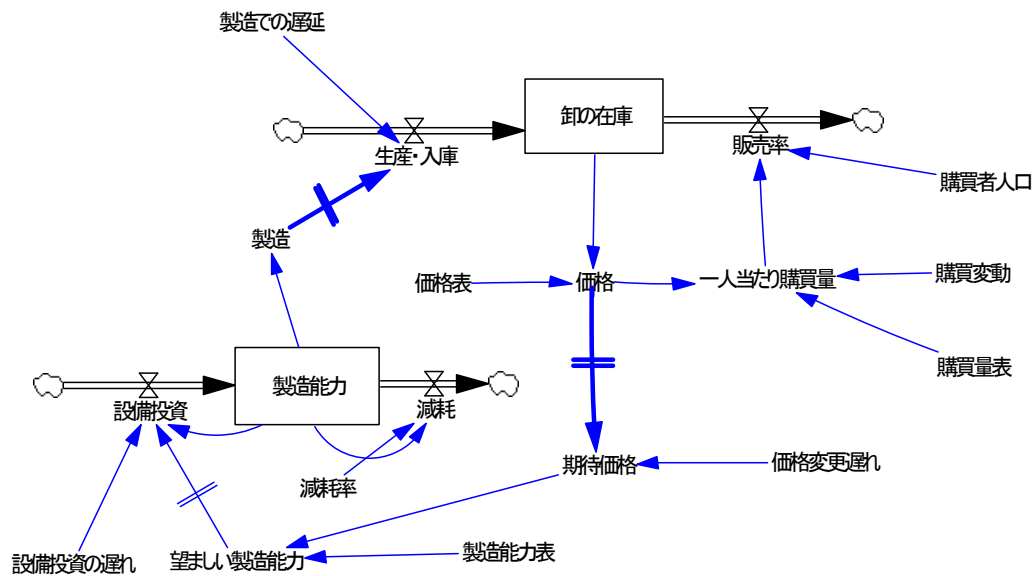


図 13-7：製造能力減耗を加えたモデル

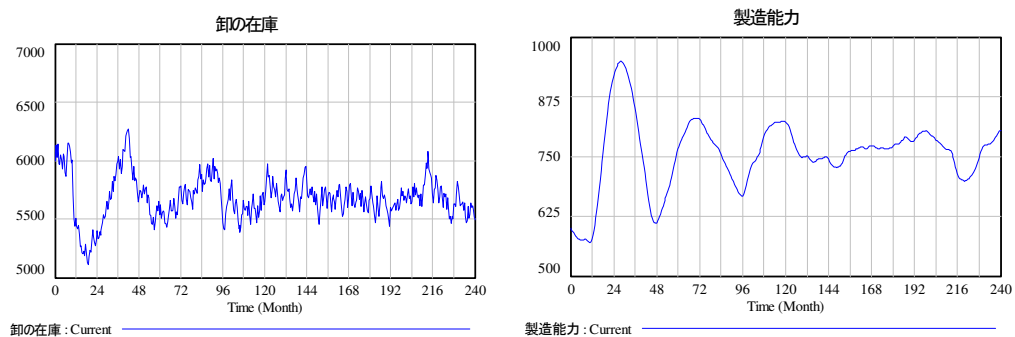


図 13-8：製造能力減耗を加えたモデルのシミュレーション結果

(5) 製品市場モデル：リサイクル製品の流通を加える

商品が売れるようになると、リサイクル業者が現れ、廃品となった商品を回収し、そのスクラップの中でまだ使える部品を集め、製品に仕立て、市場で販売することを考えるようになります。これがリサイクル商品と呼ばれているものです。

通常は、新規に部品を製造する必要がなく、単純に組み立て加工を行うだけなので、製造コストが安くつき、リサイクル製品は新品に比べ、売値も格段に安くなります。それでも、機能的には通常の製品と変わりません。

もっとも、再生紙など、商品によっては、新規に製品を製造した方が安い場合もあります。また、先に述べたコモディティ商品的な性格ではない製品では、供給によって制限されたり、パテントなどで権利が守られているので、リサイクル製品を製造できないとか、ブランドを守るためのメーカーの意向で禁止されているものもあります。

