平成２１年度（２００９年度）

信州大学大学院工学系研究科修士学位論文

専攻名　情　報　　工学専攻

学籍番号　０４　ＴＡ　５７２Ｃ

氏　　名　鳥塚達夫

目次

【0．言葉の定義】

1．図－1に示すブロック線図を例に、基本となる概念について名称を定義する。

(1) G1～G3については、一般の教科書などで「前向き伝達要素」などと呼ばれているが、ここでは **主軸要素** と呼ぶ。あわせて、主軸要素が配置される軸線を **主軸** と呼ぶ。

(2) H1、H2については一般の教科書などと同じく **フィードバック要素** と呼ぶ。

(3) フィードバック要素が形成するループを **フィードバックループ** と呼ぶ。

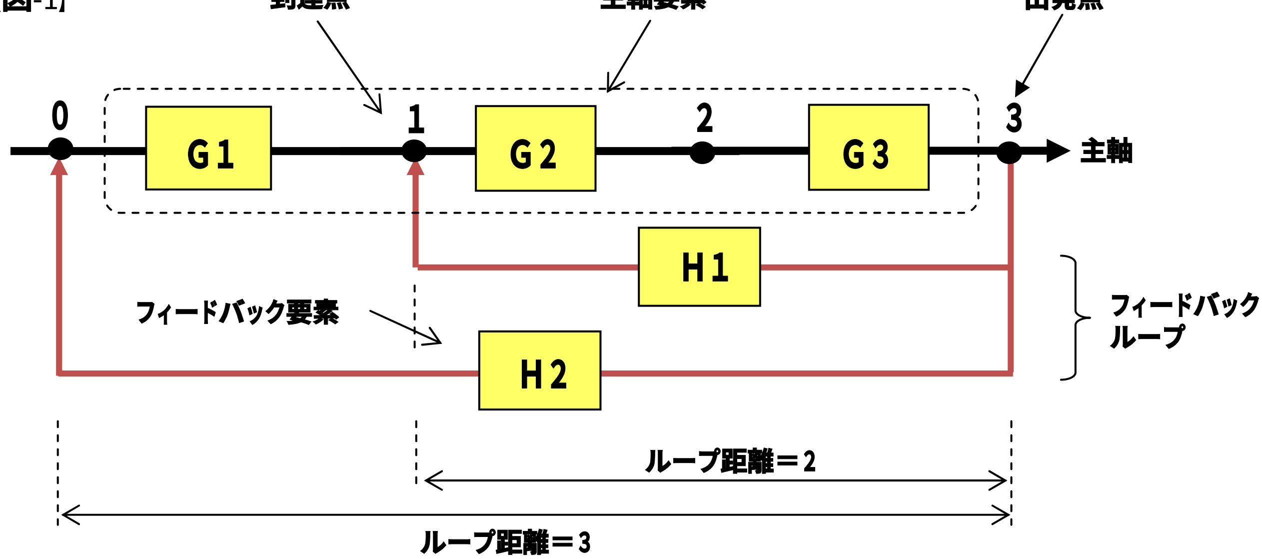
(4) フィードバックループが主軸から分岐して出発する位置を **出発点** と呼ぶ。
 a．出発点については、1番目の主軸要素と2番目の主軸要素の間を 1 、2番目の主軸要素と3番目の主軸要素との間を 2 と定義し、それ以降順次番号を付ける。

(5) フィードバックループが主軸に合流する点を **到達点** と呼び、出発点と同じ番号を使いる。

(6) フィードバックループの出発点と到達点間の位置の差を **ループ距離** と呼ぶ。

(7) 以下に 出発点、到達点 と ループ距離 の例を示す。
 a．フィードバック要素 H1 が構成するフィードバックループ
  o 出発点＝3、到達点＝1、ループ距離＝2
 b．フィードバック要素 H2 が構成するフィードバックループ
  o 出発点＝3、到達点＝0、ループ距離＝3

【図-1】



2．また、ブロック線図を等価変換した結果得られる式を、系の伝達特性を表す意味で特性式と呼ぶ。また一般に使われている伝達関数という呼称も適宜併用する。

【研究の動機】

1．電力系統は、数多くの発電機や変圧器が送電線や配電線で連系された大規模で複雑なネットワークであり、時々刻々変化する電力需要に対応して電力を供給しつつ、電圧や周波数を一定に維持していくために、さまざまな制御が行われている。

2．また、パワーエレクトロニクス技術の進歩により、交流を波形整形して使用する機器が今後ますます電力系統へ接続されていくことが予想され、電力系統で発生する現象の理解や解析には、伝統的な電力系統工学や電気機械工学のみならずパワーエレクトロニクスや制御関係など最新の知識が必要とされる状況にあり、将来はますますその傾向が強くなっていくものと考えられる。

3．そこで、実務を通じて得られる知識のほかに、基礎的な学問についていわゆる自学自習を通じて社員各自が研鑽を深め、技術的能力を深めていくことが大変重要と考えられる。

4．一方、電力会社では、毎年数百人規模で技術系社員を採用している。以前は大学卒の新入社員は電力関係や関連分野を専攻してきた人が多く、基礎力が十分にあったので、入社後の研修ないしは実務を通じてのＯＪＴなどにより自ずと業務に必要な知識を身に着けていくことができた。

5．しかしながら、近年、例えば大学の電気工学関係の学科では、電力工学などいわゆる重電関係の講座をもつ学校が減ってきており、電力会社に入社する社員も電力工学以外の分野を専攻してきた人が多くを占めるようになってきた。また、新入社員の中には工業高校を出て入社してくる人も多い。

6．これらの人に十分な知識を身につけてもらうことは会社として重要なテーマであるが、企業では学校のようにまとまった時間を教育に割り当てるのは困難であり、結局のところ個々人の努力に期待せざるを得ない状況である。

7．しかしながら、仕事をしながら勉強を続けることはかなり努力が必要である。

8．研究を始めるにあたり、電気主任技術者の資格試験に挑戦している何人かの人たちにインタビューしたところ、回路理論や電気磁気学など理論的な科目は見通しがきくので比較的理解しやすいが、自動制御やパワーエレクトロニクス、電気機械工学など抽象度が高く、教科書の説明では概念がつかみにくい科目であり勉強しにくく不得意、といった意見がきかれた。

9．このような状況に鑑み、教科書で説明しきれないばかりに学習者が苦労して理解せざるをえない抽象的な事象を、コンピュータを用いてわかりやすく解説し、学習者の負担を軽減できるような教材を開発し、社内のネットワークを通じて広く普及させれば、社員の基礎的な知識獲得に寄与できるのではないかと考えたことが研究の動機である。

【研究の方向性】

1. 大きな目標は、前述した自動制御、パワーエレクトロニクス、電気機械工学など教科書に示される図面と数式だけでは、現象が具体的にどのように推移していくのか直観的に把握することが難しく、その点が学習者の理解を阻害している教科について、
   a. コンピュータのグラフィック能力を活用して現象を動的に見せる。
      - 三相交流の交流直流変換の現象と整形波形の関係
      - 発電機・電動機内部の電磁界の動きと電圧・電流ベクトルの関係

   b. 特性式やパラメータを自らチューニング可能な機能を開発し、より深い理解を得させる。

   などを狙った教育ツールの研究を進めることにあるが、

2. 当面は、古典的な自動制御の分野に的を絞って研究を進めることとする。

3. 自動制御の分野では
   a. ブロック線図の等価変換による簡単化
   b. ボード線図やナイキスト線図の作製による系の特性の把握

   などが主な学習.のテーマになると考えられるが、今回は最も基礎的なブロック線図の簡単化をテーマとして、教材開発を試みる。

【研究のねらい】

1. インタビューの結果、自動制御の分野は数学的な要素が多く、具体的な現象をイメージしにくいのでわかりにくい、という意見があった。

2. 系の特性を解析するためのツールであるボード線図などを使って勉強する前に、最も基本的なブロック線図の簡単化のところで躓いている人も見受けられた。

3. 基本的なブロック線図等価変換のルール自体はそう難しいものではないが、ブロック線図の要素数が増えて、ループが複雑になると、基本的なルールをどのように適用して系を簡単化していくのかが分からなくなり、「問題が解けない」という状況になっているものと考えられる。

4. このようなことを踏まえ、今回の研究は、自動制御の学習では最も基本的な部分であり、その後いろいろな面で研究の基礎となると考えられるブロック線図の等価変換に的を絞った。

5. 上記の学習者の悩みに対するアプローチの方法はいろいろ考えられると思うが、本研究では第3種電気主任技術者受験者を対象レベルとして、以下の事項を重点として研究を行うこととした。
   a. 学習者がなぜ難しく感じているのかを現状の教科書の記述内容を基に考察し、その要因を探って、ブロック線図と等価変換プロセスの対応関係が明確に理解できる学習教材のモデルを試作する
   b. 学習教材を補助する問題演習システムとして、文字式による等価変換問題を出題する教材を試作し、出題される個々のブロック線図の形を計算機が認識して、等価変換結果である伝達関数を計算機が構成できることを確認して、実用化可能なツールが作成できるか確認する

【II．現状の教科書･演習書の調査】

1．研究を始めるにあたり、計算機を使って文字式によるブロック線図の等価変換を扱った教科書･演習書があるかどうか 調査したが、調査した範囲では見当たらなかった。

2．MATLAB、Scilabなどの数値計算ソフトを用いて、ブロック線図で表現した周波数伝達関数の計算方法を取り扱った演習書はあったが、内容は数値計算が主でブロック線図の等価変換を文字式で扱ったものではなく、また初学者向けの内容ではなかった。