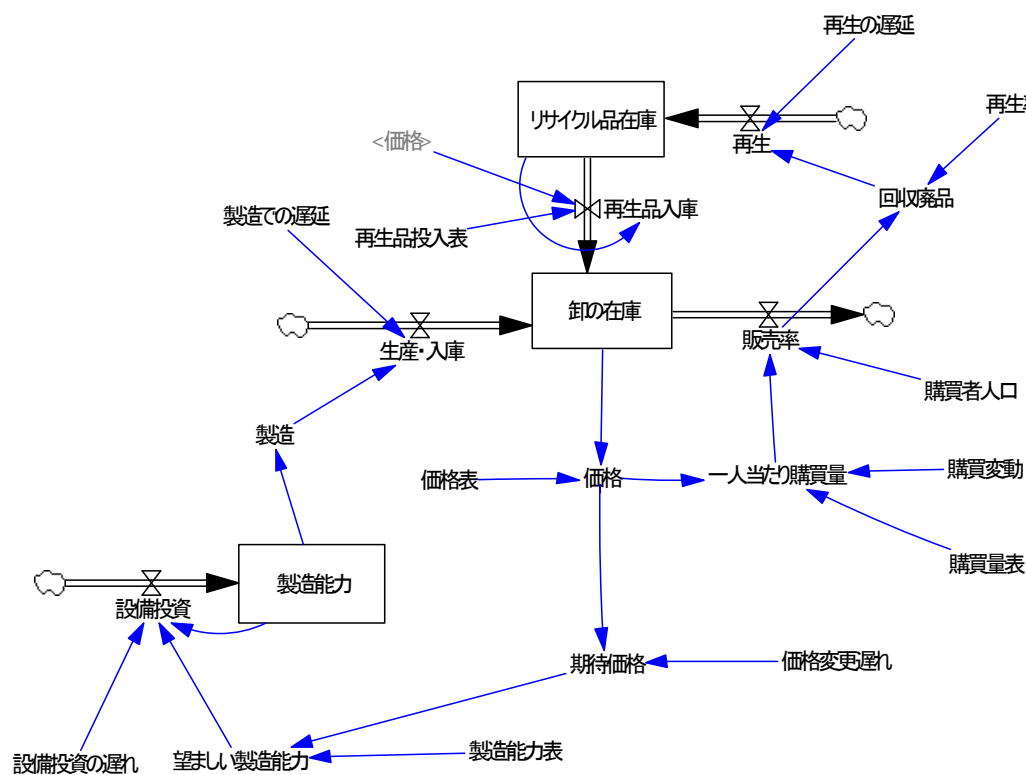


図 13-9：リサイクル製品市場を加えたモデル

表 13-2：リサイクル製品と価格

価格とリサイクル製品の市場投入率					
価格	0	25	50	75	100
リサイクル製品投入率	0	0	0.1	0.2	0.4

ここでは、価格が 100 であれば、40%の製品がリサイクルされ、再生品となって市場流通するとします。表 13-2 に示したような非線形で、市場価格によって投入率が決まるとします。一度使われた製品を集めて、スクラップから製品を再生するので、リサイクル製品からの製品再生には遅れが 20 年あるとします。

シミュレーションしてみると、リサイクル製品の登場で、卸の在庫が増え、価格が下が

り、製造能力の増加が減っていくことが分かります。丁度、リサイクル製品の増加分だけ、製造能力が下がっていくように見えます。

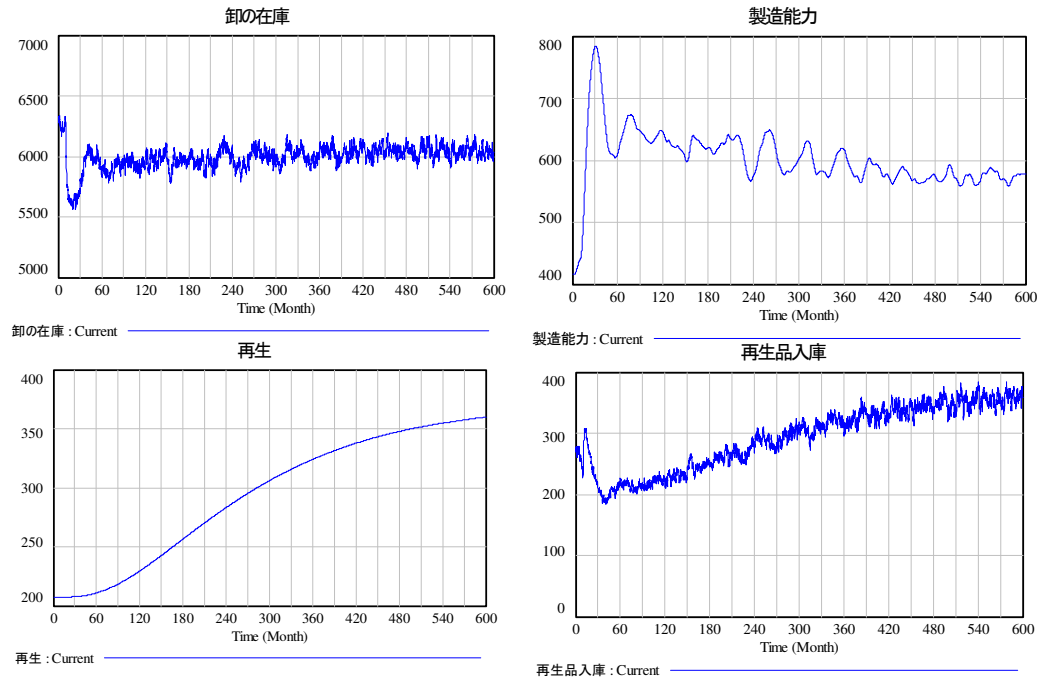


図 13-10：リサイクル製品市場を加えたモデルでのシミュレーション結果

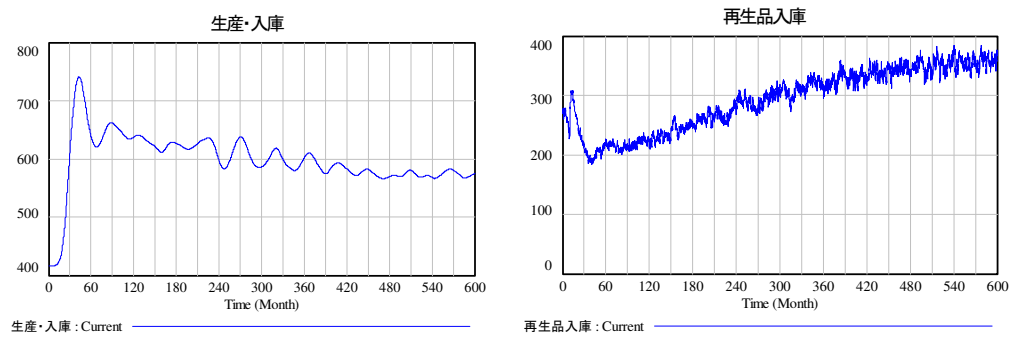


図 13-11：リサイクル製品市場を加えたモデルでのシミュレーション結果

2 番目の購買変動を加えたモデルの等式は、以下のとおりです。最初の購買変動が無いモデルは、式(21)が無く、(05)式が：
 一人当たり購買量=購買量表(価格)
 となります。