3．市販の教科書･演習書で紹介されているブロック線図の等価変換の例題について、2つの典型的な例を次に示す（注釈などを除き、ほぼ原文通り）。

（1）市販の演習書の方法の例（例題1と呼ぶ）

例題演習　自動制御入門　（若山伊三雄　昭和63年3月　産業図書株式会社出版）　52ページ

［例題1・62］　図のブロック線図を簡単化せよ。

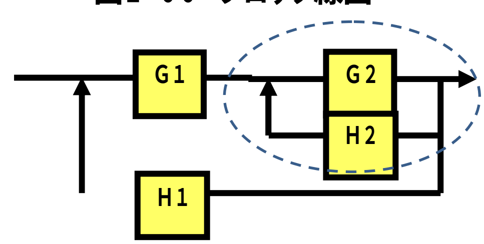
［解］

（1）図1・66の第2ループ（注釈：点線部分）を演算規約8を使って簡単化すると

$$W_2(s)=\frac{G2}{1+G2H2}$$

となり、図1・67（a）のようになる。

図1・66　ブロック線図

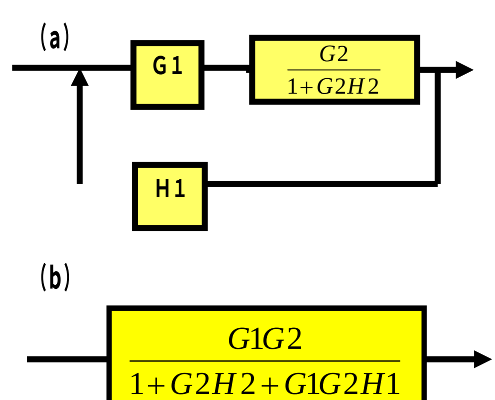


（2）図（a）のループを演算規約8を使って簡単化すると

$$W_2(s)=\frac{\dfrac{G1G2}{1+G2H2}}{1+\dfrac{G1G2H1}{1+G2H2}}=\frac{G1G2}{1+G2H2+G1G2H1}$$

となり、図（b）のようになる。

図1・67　ブロック線図の簡単化



(2) 市販の演習書の方法の例 (例題2)

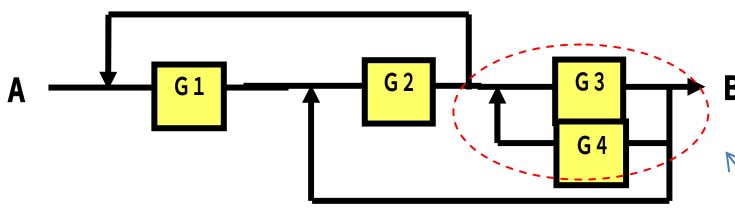
電気工学入門演習　自動制御　(佐藤達男・佐藤雄司　1999年3月　学献社出版)　42ページ

【例題1・39】　第1・86図 (a) のブロック線図を簡単化にし、入力A、出力Bの場合の周波数伝達関数を求めよ。

【解】　第1・86図 (a) において、G3、G4の小ループをまとめると、(b) 図のようになる。

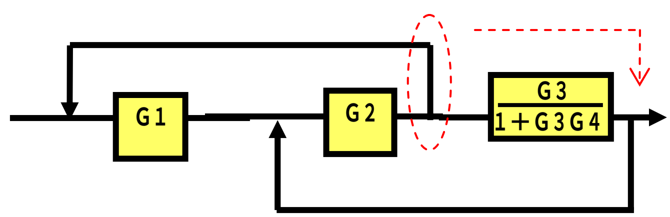
第1・86図

(a)



(注1)
ループ1を1つの主軸要素に等価変換

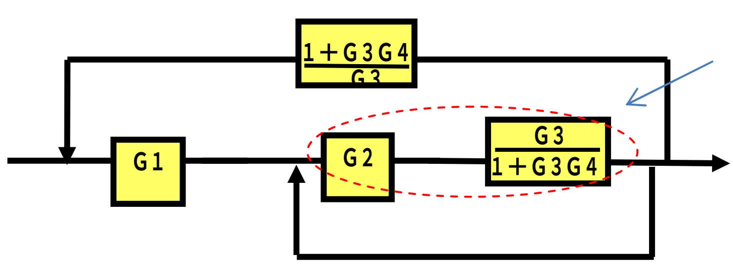
(b)



(注2)
ループ2の出発点を主軸要素3の後に移動

次に (b) 図において、引き出し点を伝達要素G3／1+G3G4をこえて信号B側へ移動させると、移動させたフィードバック経路に1+G3G4／G3が伝達要素として入るので (c) 図のようになる。

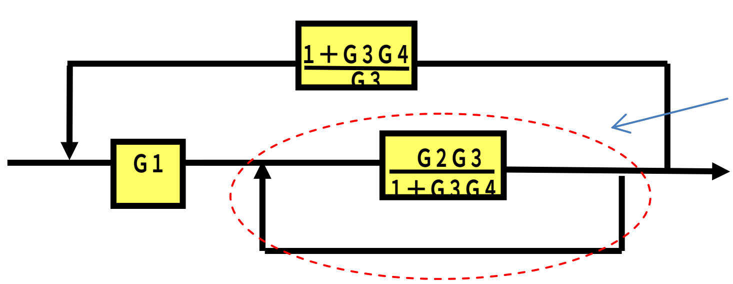
(c)



(注3)
直列する主軸要素を1つの主軸要素に等価変換

(c) 図においては、小ループの前向きの要素をまとめると (d) 図が得られる。

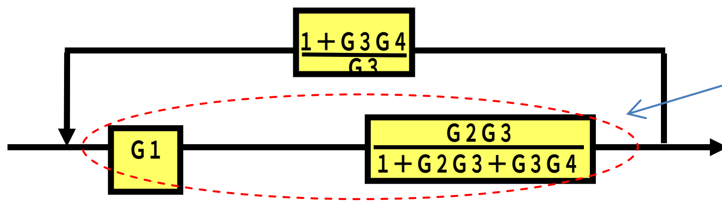
(d)



(注4)
ループ3を1つの主軸要素に等価変換

さらに (d) 図の小ループを整理して (e) 図が得られる。

(e)



(注5)
直列する主軸要素を1つの主軸要素に等価変換

すなわち、小ループの周波数伝達関数G'は

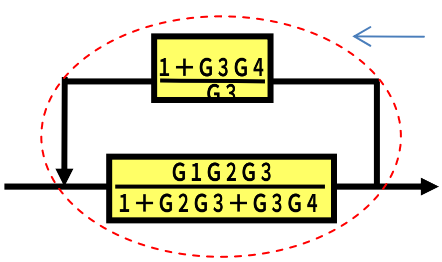
$$G' = \frac{\dfrac{G2G3}{1+G3G4}}{1+\dfrac{G2G3}{1+G3G4}} = \frac{G2G3}{1+G2G3+G3G4}$$

となる。同様にして、(e) 図をまとめた (f) 図から合成周波数伝達関数Gが

$$G = \frac{\dfrac{G1G2G3}{1+G2G3+G3G4}}{1+\dfrac{G1G2G3}{1+G2G3+G3G4}\,\dfrac{1+G3G4}{G3}} = \frac{G1G2G3}{1+G1G2+G2G3+G3G4+G1G2G3G4}$$

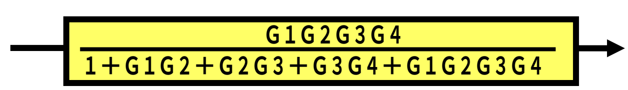
として求まるので、(g) 図のような簡単な形になる。

(f)



(注6)
ループ2を1つの主軸要素に等価変換

(g)



4．例題で用いられている等価変換の方法の特徴

(1) 例題の特徴

a．例題1　；　主軸要素数＝2でループ数は2。等価変換回数は2。
b．例題2　；　主軸要素数＝3でループ数は3。等価変換回数は6。

(2) 解法の特徴

a．最も内側にあるループから変換を開始して逐次外側のループを等価変換している。
b．内側のフィードバックループを主軸要素に等価変換する。この変換された主軸要素と外側のフィード バック要素で構成されるフィードバックループを再度主軸要素に等価変換する操作を反復している。
c．フィードバックループを等価変換するための計算式には連分数を用いている。
d．等価変換を進めていくたびにブロック線図を書き 換え段階的に計算を進めている。

5．分析

(1) 例題1は主軸要素数＝2であり、等価変換回数も2回で済むため理解しやすいと考えられる。
(2) 例題2は主軸要素数＝3であり比較的小規模なモデルであるが、複雑なループ構成が可能となり問題の難易度が上がるので、等価変換回数も6回と多くなるものと考えられる。
(3) 計算手法に連分数を用いているが、例題3では多重の連分数構成となるので1回で式を書くことが困難。このため、等価変換のたびにブロック線図を書き換えて段階的に問題を解く方法を採用せざるを得ないものと考えられる。
(4) 例題1・2とも物理的な問題ではなく数学の計算問題となってしまっている。

6．考察

(1) これらの例題は、学習者がブロック線図の形を見てどの部分にどの等価変換規則をどの順序で適用していくかを学ぶガイドとなっており、与えられた問題を解かせることにより基本的な等価変換規則の適用方法を学習者に自得させる構成となっている。文字と図面を使った説明では必然的にこのような方法になるものと考えられるが、

a．ブロック線図の等価変換は実質的には数学の計算問題であるため、ループ構成が複雑になると物理的なイメージが掴みつかみにくくなり、学習者から「数学的な要素が多く抽象的で分かりにくい」という意見が出ると考えられる。
b．また、主軸要素数が2から3に増えると問題が難しくなり等価変換回数も増える。一方、連分数を用いているので、全体を見通した解き方を示すのが難しく、内側のループから一つ一つ等価変換を積み重ねていく方法しかとれない。このため、学習者から「テキストが分かりにくい」などの意見が出るものと考えられる。

(2) 従って、これらブロック線図の等価変換にかかわる問題を解決するためには、

a．連分数のように計算式の構造自体が複雑な計算方法ではない簡単な計算方法
b．ブロック線図の形（ループ構造）を計算式に反映可能で、ブロック線図の構造を計算式からも把握できるようなできる計算方法
・・・信号伝達の物理現象を反映した方法は難しいので形が理解できるような
c．可能であれば、ブロック線図の構造と計算式の形との対応関係を視覚などにうったえて理解の促進を図ることができる学習支援手段など