(01)	FINAL TIME = 240
(02)	INITIAL TIME = 0
(03)	SAVEPER =TIME STEP
(04)	TIME STEP = 0.5
(05)	一人当たり購買量=購買量表(価格)*(1+購買変動)
(06)	価格=価格表(卸の在庫)

(07)	価格変更遅れ=5
(08)	価格表([(0,0)-(10000,100)],(0,100),(2000,94),(4000,80),(6000,50),(8000,20),(10000,10))
(09)	卸の在庫= INTEG (生産・入庫-販売率,6000)
(10)	望ましい製造能力=製造能力表(期待価格)
(11)	期待価格=SMOOTH(価格,価格変更遅れ)
(12)	生産・入庫=DELAY3(製造,製造での遅延)
(13)	製造=製造能力
(14)	製造での遅延=12
(15)	製造能力= INTEG (設備投資,600)
(16)	製造能力表([(0,0)-(100,2000)],(0,0),(20,40),(40,200),(60,1000),(80,1200),(100,1280))
(17)	設備投資=(望ましい製造能力-製造能力)/設備投資の遅れ
(18)	設備投資の遅れ=4
(19)	販売率=一人当たり購買量*購買人口
(20)	購買人口=200
(21)	購買変動=RANDOM UNIFORM(-0.5,0.5,2374)+STEP(0.5,10)
(22)	購買量表([(0,0)-(100,10)],(0,7),(20,6),(40,4),(60,2),(80,1),(100,0))

3 番目の購買者人口増のモデルは、以下のようになります。このモデルは、さらに発達させて、購買者人口が、広告などによって増加したり減少したりするように改良してみると面白いと思います。そのため、本来は不要である、「購買者人口減」というフローも加えています。

(01)	FINAL TIME = 240
(02)	INITIAL TIME = 0
(03)	SAVEPER = TIME STEP
(04)	TIME STEP = 0.5
(05)	一人当たり購買量=購買量表(価格)*(1+購買変動)
(06)	価格=価格表(卸の在庫)
(07)	価格変更遅れ=5
(08)	価格表([(0,0)-(10000,100)],(0,100),(2000,94),(4000,80),(6000,50),(8000,20),(10000,10))
(09)	卸の在庫= INTEG (生産・入庫-販売率,6000)
(10)	望ましい製造能力=製造能力表(期待価格)
(11)	期待価格=SMOOTH(価格,価格変更遅れ)
(12)	生産・入庫=DELAY3(製造,製造での遅延)
(13)	製造=製造能力
(14)	製造での遅延=12
(15)	製造能力= INTEG (設備投資,600)
(16)	製造能力表([(0,0)-(100,2000)],(0,0),(20,40),(40,200),(60,1000),(80,1200),(100,1280))
(17)	設備投資=(望ましい製造能力-製造能力)/設備投資の遅れ
(18)	設備投資の遅れ=4
(19)	販売率=一人当たり購買量*購買人口
(20)	購買人口=200
(21)	購買変動=RANDOM UNIFORM(-0.5,0.5,2374)+STEP(0.5,10)
(22)	購買量表([(0,0)-(100,10)],(0,7),(20,6),(40,4),(60,2),(80,1),(100,0))
(23)	購買者人口増加率=0.005
(24)	購買者人口減=0
(25)	購買量表([(0,0)-(100,10)],(0,7),(20,6),(40,4),(60,2),(80,1),(100,0))

4 番目の製造能力減耗を加えたモデルは、先の 2 番目のモデルに

減耗=製造能力/減耗率

減耗率=120

を加えただけです。ただし、ストックである、「製造能力」は、「減耗」分がストックから出ていくので、式が下のように変わります。

製造能力= INTEG (設備投資-減耗,600)

Vensim PLE では、2 番目のモデルを使って、先のフローと変数を追加します。その後、「製造能力」というストックの定義画面を開くと、ソフトがこのように自動定義してくれているので、そのまま承認の意味で OK ボタンを押せばいいでしょう。

最後のリサイクル製品市場を加えたモデルの等式は以下のようになります。基本的には、2 番目のモデルにリサイクル商品セクターを加えただけでのものです。(05)、(10)から(14)、(16)式が新しく加わったものです。(15)式は、再生品が加わり少し変わります。なお、再生品を考慮し、卸の在庫が 6,000 個から 6,307 個に変わります。また、設備投資の初期値も、600 から 415.39 に下がります。

(01)	FINAL TIME = 1200
(02)	INITIAL TIME = 0
(03)	SAVEPER = TIME STEP
(04)	TIME STEP = 0.125
(05)	リサイクル品在庫= INTEG (再生-再生品入庫,3396)
(06)	一人当たり購買量=購買量表(価格)*(1+購買変動)
(07)	価格=価格表(卸の在庫)
(08)	価格変更遅れ=5
(09)	価格表([(0,0)-(10000,100)],(0,100),(2000,94),(4000,80),(6000,50),(8000,20),(10000,10))
(10)	再生=DELAY3(回収廃品,再生の遅延)
(11)	再生の遅延=240
(12)	再生品入庫=再生品投入表(価格)*リサイクル品在庫
(13)	再生品投入表([(0,0)-(100,0.5)],(0,0),(25,0),(50,0.1),(75,0.2),(100,0.4))
(14)	再生率=0.4
(15)	卸の在庫= INTEG (再生品入庫+生産・入庫-販売率,6307)
(16)	回収廃品=販売率*再生率
(17)	望ましい製造能力=製造能力表(期待価格)
(18)	期待価格=SMOOTH(価格,価格変更遅れ)
(19)	生産・入庫=DELAY3(製造,製造での遅延)
(20)	製造=製造能力
(21)	製造での遅延=12
(22)	製造能力= INTEG (設備投資,415.39)
(23)	製造能力表([(0,0)-(100,2000)],(0,0),(20,40),(40,200),(60,1000),(80,1200),(100,1280))
(24)	設備投資=(望ましい製造能力-製造能力)/設備投資の遅れ
(25)	設備投資の遅れ=4
(26)	販売率=一人当たり購買量*購買者人口
(27)	購買変動=RANDOM UNIFORM(-0.5,0.5,2374)+STEP(0.5,10)
(28)	購買者人口=200
(29)	購買量表([(0,0)-(100,10)],(0,7),(20,6),(40,4),(60,2),(80,1),(100,0))

14. ビール・ゲーム

ビール・ゲームは、小売店、二次卸、一次卸、工場を担当する 4 人がチームを組み、経営コストを最小限にすることを競うものです。初期条件では、小売店、二次卸、一次卸、工場が 12 個のストックを持ちます。また、それぞれの間に、4 個の配送中のストックが 2 個ずつ挟まれ、商品は 2 週間の遅れで工場、一次卸、二次卸、小売店に配送されます。

それぞれに、2 ケ所の注文遅れが挟まれ、小売店、二次卸、一次卸、工場に 2 週間遅れで注文が配達され、さらに、工場からの製造指示と工場倉庫までの製品搬入に 2 週間の製造遅れが存在します。