が必要と考えられる。

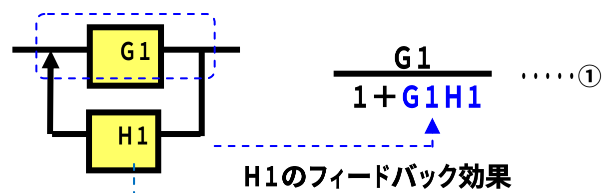
【III. ブロック線図の形と等価変換後の特性式の形について】

1．現状の教科書・演習書の調査で提示した例題は一般的なケースを網羅していないので、ブロック線図の形と特性式の形の間にどのような関係があるか把握するため、典型的な形のブロック線図について対応する特性式を連分数を用いて計算した。
ここでは結果のみ示す。

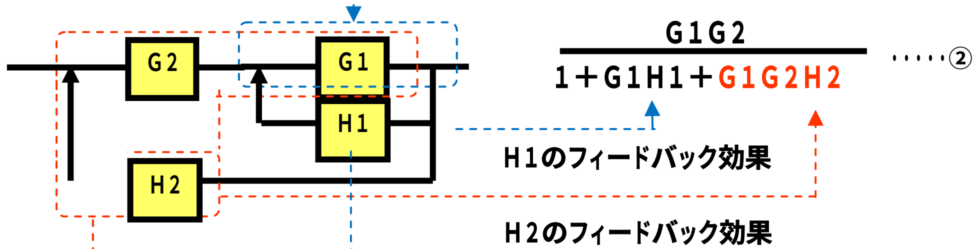
（ケース1）ループが包含関係にある場合

①～③に示すように、1つの主軸要素に対して1つのフィードバック要素が対応する場合について、主軸要素数＝1から3までの計算を行った。図の右側に計算結果を示す。

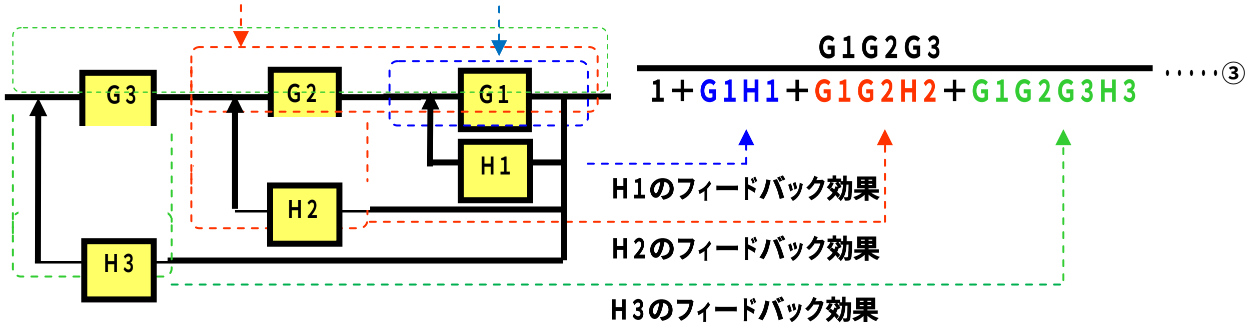
①主軸要素数＝1の場合



②主軸要素数＝2の場合



③主軸要素数＝3の場合

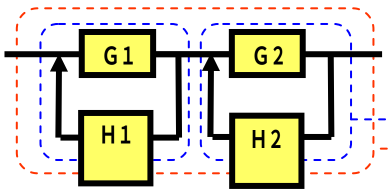


計算結果は以下の通り。
- 分子は主軸要素の積・・・G1、G1G2、G1G2G3　となる。
- 分母は各フィードバックループが単独で存在する場合のフィードバック効果の和となる。
  - 主軸要素数＝1の場合　1＋G1H1
  - 主軸要素数＝2の場合　1＋G1H1＋G1G2H2
  - 主軸要素数＝3の場合　1＋G1H1＋G1G2H2＋G1G2G3H3

（ケース２）複数のループが独立して存在する場合

④に主軸要素数＝１のループが２つ独立して存在する場合を示す（計算結果は図の右側）。

④２つの独立ループがある場合



$$\frac{G1G2}{1+G1H1+G2H2+G1H1G2H2} \quad \cdots ④$$

２つのループのフィードバック効果の積

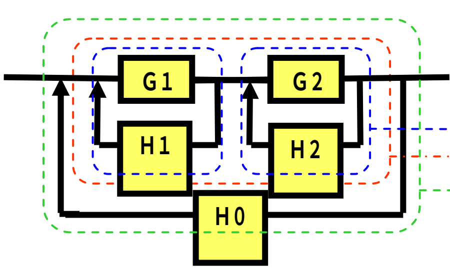
ａ．計算結果は以下の通り。
- ｏ 分子は主軸要素の積 Ｇ１Ｇ２ になる。
- ｏ 分母は２つのループのフィードバック効果の積になり
  - ・（１＋Ｇ１Ｈ１）（１＋Ｇ２Ｈ２）＝１＋Ｇ１Ｈ１＋Ｇ２Ｈ２＋Ｇ１Ｈ１Ｇ２Ｈ２

ｂ．独立したループが ｎ 個あるとき以下のようになることは明らか。
- ｏ 分子は主軸要素の積 Ｇ１･･･Ｇｎ になる。
- ｏ 分母は、ｎ個のフィードバック効果の積
  （ １＋Ｇ１Ｈ１ ）（１＋Ｇ２Ｈ２ ）･･･（ １＋ＧｎＨｎ ） となる。

（ケース３）複数の独立ループを包含するループが存在する場合

⑤に２つの独立ループを包含するループが存在する場合を示す（計算結果は図の右側）。

⑤



$$\frac{G1G2}{1+G1H1+G2H2+G1H1G2H2+G1G2H0}$$

２つの独立ループを包含するループのフィードバック効果

ａ．計算結果は以下の通り。
- ｏ 分子は主軸要素の積 Ｇ１Ｇ２
- ｏ 分母は２つのループのフィードバック効果の積 （１＋Ｇ１Ｈ１）（１＋Ｇ２Ｈ２） にこれらを包含するループのフィードバック効果 Ｇ１Ｇ２Ｈ０ を加えた形になっている。

  ・（１＋Ｇ１Ｈ１）（１＋Ｇ２Ｈ２）＋ Ｇ１Ｇ２Ｈ０
  ＝１＋Ｇ１Ｈ１＋Ｇ２Ｈ２＋Ｇ１Ｈ１Ｇ２Ｈ２＋Ｇ１Ｇ２Ｈ０

３. 上記の計算結果をまとめると、ブロック線図の形と特性式の形との間には以下のような対応関係が成立すると考えることができる。

(1) 特性式の分子は主軸要素の積になる。

(2) 特性式の分母は、

a. 複数のループが包含関係の形になっている場合
- それぞれのループのフィードバック効果の和が分母に現れる。

b. 複数のループが独立に存在する形になっている場合
- それぞれのループのフィードバック効果の積が分母に現れる。

c. 独立に存在するループとそれを包含するループが存在する形の場合
- 独立するループのフィードバック効果の積と、独立ループを包含するループのフィードバック効果の和が分母に現れる。

５．従来の連分数計算を用いて特性式を求めた結果
（各等価変換ステップに対応するブロック線図の書き換えは省略）

② $$\frac{G2\dfrac{G1}{1+G1H1}}{1+G2\dfrac{G1}{1+G1H1}H2}=\frac{G1G2}{1+G1H1+G1G2H2}$$ ①の結果を使用

③ $$\frac{G3\dfrac{G1G2}{1+G1H1+G1G2H2}}{1+G3\dfrac{G1G2}{1+G1H1+G1G2H2}H3}=\frac{G1G2G3}{1+G1H1+G1G2H2+G1G2G3H3}$$ ②の結果を使用

④ $$\frac{G1}{1+G1H1}\times\frac{G2}{1+G2H2}=\frac{G1G2}{(1+G1H1)(1+G2H2)}$$ ①の結果を使用

⑤ $$\frac{\dfrac{G1G2}{(1+G1H1)(1+G2H2)}}{1+\dfrac{G1G2}{(1+G1H1)(1+G2H2)}H0}=\frac{G1G2}{(1+G1H1)(1+G2H2)+G1G2H0}$$ ③の結果を使用

なお、③の計算で②の結果を流用せず、はじめから計算した場合は以下のような複雑な式となる。

$$\frac{G3\dfrac{G2\dfrac{G1}{1+G1H1}}{1+G2\dfrac{G1}{1+G1H1}H2}}{1+G3\dfrac{G2\dfrac{G1}{1+G1H1}}{1+G2\dfrac{G1}{1+G1H1}H2}H3}=\frac{G3\dfrac{G1G2}{1+G1H1+G1G2H2}}{1+G3\dfrac{G1G2}{1+G1H1+G1G2H2}H3}=\frac{G1G2G3}{1+G1H1+G1G2H2+G1G2G3H3}$$

【IV. ブロック線図の形を反映した計算方法】

1. ブロック線図の等価変換に関する問題点を解決するため、ブロック線図の形を計算式や計算プロセス反映できる方法について検討を行った。

2. 既存の方法

(1) 複雑な制御系の制御ブロックを等価変換する方法として、シグナルフローグラフ がある。

a. この方法は、信号をノードとその伝達方向・要素を矢印で表したブランチで表し、ノードとブランチにより信号の入出力の関係を記述したもので、トランスミッタンスと呼ばれる信号間の関係をブランチの途中に記入して表現する方法である。

b. しかしながら、この方法は、例えば、あるノードで信号が2つに分岐する場合、上流側を入力、下流側を出力とすると、下流側の出力信号がいずれも上流側のトランスミッタンスを含む形で、信号の入出力の関係が記述できるときにノードが省略できて、等価な形に簡単化していく方法で、連分数を用いた方法より高度であり、初学者向きではなかった。

(2) また、グラフ理論の応用なども想定できるが、これについても初学者向けの内容ではない。

3. このため、連分数を使わずにブロック線図の等価変換演算を表現できる方法として、第三種電気主任技術者試験を受験するレベルの学習者であれば当然知っている、回路理路の4端子定数の計算方法をヒントに行列を使った計算方法の適用について検討を行った。