在庫数に対し、0.5\$ の費用、受注残に対し \$1.0 の費用が発生します。この費用発生を最少にするように発注します。

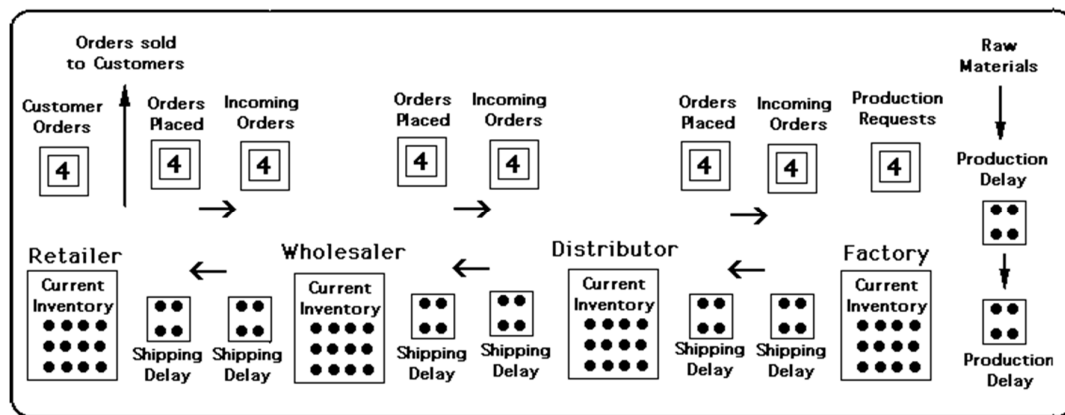


図 14-1：ビール・ゲームの配盤

ビール・ゲームの配置

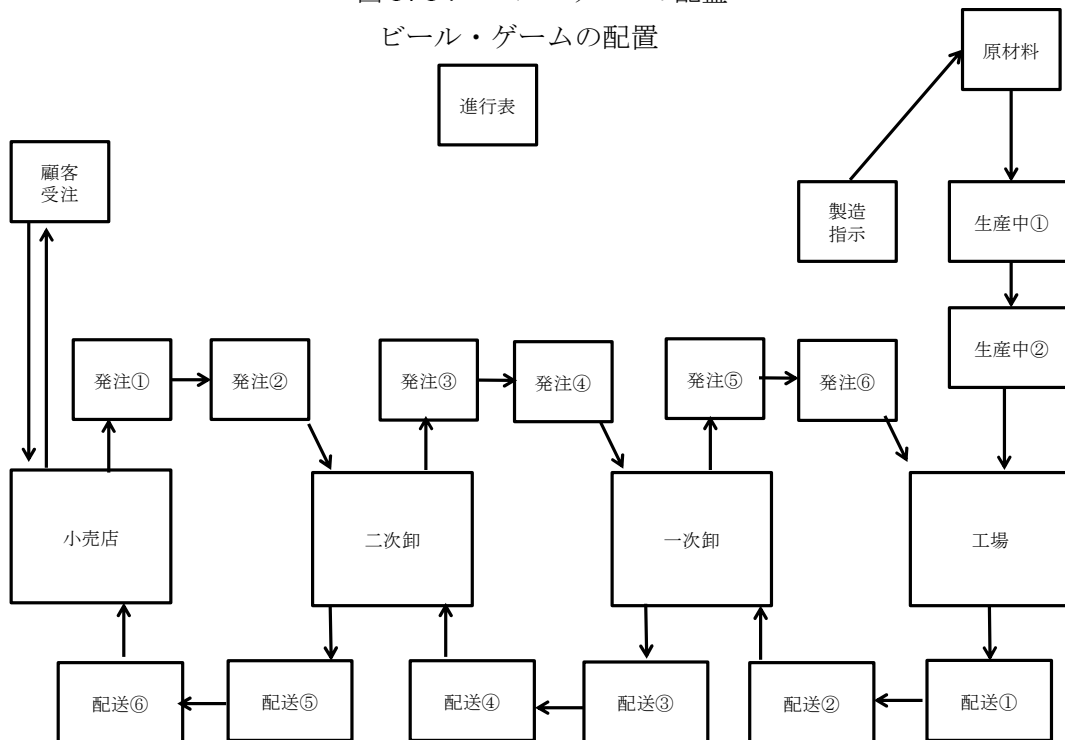


図 14-2：ビール・ゲームの配盤

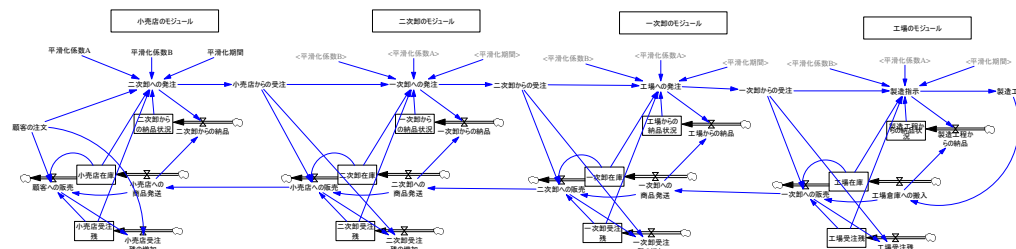


図 14-3：ビール・ゲームのモデル

すでに演習で説明していますが、前にモデル深化で取り上げた、小売店と工場のモデルで、小売店と工場の中に、二次卸、一次卸を挿入しただけで、このビール・ゲームのモデルになります。

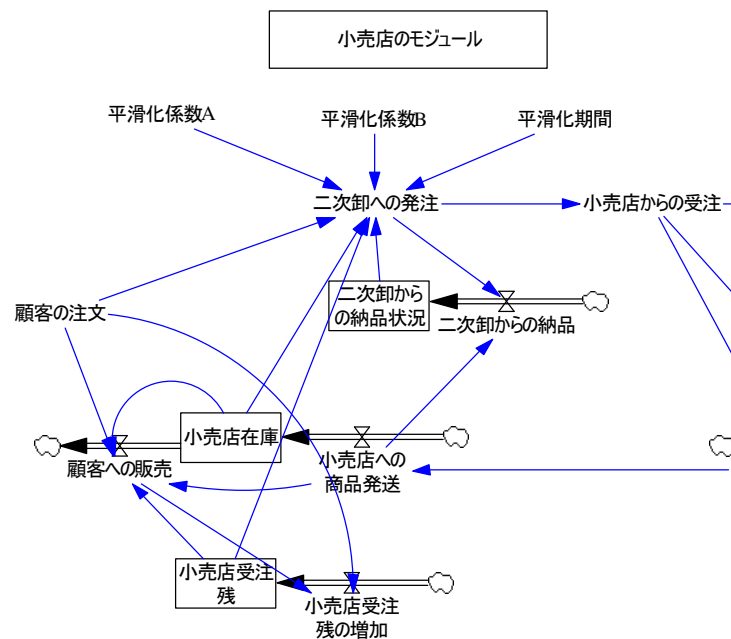


図 14-4：小売店のモジュール

小売店、二次卸、一次卸、工場共に、基本的には同じ構造、同じ式ですので、小売店のモジュールを使って説明します。

前のモデル深化で取り上げた小売店のモジュールは、顧客の中も、在庫と発注、工場からの納品だけでした。ここでは、在庫を在庫と受注残に分けています。また、工場からの納品は、小売店在庫に入るのですが、定常発注分を計算するために、「二次卸からの納品状況」というストックを用意しています。

小売店の発注は、

- 1) 顧客に販売した分の補充分：平滑化する
- 2) 在庫調整：標準在庫は 12 なので、それを保持するように発注する
- 3) 定常発注：パイプライン発注分で、定常状態、あるいは均衡状態を保つための発注。ここは、「二次卸からの納品状況」というストックを使って平滑化

の3種類の発注要素があることを説明しました。

上の1)、2)、3)をそのまま調整しないで発注すると過剰発注になるので、係数A、係数Bを使って調整します。

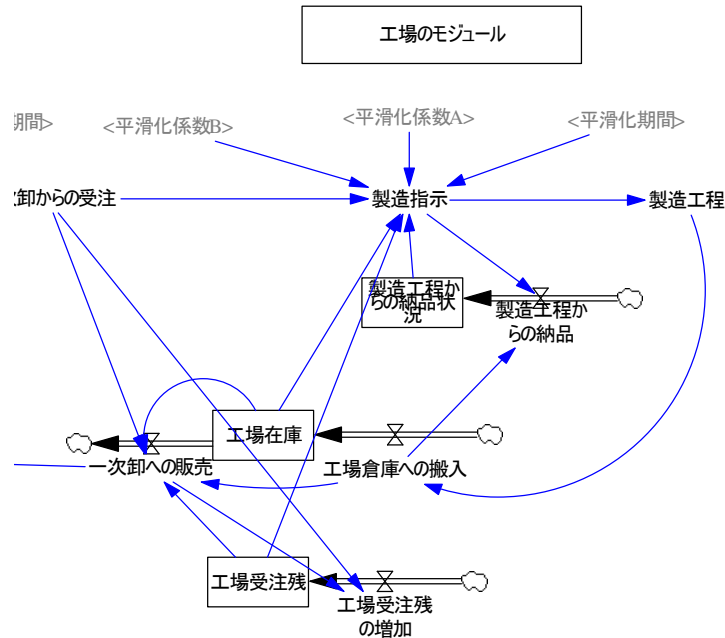


図 14-5：工場のモジュール

ただ、工場のモジュールだけは、製造工程から工場在庫に入荷するようになっています。

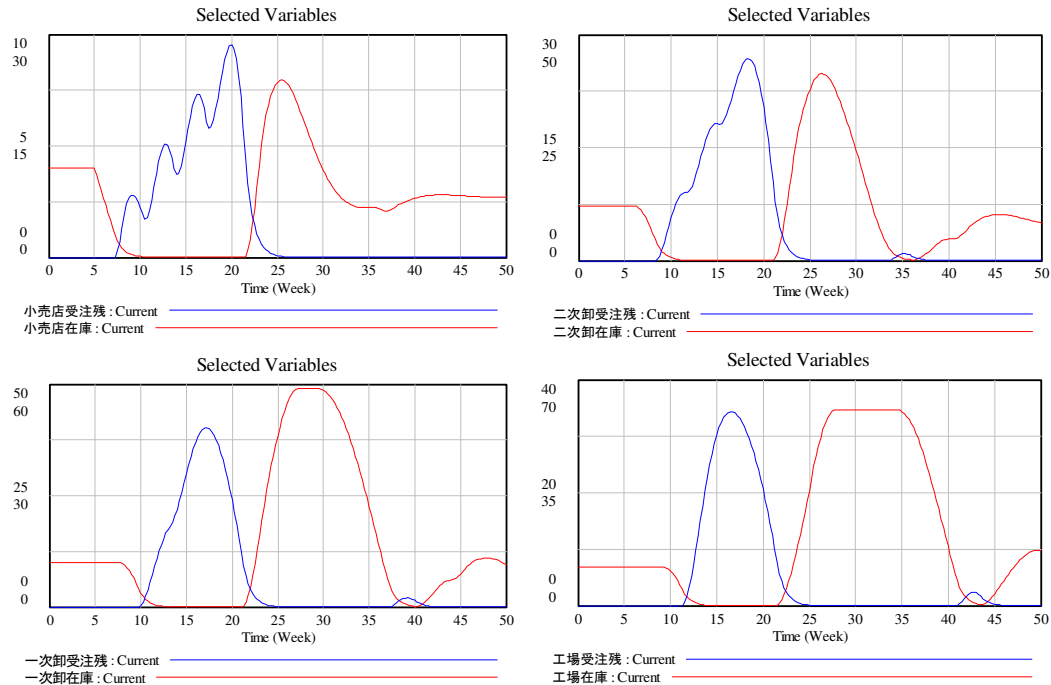


図 14-6：シミュレーション結果：在庫と受注残

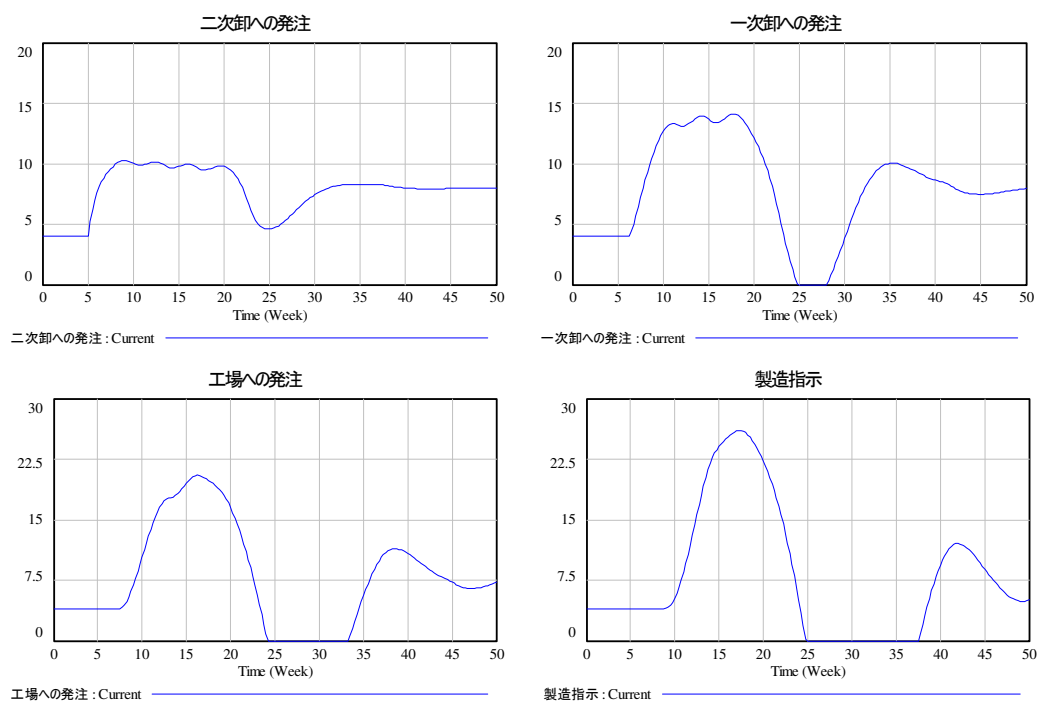


図 14-7：シミュレーション結果：発注

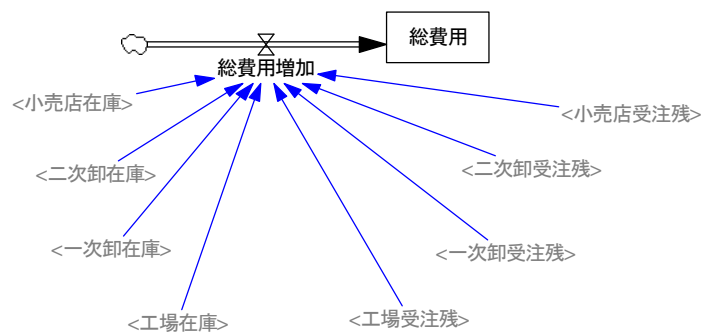


図 14-8：総費用計算モジュール

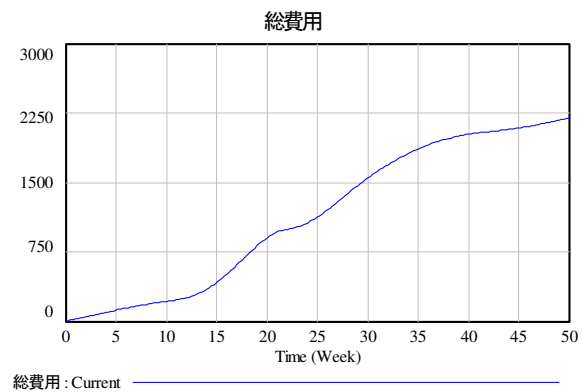


図 14-8：総費用

このモデルでのシミュレーションの総費用は\$2,250 ぐらいで、これは実際のビール・ゲームでの平均点に近いそうです。ただ、Sternan によれば、製造モジュールはもう少し改善する必要があるとのことでした。

実際のビール・ゲームでも、顧客は小売店に毎週 4 ケースのビールを注文していて、8 週目から 2 倍の 8 ケースに発注量を変えます。その他には顧客は注文を変えていないのですが、小売店の思惑で発注した量に一次卸の思惑、二次卸の思惑が入って、工場に届くころには注文数が 8 ケースではなく、とんでもない数量に跳ね上がっています。