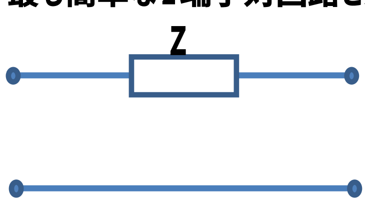
4. 行列を使った計算方法について

(1) 回路理論では、4端子定数を用いた計算方法が周知である。回路理論では電圧と電流の2つの変数に関する入力と出力の比として4端子定数を設定する。ブロック線図で扱うのは信号という1つの変数であるが、ブロック線図は信号の入力と出力の比と考えることができるので、回路理論との共通性がある。

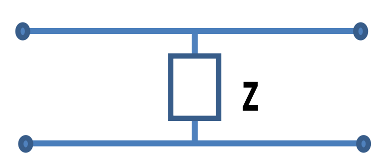
(2) また、4端子定数を使った計算方法では、複雑な構成の回路であっても、それを最も簡単な構成の2端子対回路の直列・並列に組み合わせて表現しているこので、フィードバックループの直列結合と並列結合で構成されるブロック線図との共通性がある。

(3) 更に、4端子定数の計算では、最も簡単な構成の2端子対回路の定数を2行2列の行列で表現し、回路構成に対応した行列の積を作ることで回路全体の定数を表現している。

最も簡単な2端子対回路と対応する4端子定数(A,B,C,D)の行列表現の例



$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 1 & Z \\ 0 & 1 \end{pmatrix}$$



$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1/Z & 1 \end{pmatrix}$$

(4) これらのことから、4端子定数の計算と同じように、ブロック線図に現れる伝達要素を簡単な行列で表現し、行列を組み立てることで全体の特性が記述できるのではないかと考えた。

5．検討内容

ブロック線図の特性を行列で表現するに当たり、以下の方法で検討を行った。

⑴ 4端子定数と同様、行列の形は最も簡単な2行2列と仮定した。

⑵ 行列の種類

ａ．2つの主軸要素Ｇ1とＧ2の直列結合は積Ｇ1Ｇ2となる

ｂ．主軸要素Ｇ1とフィードバック要素Ｈ1から構成されるループの伝達関数は $\dfrac{G1}{1+G1H1}$ となる

ことから、主軸要素とフィードバックループにはそれぞれ異なる行列が対応すると考え、

- 主軸要素に対応する行列を主軸行列
- フィードバックループに対応する行列をフィードバック行列

として2種類の行列を設定しその形を検討した。

⑶ 前述したように、2つの単純な主軸要素Ｇ1とＧ2の直列結合は単純な積Ｇ1Ｇ2となるので、

主軸行列の形は $\begin{pmatrix} G1 & 0 \\ 0 & G1 \end{pmatrix}$ または $\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix}$ の対角行列となると考えた。

⑷ また、前述のようにフィードバックループの伝達関数が $\dfrac{G1}{1+G1H1}$ となることから、

主軸行列とフィードバック行列の積をとった場合、行列の要素にG1、G1H1、1の要素が 現れなければならないと考え、いろいろな形の行列の積をとってみた。

⑸ その結果、フィードバック行列の形を $\begin{pmatrix} 1 & H1 \\ 0 & 1 \end{pmatrix}$ として主軸行列との積をとると

ａ．主軸行列が $\begin{pmatrix} G1 & 0 \\ 0 & G1 \end{pmatrix}$ の場合 $\begin{pmatrix} G1 & 0 \\ 0 & G1 \end{pmatrix}\begin{pmatrix} 1 & H1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G1 & G1H1 \\ 0 & G1 \end{pmatrix}$

ｂ．主軸行列が $\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix}$ の場合 $\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & H1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G1 & G1H1 \\ 0 & 1 \end{pmatrix}$

となって、主軸行列の形を $\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix}$ フィードバック行列の形を $\begin{pmatrix} 1 & H1 \\ 0 & 1 \end{pmatrix}$ とすれば、

行列の要素を G1，G1H1，1 とできることがわかった。

なお、主軸行列とフィードバック行列の積の順序は $\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & H1 \\ 0 & 1 \end{pmatrix}$ である。これが逆になると積の結果が $\begin{pmatrix} G1 & H1 \\ 0 & 1 \end{pmatrix}$ となって正しい値とならない。

(6) また、主軸行列とフィードバック行列の積の結果を $\begin{pmatrix} G1 & \mathrm{G1H1} \\ 0 & 1 \end{pmatrix}$ の形のままにしておくと、

フィードバックループの直列結合の場合、正しい答えが得られない。(6－(6) 参照)

これを解決するため、フィードバックループを等価変換した結果は主軸要素となることをヒントに

主軸行列 $\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix}$ とフィードバック行列 $\begin{pmatrix} 1 & \mathrm{H1} \\ 0 & 1 \end{pmatrix}$ の積をとった行列 $\begin{pmatrix} G1 & \mathrm{G1H1} \\ 0 & 1 \end{pmatrix}$ については

1行2列の要素　G１H１　を　2行2列　に移項して加え、対角行列化する演算規則を設定した。

$$\begin{pmatrix} G1 & \mathrm{G1H1} \\ 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} G1 & 0 \\ 0 & 1+\mathrm{G1H1} \end{pmatrix}$$

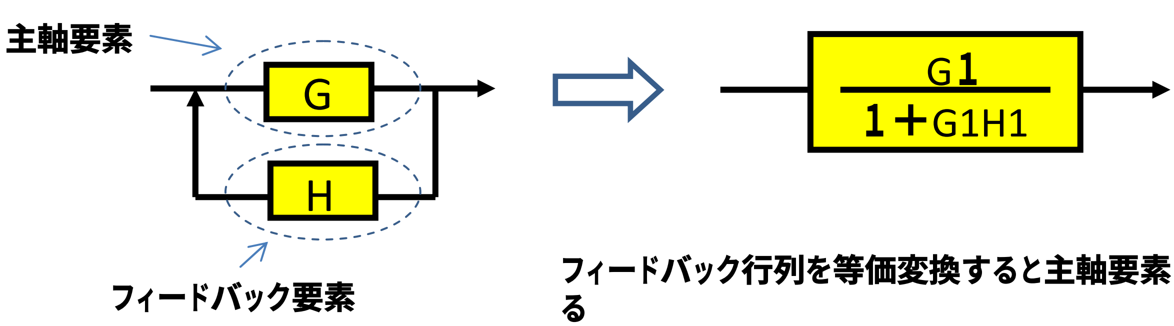
(7) その結果、フィードバックループの直列結合の計算で発生する演算規則の不整合を解消することができ (6－(6) 参照) 、設定した演算規則を全ての形のブロック線図へ適用することが可能になった。

(8) また、この形にすることで等価変換後の主軸行列について図Ⅳ－1に示すように
a．1行1列の要素を特性式の分子
b．2行2列の要素を特性式の分母　　　　に対応する形とすることができた。

(9) 行列の組み立て順序 (計算の順序) は、従来の方法と同じく以下の2つである。
a．内側のループから先に計算を行う順序で組み立てる
b．ループ内に直列の要素があればそちらを優先して計算を行う順序で組み立てる

(10) なお、2つのループの出発点と到達点が交差する場合は、分岐点･合流点の変更のルールを適用してフィードバック要素を修正して適用する (この部分は学習者の判断になる) 。

図Ⅳ－1



フィードバック行列を等価変換すると主軸要素になる
主軸行列とフィードバック行列の積をとって

$$\begin{pmatrix} G1 & 0 \\ 0 & 1+\mathrm{G1H1} \end{pmatrix}$$

の形にすると、
o　1行1列の要素は特性式の分子
o　2行2列の要素は特性式の分母
に対応させることができる。

6. 演算規則の検証

任意の形のブロック線図の等価変換を正しく行うためには、主軸行列とフィードバック行列により基本的な演算が正しく行えることを検証しなければならない。ここでは次の (1) から (5) までを基本的な演算として検証を行った。

(1) 主軸要素の直列結合

a. ブロック線図の形

b. 行列を使った演算

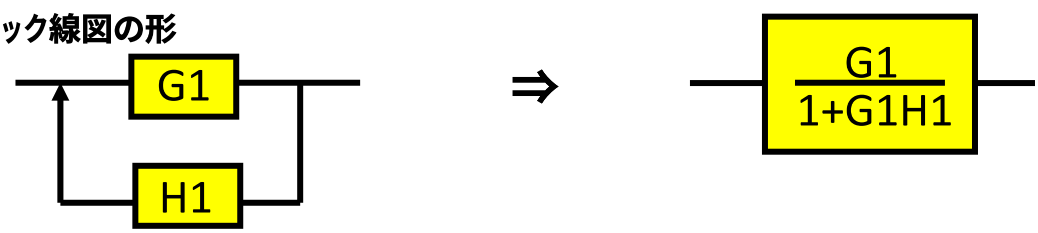
$$\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} G2 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G1G2 & 0 \\ 0 & 1 \end{pmatrix}$$

c. 結果

- 1行1列の要素の積が主軸要素の積に対応しG1G2となる。
- フィードバックがないので、2行2列の要素は1のままであり、主軸要素行列の積をとると直列結合が正しく演算できる。

(2) フィードバックループ

a. ブロック線図の形



b. 行列を使った演算 (積の結果の行列の1行2列の要素を2行2列に移項して加える)

$$\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & H1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G1 & G1H1 \\ 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} G1 & 0 \\ 0 & 1+G1H1 \end{pmatrix}$$