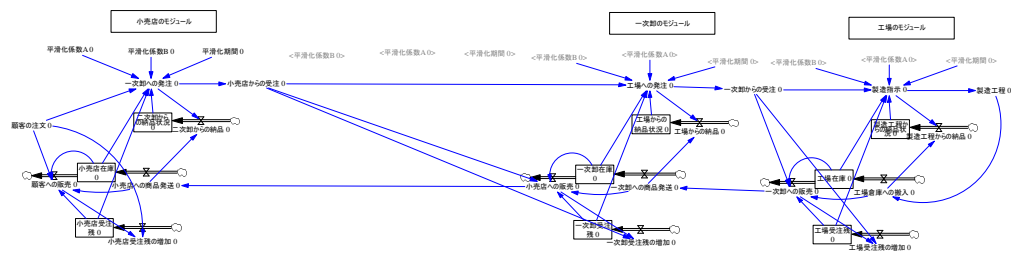
発注しても商品には、工場での製造で 2 週間、工場から一次卸の間で 2 週間、一次卸から二次卸の間で 2 週間、そして二次卸から小売店までの間で 2 週間の配送遅れがあるので、それぞれ、注文してもなかなか納品されなく、その間、顧客の注文が続くので、在庫がどんどん減っていきます。さらには在庫がゼロになり、受注残になってしまいます。こうなると焦って、発注数を多くし、ますます発注数変動が大きくなっていきます。

問題は、遅れてどんどん商品が入荷するようになってからです。入荷する数量の方が顧客の発注よりも多いので、今度はどんどん在庫が溜まっていきます。あわてて注文数をゼロにしても、それでも商品の納入が続きます。

この様子が図 14-6、14-7 に示されています。なお、小売店の発注変動が、二次卸、一次卸、工場に伝わっていく際に増幅されていく様子が見られます。また、在庫や受注残では、小売店の在庫変動が、二次卸、一次卸、工場では滑らかになっていることが分かります。

こういった状況を改善するにはどうすればいいのでしょうか？ 小売店、二次卸、一次卸の発注数や工場の製造指示での決定のルールを変えるという手もあるでしょうが、システムになっているので、そう簡単ではありません。システムを変えるという見地からは、二次卸を抜いて、小売店は一次卸に発注する、さらには、工場に直接発注するというやり方があるでしょう。その結果を確かめてみましょう。

二次卸を取り除いたモデル



二次卸と一次卸を取り除いたモデル

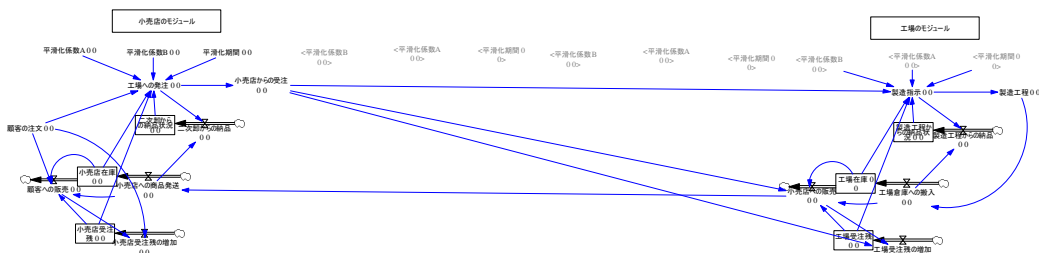


図 14-9：二次卸と一次卸を取り除いたモデル

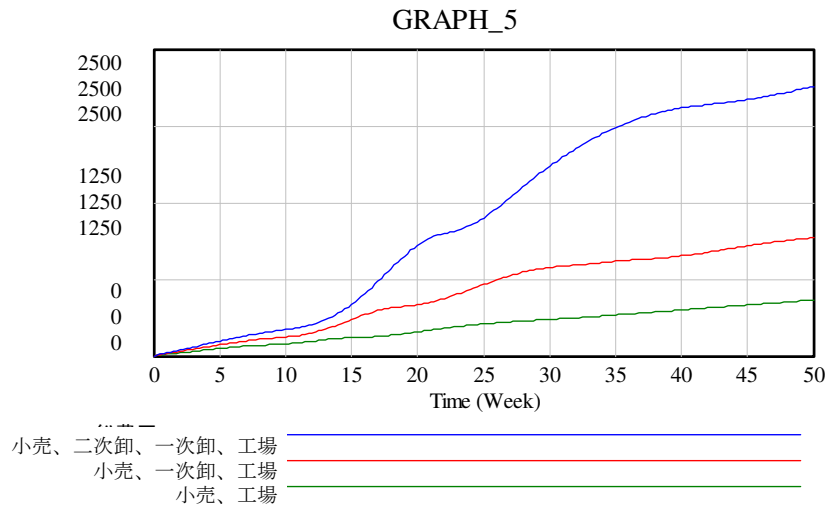


図 14-10：二次卸と一次卸を取り除いたモデルでのシミュレーション結果

図 14-10 に、その比較を行っています、確かに、工場に直接発注するというやり方が一番合理的であることが分かります。

なお、この最初のビール・ゲームのモデルの等式は以下の通りです。

- | | |
|------|--|
| (01) | FINAL TIME = 50 |
| (02) | INITIAL TIME = 0 |
| (03) | SAVEPER = TIME STEP |
| (04) | TIME STEP = 0.25 |
| (05) | 一次卸からの受注=DELAY FIXED(工場への発注,1,4) |
| (06) | 一次卸からの納品=二次卸への商品発送-一次卸への発注 |
| (07) | 一次卸からの納品状況= INTEG (一次卸からの納品,0) |
| (08) | 一次卸への商品発送=DELAY FIXED(一次卸への販売,2,4) |
| (09) | 一次卸への発注=MAX(0,SMOOTH(小売店からの受注,平滑化期間)+平滑化係数 A*(12-(二次卸在庫-二次卸受注残)+平滑化係数 B*一次卸からの納品状況)) |
| (10) | 一次卸への販売=MIN(工場在庫+工場倉庫への搬入,一次卸からの受注+工場受注残) |
| (11) | 一次卸受注残= INTEG (一次卸受注残の増加,0) |
| (12) | 一次卸受注残の増加=二次卸からの受注-二次卸への販売 |
| (13) | 一次卸在庫= INTEG (一次卸への商品発送-二次卸への販売,12) |
| (14) | 二次卸からの受注=DELAY FIXED(一次卸への発注,1,4) |
| (15) | 二次卸からの納品=小売店への商品発送-二次卸への発注 |
| (16) | 二次卸からの納品状況= INTEG (二次卸からの納品,0) |
| (17) | 二次卸への商品発送=DELAY FIXED(二次卸への販売,2,4) |
| (18) | 二次卸への発注=MAX(0,SMOOTH(顧客の注文,平滑化期間)+平滑化係数 A*(12-(小売店在庫-小売店受注残)+平滑化係数 B*二次卸からの納品状況)) |
| (19) | 二次卸への販売=MIN(一次卸在庫+一次卸への商品発送,二次卸からの受注+一次卸受注残) |
| (20) | 二次卸受注残= INTEG (二次卸受注残の増加,0) |
| (21) | 二次卸受注残の増加=小売店からの受注-小売店への販売 |

(22)	二次卸在庫= INTEG (二次卸への商品発送-小売店への販売,12)
(23)	小売店からの受注=DELAY FIXED(二次卸への発注,1,4)
(24)	小売店への商品発送=DELAY FIXED(小売店への販売,2,4)
(25)	小売店への販売=MIN(二次卸在庫+二次卸への商品発送,小売店からの受注+二次卸受注残)
(26)	小売店受注残= INTEG (小売店受注残の増加,0)
(27)	小売店受注残の増加=顧客の注文-顧客への販売
(28)	小売店在庫= INTEG (小売店への商品発送-顧客への販売,12)
(29)	工場からの納品=一次卸への商品発送-工場への発注
(30)	工場からの納品状況= INTEG (工場からの納品,0)
(31)	工場への発注=MAX(0,SMOOTH(二次卸からの受注,平滑化期間)+平滑化係数 A*(12-(一次卸在庫-一次卸受注残)+平滑化係数 B*工場からの納品状況))
(32)	工場倉庫への搬入=DELAY FIXED(製造工程,2,4)
(33)	工場受注残= INTEG (工場受注残の増加,0)
(34)	工場受注残の増加=一次卸からの受注-一次卸への販売
(35)	工場在庫= INTEG (工場倉庫への搬入-一次卸への販売,12)
(36)	平滑化係数 A=0.25
(37)	平滑化係数 B=0.33
(38)	平滑化期間=1
(39)	総費用= INTEG (総費用増加,0)
(40)	総費用増加=(小売店在庫+二次卸在庫+一次卸在庫+工場在庫)*0.5+(小売店受注残+二次卸受注残+一次卸受注残+工場受注残)
(41)	製造工程=DELAY FIXED(製造指示,1,4)
(42)	製造工程からの納品=工場倉庫への搬入-製造指示
(43)	製造工程からの納品状況= INTEG (製造工程からの納品,0)
(44)	製造指示=MAX(0,SMOOTH(一次卸からの受注,平滑化期間)+平滑化係数 A*(12-(工場在庫-工場受注残)+平滑化係数 B*製造工程からの納品状況))
(45)	顧客の注文=4+STEP(4,5)
(46)	顧客への販売=MIN(小売店在庫+小売店への商品発送,顧客の注文+小売店受注残)

(以上)