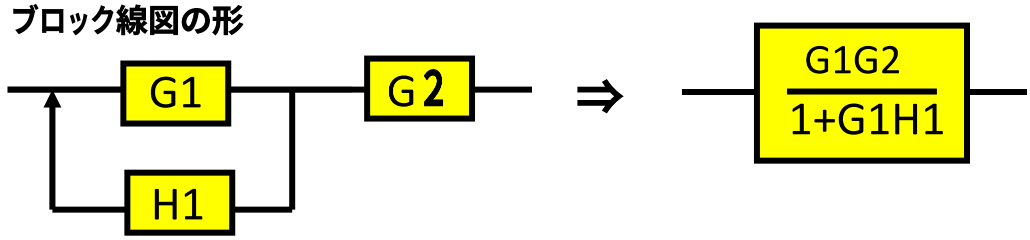
c. 結果

- 1行1列の要素は主軸要素G1となる。
- 2行2列の要素は1＋G1H1となり、フィードバック効果を表すことができる。
- 主軸要素行列とフィードバック行列の積をとることでフィードバックが正しく演算できる。

d. この移項操作により1行1列の要素を特性式の分子、2行2列の要素を特性式の分母に対応させることができる。
＊ この操作を行わない場合の結果については (6) の注意参照。

(3) フィードバックループと主軸要素の直列結合の場合

a. ブロック線図の形



b. 行列を使った演算

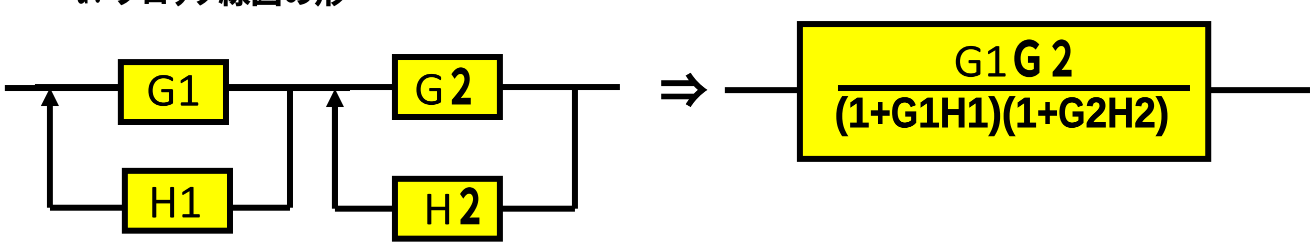
$$\begin{pmatrix} G1 & 0 \\ 0 & 1+G1H1 \end{pmatrix}\begin{pmatrix} G2 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G1G2 & 0 \\ 0 & 1+G1H1 \end{pmatrix}$$

c. 結果

- 1行1列の要素は主軸要素G1とG2の積となる。
- 2行2列の要素は1＋G1H1のままで変わらない。
- 主軸要素行列と等価変換したフィードバックループの行列の積をとることで正しく演算できる。

(4) 2つのフィードバックループの直列結合

a. ブロック線図の形



b. 行列を使った演算

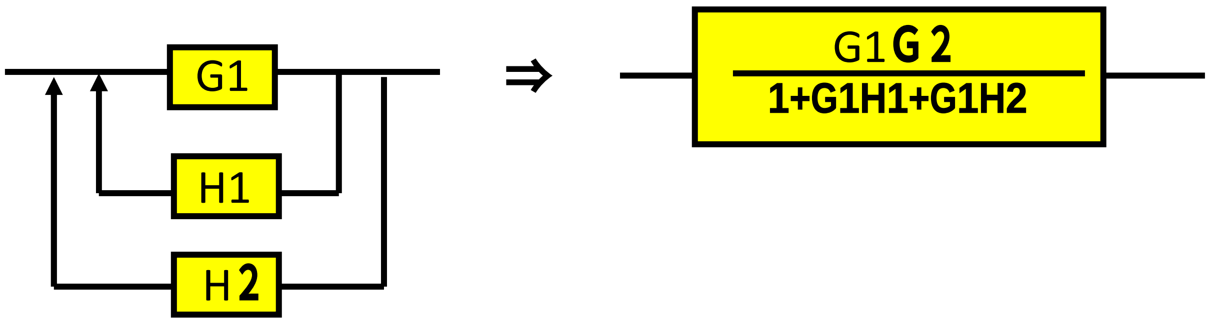
$$\begin{pmatrix} G1 & 0 \\ 0 & 1+G1H1 \end{pmatrix}\begin{pmatrix} G2 & 0 \\ 0 & 1+G2H2 \end{pmatrix} = \begin{pmatrix} G1G2 & 0 \\ 0 & (1+G1H1)(1+G2H2) \end{pmatrix}$$

c. 結果

- 1行1列の要素は主軸要素G1とG2の積となる。
- 2行2列の要素は2つのループのフィードバック効果の積（1＋G1H1）（1＋G2H2）となる。
- 等価変換したフィードバックループの行列の積をとることで正しく演算できる。

(5) 包含関係にあるループの場合

a. ブロック線図の形



b. 行列を使った演算

$$\begin{pmatrix} G1 & 0 \\ 0 & 1+G1H1 \end{pmatrix}\begin{pmatrix} 1 & H2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G1 & G1H2 \\ 0 & 1+G1H1 \end{pmatrix} \Rightarrow \begin{pmatrix} G1 & 0 \\ 0 & 1+G1H1+G1H2 \end{pmatrix}$$

c. 結果

- 1行1列の要素は主軸要素G1。
- 2行2列の要素は2つのループのフィードバック効果の和 1＋G1H1＋G2H2 となる。
- 等価変換したフィードバックループの行列とにフィードバック行列の積をとることで正しく演算できる。

(6) 注意

a. 1行2列の要素を2行2列に移項しないまま演算すると正しい結果を導かない。

b. 例えば、(4) の2つのフィードバックループの直列結合の場合

$$\begin{pmatrix} G1 & G1H1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} G2 & G2H2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} G1G2 & G1G2H2+G1H1 \\ 0 & 1 \end{pmatrix}$$

の形になって、この段階で1行2列の要素を2行2列に移項しても

$$\begin{pmatrix} G1G2 & 0 \\ 0 & 1+G1G2H2+G1H1 \end{pmatrix}$$

となって、 2行2列の要素は (1＋G1H1) (1＋G2H2) と一致しなくなる。

c. 移項してあれば、以下のように正しく演算ができる。

$$\begin{pmatrix} G1 & 0 \\ 0 & 1+G1H1 \end{pmatrix}\begin{pmatrix} G2 & 0 \\ 0 & 1+G2H2 \end{pmatrix} = \begin{pmatrix} G1G2 & 0 \\ 0 & (1+G1H1)(1+G2H2) \end{pmatrix}$$

7．行列を用いた演算方法に関するまとめ

(1) 演算規則

a．主軸要素は $\begin{pmatrix} G & 0 \\ 0 & 1 \end{pmatrix}$ の形の主軸行列で表し、フィードバック要素は $\begin{pmatrix} 1 & H \\ 0 & 1 \end{pmatrix}$ の形のフィードバック行列で表す。

b．フィードバックループの等価変換ではフィードバック行列を主軸行列の右側に置いて積をとる。

c．フィードバックループの等価変換を行う都度、1行2列の要素を2行2列に移項して加算する。

d．2つのループの出発点と到達点が交差する場合は、分岐・合流点の変更の規則を適用してフィードバックループの出発点または到達点の変更を行い、フィードバック要素を補正する。

e．合わせて、主軸要素またはフィードバック要素が並列に結合している場合は単純に行列の和をとることができないので、手で補正する規則を追加する。

o 主軸要素の場合 $\begin{pmatrix} G1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} G2 & 0 \\ 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} G1+G2 & 0 \\ 0 & 1 \end{pmatrix}$

oフィードバック要素の場合 $\begin{pmatrix} 1 & H1 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & H2 \\ 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & H1+H2 \\ 0 & 1 \end{pmatrix}$

(2) ブロック線図の等価変換を行う際に出現する5つの基本的な形について、いずれも行列を使った方法で正しく計算できることがわかった。

a．2つの主軸要素の直列結合
b．フィードバックループの等価変換
c．フィードバックループと主軸要素の直列結合
d．フィードバックループの直列結合
e．フィードバックループの並列結合（二重のフィードバックループ）

(3) 従って、どのような形のブロック線図であってもこれらの演算式を組み合わせることにより、等価変換が可能であるといえる。

8．基本的な演算は行列を使って可能であることが分かったが、この方法が従来の連分数を使った方法よりわかりやすく、学習者の理解をより促進できる方法であるかどうか検証する必要がある。今回は学習者に使用してもらうことができなかったので、少し複雑なブロック線図の等価変換の計算を行って、わかりやすさと学習者の理解促進の可能性を考察した。

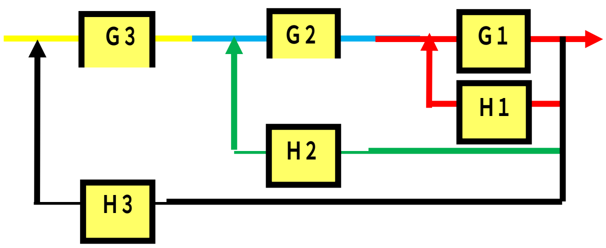
10．より複雑な例として、フィードバックループが3重になっている場合と2つのフィードバックループの直列結合の外側にフィードバックループが並列して結合している場合の2つのケースについて行列を使った方法を適用して計算した。

o これらのケースは、これまでの第3種電気主任技術者の試験問題の出題傾向（主軸要素数は2で、フィードバックループ数も2程度が主流、最近は少しずつ難しくなりつつある）を考慮して設定した。結果を次に示す。

11．行列を使った計算方法の有効性を示すための少し複雑なブロック線図の計算例。

(1) 少し複雑な場合　例－１ (３重ループ)

a．ブロック線図の形



ループの構成と特性式の対応を色で示す

b．行列を使った計算　(詳細に実施　；　演算順序は中括弧で示す)

$$\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1 & G1H1\\ 0 & 1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1G2 & G1G2H2\\ 0 & 1+G1H1\end{pmatrix}\right\}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H1+G1G2H2\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\ 0 & 1+G1H1+G1G2H2\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

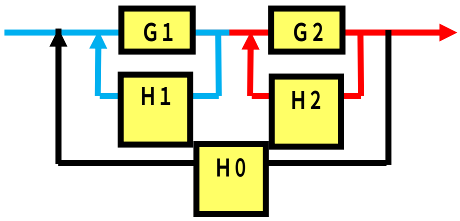
$$=\begin{pmatrix}G1G2G3 & G1G2G3H3\\ 0 & 1+G1H1+G1G2H2\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\ 0 & 1+G1H1+G1G2H2+G1G2G3H3\end{pmatrix}$$

c．先に計算済みの結果　$\dfrac{G1G2G3}{1+G1H1+G1G2H2+G1G2G3H3}$ ·····③　と一致する。

(2) 少し複雑な場合　例－２　(直列した２つのループに対して外側のループが並列に結合)

a. ブロック線図の形



ループの構成と特性式の対応を色で示す

b. 行列を使った計算　(詳細に実施　；　演算順序は中括弧で示す)

$$\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H0\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1 & G1H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H0\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\left\{\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H0\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\left\{\begin{pmatrix}G2 & G2H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H0\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1+G2H2\end{pmatrix}\right\}\begin{pmatrix}1 & H0\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & (1+G1H1)(1+G2H2)\end{pmatrix}\begin{pmatrix}1 & H0\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & G1G2H0\\ 0 & (1+G1H1)(1+G2H2)\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & (1+G1H1)(1+G2H2)+G1G2H0\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H1+G2H2+G1H1G2H2+G1G2H0\end{pmatrix}$$

c. 先に計算済みの結果

$$\frac{G1G2}{1+G1H1+G2H2+G1H1G2H2+G1G2H0} \quad \cdots ⑤$$

と一致する。

１２．考察

(1) 例－１では6つ、例－２では5つの行列により計算の組み立てが可能であり、最初に書き下した式でブロック線図全体の構造を表現できる。これは、従来の連分数を用いた方法では難しかった事項であるので、行列を使った方法を採用すれば学習者は、ブロック線 図の全体構造と系の特性を表現する式の構造の関係を把握することが容易になると考えられる。

o 連分数を使った方法でも最初に全体の式を書くことは不可能ではないが、式が複雑になりかなり難しいと思われる。また、ブロック線図の構成との結びつけも困難と思われる。

(2) 計算は2行2列の行列の積をとるだけなので容易であり、内側ループから外側ループに逐次等価変換が行われ、特性式の分子と分母の形が変わっていくことを確認しながら計算を進めることができるので、連分数を使った方法で等価変換のたびに図面を書き換えることと同じような効果をより簡単に得ることができると考えられる。

９．学習教材への適用

(1) 以上のことから、原理が理解できれば初学者でも簡単に等価変換の計算ができると思われ、また従来の連分数を用いた方法より学習者の理解を深めることができると考えられるので、行列を使ったブロック線図の等価変換を基本に教材を試作することにする。

(2) 教材は以下の構成で試作する。

ａ．ブロック線図の等価変換方法の解説

ｂ．合わせて、パソコン上で稼働する演習システム

(3) 行列を使った方法で課題があるとすれば、はじめに組み立てる行列の数が多いので、学習者が難しいと感じ、拒否感を持つことが考えられる。この点については、計算機のシミュレーション機能など活用して、できるだけわかりやすい教材を作成することを試みる。

【Ｖ．教材の試作】

１．行列を使ったブロック線図の等価変換の教材のモデルを以下のような条件で試作した。

(1) 試作であるため、内容は行列を使った特性式の分母の計算の説明方法にポイント を絞る。

(2) 学習者が会社のネットワークが接続されていない自宅で勉強できるよう、個人のパソコン上で単独に使用できるＣＤで配布型を念頭に教材を試作する。

(3) 但し、今後の展開を考慮し、信州大学のＣＡＩシステム (以下ＣＡＩシステムと呼ぶ) を活用し、主要な 説明はＣＡＩシステム上で行い、例題や練習問題はＣＡＩシステムにリンクさせたＥｘｃｅｌのプログ ラミング機能 (マクロ) を活用して作成する。

(4) 教材の内容は、行列を使った計算方法になじんでもらうことを考え

ａ．簡単な行列の積の計算の復習

ｂ．行列を使った等価変換の計算規則の説明

ｃ．シミュレーションによる例題の説明

などで構成した。

以下の説明では、行列を使った等価変換プロセスの理解を進めるための、シミュレーションによる例題について述べる。

２．簡単なシミュレーションによる例題　（図Ｖ－１を参照）

（1）２重ループで構成されるブロック線図の特性式の分母の計算を題材にした例。

（2）できるだけわかりやすい教材を作ることを念頭に、画面にはブロック線図の画像を表示し、合わせて対応する特性式の分母を計算する行列 の積の式を表示した。

（3）単に説明するだけでは従来の教科書と同じであるので、行列の積の表示には簡単なシミュレーション機能を持たせ、学習者が主軸行列の1行1列、およびフィードバック行列の1行2列に任意の文字式（現段階では単項式）を入力して、計算ステップを示すボタンを押すことで計算の進行を確認できるよう構成した。

（4）更に、内側ループの計算結果が外側ループ計算の初期値となって計算が進行することをわかりやすく表示するため、内側ループの計算と外側ループの計算式を分割して表示し、内側ループの計算結果を外側ループの計算に引き継ぐステップを設けた。

（4）計算を進めるボタンは以下の５種類用意し、各ボタンを押すことで学習者が計算を進めることができる構成とした。
①内側のループに対応する行列の積
②1行2列の要素の2行2列への移項･･･このステップで外側のループ計算の初期値を設定
③外側のループに対応する行列の積
④1行2列の要素の2行2列への移項
⑤計算結果のリセット

（6）また、ループの行列の積の計算が終了するたびに特性式を分数の形で表示する機能も備えた。

３．プログラム作成のポイント
（1）特別な画面設計を必要とせずに行列要素の配列を画面上で構成できるＥｘｃｅｌを用いて例題表示プログラムを作成した。

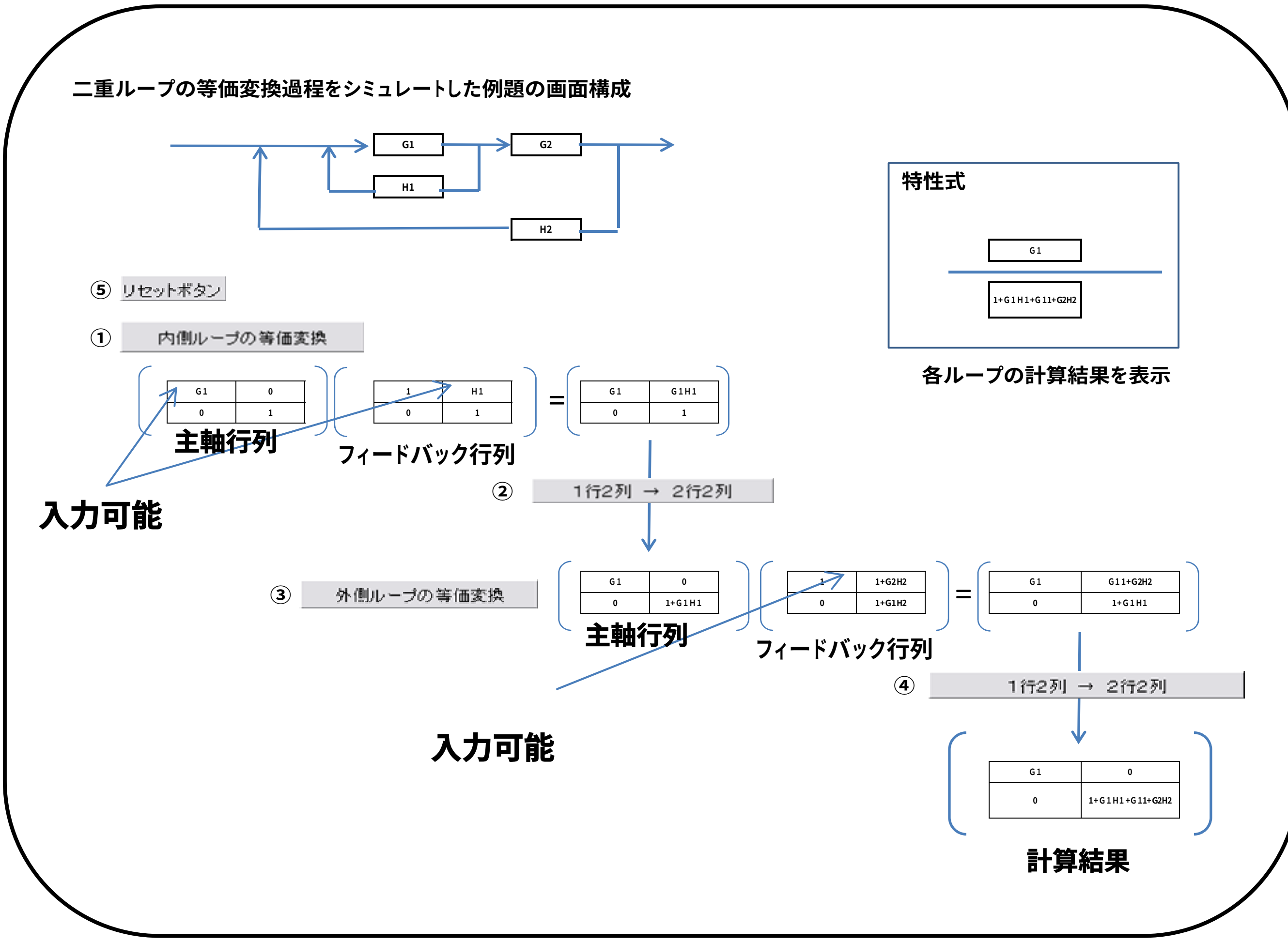
（2）シミュレーションは、セルに入力されたデータの和と積をマクロ機能で計算して表示する方式を用いた。

４．自己評価
（1）教材作成面
ａ．元来が表計算ソフトであるＥｘｃｅｌを用いたので、行列計算式の表示、行列に対する文字式の入力や計算結果の表示など、入出力に関係する画面が効率的に設計できた。
ｂ．Ｅｘｃｅｌで作成した簡単なシミュレーションをＣＡＩシステムから呼び出して使用する方策をとったことにより、他のプログラムとの併用がＣＡＩシステムの改造を伴わずにＣＡＩシステムを活用する１つの方策であることを再認識した。

（2）教育効果面
ａ．例題の説明にシミュレーション機能を付加したので、行列を使った計算過程を分かりやすく説明できたのではないかと考えられる。
ｂ．しかしながら、時間の関係で学習者に使ってもらっていないので、今後社内で試験的に使ってもらって評価してもらうこととしたい。

図Ｖ－１　簡単なシミュレーションによる例題の表示画面の構成

ＣＡＩシステムにリンクを張ったが、Ｅｘｃｅｌで作成してあるので単独でも使用できる。



- ｏ　各ボタンにマクロ機能が設定されており、裏側で行列計算を行っている。
- ｏ　内側ループの主軸行列の１行１列の要素とフィードバック行列の１行２列の要素に文字を入力して　「内側ループの等価変換」のボタンを押すと計算が行われる。
- ｏ　「１行２列→２行２列」のボタンを押すと、計算結果の１行２列の要素が２行２列に移項され加算され、外側ループの主軸行列に代入される。
- ｏ　外側ループのフィードバック行列の１行２列の要素に文字を代入して、外側ループの等価変換のボタンを押すと行列計算が実行される。
- ｏ　次のステップは内側ループと同じ。
- ｏ　計算結果は右下の行列に表示される。
- ｏ　合わせて、右上の特性式と表示した分数式の中には、伝達関数の形でループの特性が表示される。

【VI．計算機による等価変換演習教材の試作に関する課題の検討】

1．行列によるブロック線図の等価変換の計算方法が考案できたので、補助教材として計算機による演習教材を試作した。

2．教材はパソコン上で稼働し、基本的な部品をランダムに組み合わせて画面上に問題として表示したブロック線図に対して、学習者が回答すればパソコンが答え合わせを行い結果を表示する方式とした。また、ブロック線図の等価変換のルールを学習することを目的としたため、数値計算ではなく、文字式の演算による演習を行う方式とした。

○　なお、ブロック線図を等価変換して得られる伝達関数の分子には主軸要素の積のみが現れることは自明であるので、演習問題として出題するのは分母の特性式とした。

3．本研究で課題とした事項

(1) ブロック線図の描画と答え合わせについては、細かい作りこみを除けばプログラム作成に関して解決すべき課題はなかった。

○　プログラムの全体構成については　【VII．計算機による等価変換の問題演習教材】参照。

(2) システム作成で課題としたのは、問題として出題されるさまざまなパターンのループ構成をいかに計算機に認識させるかという点であり、これができないと出題されたブロック線図に対応する解答を計算機内部で自動的に生成することができない。

(3) 代案として、個々のブロック線図に対応する正解を予め手で計算しておき、計算機に回答を記憶させておく方法もあるが、問題数が多くなると出題者が正解を作成し、計算機に入力して確認する負担が大きくなり現実的な方策とは言えなくなってくる。

(4) 調査した範囲では、ランダムに出題したブロック線図の文字式による等価変換を扱った先例は見当たらなかった。グラフ理論等を応用してループ構成を判定させることも考えられるが、具体的なプログラム化はかなり難しいと思われた。先行する例がないのは、このようなことも一因になっているのではないかと推測される。　(次ページの調査結果の概略参照)

(5) このため、本研究ではできるだけ簡単で、わかりやすいプログラムにより、正解を効率的に作成できることを目標に試作を行いシステムが実用化できるかどうか確認を行った。

（4－1）調査結果の概略

調査した範囲では、計算機を使用して伝達要素を文字式で表現してブロック線図の等価変換の演習問題出題を行うソフトウエアは見当たらなかった。

以下の①－③は解析ツールであり、演習問題を出題する構造にはなっていない。
また、④は教育ソフトであるが、ブロック線図の等価変換をランダムに出題して回答する方式ではない。⑤も同様。いずれの場合も、入力したブロック線図に対する伝達関数は計算するが、計算機がブロック線図を出題し、対応する正解を内部で生成する方式ではなかった。

①MATLAB；mathworks社
- 制御系に特化したシミュレーションツール
- s＝jωを使った多項式によるブロック線図の等価変換 はできるが、伝達要素を文字式で表した等価変換はできない・・・参考文献2に明記
- 商用であり高価、使用するには熟練が必要

②Scilab；INRIA（フランス国立コンピュータ科学・制御研究所）
- 多目的に活用可能なシミュレーションツール
- s＝jωを使った多項式によるブロック線図の等価変換はできる
- インターネットからダウンロードして無料で利用可能、使用するには熟練が必要

③Mathematica；
- 画面上に制御ブロックを作図して文字式で表した伝達要素を使った等価変換が可能
- ブロック線図の等価変換を問題として出題する場合はプログラム開発が必要と考えられる
- 商用であり高価、使用するには熟練が必要

④Windowsを使って制御工学演習・・・参考文献4
- 画面上に制御ブロックを作図して文字式で表した伝達要素を使った等価変換が 可能
- ブロック線図の等価変換を問題として出題する場合はプログラム開発が必要と考えられる
- 市販されているがMATLABなどに比較するとそれほど高価ではない、比較的簡単に使用可能と思われる（購入していないので解説書から推定）

⑤その他、インターネット上で自動制御関係の講座を検索した結果、電子回路の解析等に用いられるシミュレーションツールである Circuit Maker を使って計算演習を行わせる方式を採用した宮崎技術研究所のサイトがあったが、これはボード線図などを実際に数値計算することに重きを置いたものであった。

4．本研究では、ループの出発点とループ距離を用いてループ構造を特定することを試みた。

出発点と到達距離はループの構造を規定する変数であり、以下のように設定している。
○ 【0．言葉の定義】参照

(1) 主軸要素数とフィードバックループ数
- o 第3種電気主任技術者試験より若干難しい、主軸要素数＝3、ループ数＝3を上限とした。
- o フィードバックループ数は主軸要素数に等しくなるよう出題し、主軸要素数＝1～3の問題を出題することとした。

(2) フィードバックループの生成数
主軸要素数と同数であるので i で番号付を行った。
これは、フィードバックループが描画される段数にも対応している。
- o 主軸要素数＝1の場合　i＝1　(フィードバックループ数＝1)
- o 主軸要素数＝2の場合　i＝1，2　(フィードバックループ数＝2)
- o 主軸要素数＝3の場合　i＝1，2，3　(フィードバックループ数＝3)

(3) 出発点の位置
フィードバックループの番号 i に基づき st(i) として設定した。
- o 主軸要素数＝1の場合　st(i) ＝1　(出発点は位置 1 のみ)
- o 主軸要素数＝2の場合　st(i) ＝1，2　(出発点は位置 1と 2)
- o 主軸要素数＝3の場合　st(i) ＝1，2，3　(出発点は位置 1 と 2 と 3)

(4) ループ距離
フィードバックループの番号 i に基づき lo(i) として設定した。
出発点位置st(i) より大きなループ距離lo(i) は存在しないので、設定可能なループ距離は以下のようになる。
- o st(i) ＝1の場合　lo(i) ＝1　(出発点1からは距離＝1のループのみ可能)
- o st(i) ＝2の場合　lo(i) ＝1，2　(出発点2からは距離＝1と2のループが可能)
- o st(i) ＝3の場合　lo(i) ＝1，2，3　(出発点3からは全ての距離のループが可能)

5．出発点 st(i) とループ距離 lo(i) を直接使って構造を特定する方法

(1) 行列を使った計算で述べたように、特性式の分母はブロック線図の形に合わせて行列の2行2列の要素の積や和をとれば計算できる。

(2) 例えば距離＝1のループが3重になっているような単純なループ構成であれば、以下で示すように、ループの形を、ループ距離と出発点により判定し、2行2列の要素の和をとることで比較的容易にループ構造の特定と対応する特性式の分母を表す式を書くことができる。

［プログラムの例］

```
If lo（i） = 1 & lo（2） = 1 & lo（3） = 1 Then        ･･･ ループ距離は全て1

If st(1) = st(2) Then                                   ･･･ 出発点は全て同じ
If st(1) = st(2) Then

For i = 1 To 3
buf（i） = “G” & st（i） & “H” & i & st（i）          ･･･ 1段目から順にフィードバック特性を記述
OUTPUT = OUTPUT & “+” & buf（i）                      ･･･ フィードバック特性を加算
Next  i
End If
End If

Picture1.FontSize = 10
Picture1.CurrentY = 4500
Picture1.CurrentX = 4100
Picture1.Print “OUTPUT=”; 1 & OUTPUT   ･･･ 1＋ΣGiHi  の形で出力

End If
```

（3）上のような方式で、原理的にはループ構造に特性式を結び付けられることが分かった。

（4）一方、フィードバックループの直列結合と並列結合が併存する場合、特に2つのループの出発点と到達点が交差する場合は、どのループから先計算するかを順序付けをしなければならない。

（5）これに加えて、上記のように　buf（i） = “G” & st（i） & “H” & i & st（i）　として主軸要素の出発点を変数で表現した形にしておくと、距離＝2と距離＝3のループの特性式は

- “G” &（st（i）－1） & “G” & st（i） & “H” & i & st（i）　式-1
- “G” &（st（i）－2） &“G” &（st（i）－1） & “G” & st（i） & “H” & i & st（i）　式-2

であるため、距離＝2　または　距離＝3　のループの有無をその都度判定してフィードバック特性を式-1、式-2の形に書いてやる必要があること（特にループ距離＝2の場合は、出発点が2と3の2つのケースがあるので、これも判定しなければならない）から、ループ構造の判別が非常に複雑になった。

（6）このため、ループ構造と特性式の結びつけが分かりにくく、プログラム作成と検証手間がかかるので、シンプルでループ構造と特性式の結びつけが分かりやすいプログラムを作成することとし、再度検討を行った。

6．再検討

(1) 見直しの観点

前述の方法ではループ構造の判定が複雑なので、これを簡素化するため次の観点で見直しを行った。

a．ブロック線図は、基本的にはフィードバックループの直列結合と並列結合で構成されており、直列 結合の場合は、2つのループは独立して存在し、並列結合の場合は、一方のループが他方を包含する関係になる。

b．2つのループが独立している場合、特性式の分母は等価変換したフィードバックループの行列の2行2列の要素の積で表わすことができ、2つのループが包含関係にあるときは、同じく2行2列の要素の和で表すことができるので、ブロック線図上に生成するループ相互の包含関係が判別できれば、ループの形（構造）と特性式の分母の形との関連付けが可能になる。

7．出発点とループ距離による包含関係の判別

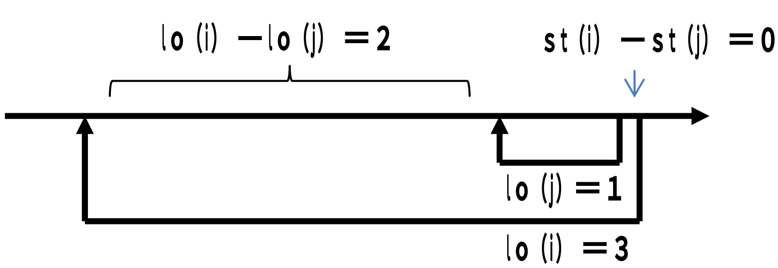
(1) 2つのループをループ ｉ とループｊ として、出発点とループ距離の概念を使い包含関係の判別方法を検討した。

(2) ループ ｉ が ループｊ を包含している場合

a．出発点の位置の差 st (i) － st (j) がループ距離の差 lo (i) －lo (j) と等しいか小さければループ ｉ がループ ｊ を包含する状態となる。

b．従って st (i) －st (j) ＜＝lo (i) －lo (j) が成立すれば、ループ ｉ が ループｊ を包含していると判定できる。

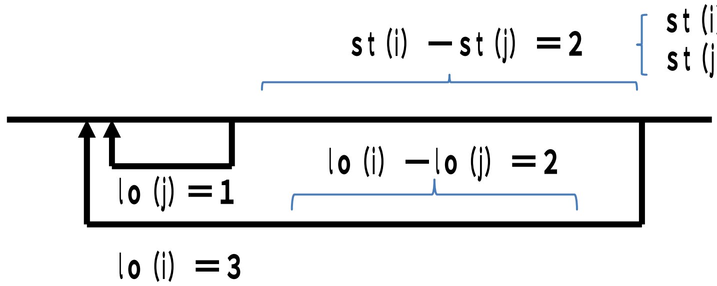
o 例（出発点が同一で到達点が異なる場合）



st (i) ＝3
st (j) ＝3

st (i) －st (j) ＝0
lo (i) －lo (j) ＝2
∴st (i) －st (j) ＜＝lo (i) －lo (j)

o 例（出発点が異なり到達点が同一の場合）



st (i) ＝3
st (j) ＝1

st (i) －st (j) ＝2
lo (i) －lo (j) ＝2
∴st (i) －st (j) ＜＝lo (i) －lo (j)

(3) ループ i が ループ j と独立している場合

a．ループ i の出発点がが ループ j の出発点より大きいと仮定すると、ループ i の到達点が ループ j の発点より大きいか等しくなる。

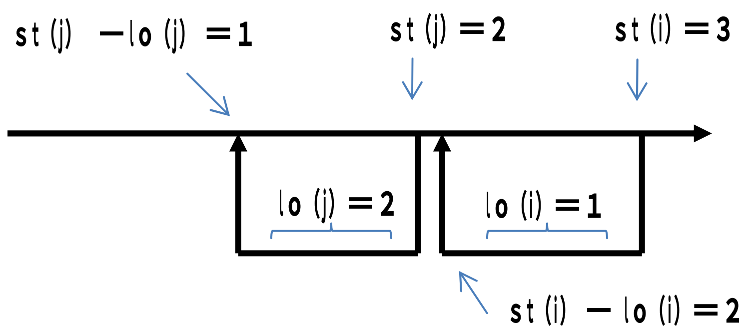
b．従って　st (i) ー lo (i) ＞＝st (j)

c．この式が成立すれば　st (i) ー lo (i) ＞st (j) ーlo (j)　が成立するので、包含の場合の判定条件との整合性を考えて、独立の判定も包含の場合と同じ式を用いることとし

st (i) ー lo (i) ＞st (j) ーlo (j)　即ち　st (i) ーst (j) ＞lo (i) ーlo (j)

が成立すれば、ループ i は ループ j と独立していると判定できる。

o　例示



st (i) ーst (j) ＝1
lo (i) ーlo (j) ＝ー1
∴st (i) ーst (j) ＞lo (i) ーlo (j)

(4) なお、2つのループの出発点とループ距離が同一で、お互いに重なりあっている状態は包含に含むものとする。

8．これらの関係を基に、以下のように分類指標を定義した。

(1) 2つのフィードバックループの出発点の位置の差
- o　ds12＝st (1) ーst (2)　　(主軸要素数＝2の場合はこの指標のみ)
- o　ds23＝st (2) ーst (3)
- o　ds31＝st (3) ーst (1)

(2) 2つのフィードバックループ距離の差
- o　dl12＝lo (1) ーlo (2)　　(主軸要素数＝2の場合はこの指標のみ)
- o　dl23＝lo (2) ーlo (3)
- o　dl31＝lo (3) ーlo (1)

(3) 出発点の位置の差 (ds) とループ距離の差 (dl) の差分
- o　Det1＝ds12ーdl12　　(主軸要素数＝2の場合はこの指標のみ)
- o　Det2＝ds23ーdl23
- o　Det3＝ds31ーdl31

(4) 従って2つのループが包含関係にある場合は、例えばDet1＜＝0、独立の関係にある場合は、Det1＞0　となり、Det i の符号で包含関係を判定できる。

9．主軸要素数＝2の場合のループの形の判別

(1) 主軸要素数=2の場合について指標によるループの形の判別を行ってみた。この場合、ループの数は2となるので、判別は1回で済む。

(2) 1つのループについて生成される図形のパターンは添付図－4に示すように 3種類あるので、これらを組み合わせた合計9パターンのブロック線図が生成される。これら9パターンのブロック線図について、構造定数st(i)、lo(i) とそれに対応する指標 Det1 の符号および2つのループの包含・独立の関係を確認し、表VI－1にまとめた。

(3) その結果

a．2つのループが直列し、相互に独立している場合
o Det1＞0 ・・・ パターン2，4

b．2つのループが並列し、一方のループが他方のループを包含している場合
o Det1＜＝0 ・・・ パターン1，3，5，6，7，8，9

と正しく判別できることが確認できた。

(4) また、生成される9つのパターンの特性式の分母を計算するとVI－9に示す通りとなるが、これを文字AとBを使って以下のように整理した。

a．ブロック線図の形が独立の場合 (パターン2，4) ・・・ (1＋A) (1＋B) 型
b．ブロック線図の形が包含の場合 (パターン1，3，5，6，7，8，9) ・・・ 1＋A＋B 型

(5) ブロック線図の形が包含関係を使って判別できれば、対応する特性式の分母の形も1＋A＋Bか (1＋A) (1＋B) に決める事ができるので、出発点とループ距離によって定まる個々のAとBの式を分母の形に代入すれば、生成されたブロック線図の特性式の分母を計算できる。
このことによって、複雑な判別条件を設定しなくても、ブロック線図の形とそれに対応する特性式の分母の形を結びつけられる可能性が出てきたので、主軸要素数=3の場合についてこの方法が適用できるかどうか検討を行った (次章に記載) 。

表VI－1

パターン	st(1)	lo(1)	st(2)	lo(2)	Ds12	dl12	Det1	ブロック線図の形	特性式の分母の形
1	1	1	1	1	0	0	0	包含	1＋A＋B
2 *	1	1	2	1	1	0	1	独立	(1＋A) (1＋B)
3 *	1	1	2	2	1	1	0	包含	1＋A＋B
4	2	1	1	1	1	0	1	独立	(1＋A) (1＋B)
5	2	1	2	1	0	0	0	包含	1＋A＋B
6 *	2	1	2	2	0	1	-1	包含	1＋A＋B
7	2	2	1	1	1	1	0	包含	1＋A＋B
8	2	2	2	1	0	1	-1	包含	1＋A＋B
9	2	2	2	2	0	0	0	包含	1＋A＋B

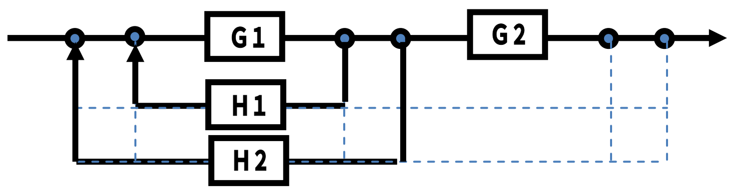
次ページ以降に、ここで用いたブロック線図の形と対応する特性式の計算結果を記載する。

(6) ブロック線図の形と分類指標の符号

- ｉ＝１の場合をループ１、ｉ＝２の場合をループ２と呼ぶ。
- ループ１とループ２が包含関係（並列）にあるか独立（直列）かについては以下の図の通りとなり、
  - 包含関係にあるときは　Ｄｅｔ１＜＝０
  - 独立の場合は　　　　　Ｄｅｔ１＞０　　　となっている。

パターン１

Ｄｅｔ１＜＝０
包含

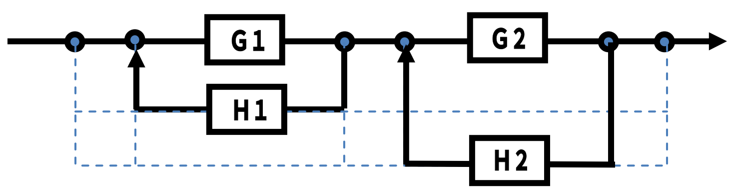


構造定数と指標
ｓｔ（１）＝１　ｌｏ（１）＝１
ｓｔ（２）＝１　ｌｏ（２）＝１
ｄｓ１２＝０　ｄｌ１２＝０
Ｄｅｔ１＝０

特性式の分母　１＋Ｇ１Ｈ１＋Ｇ１Ｈ２　→　１＋Ａ＋Ｂ型

パターン２

Ｄｅｔ１＞　０
独立

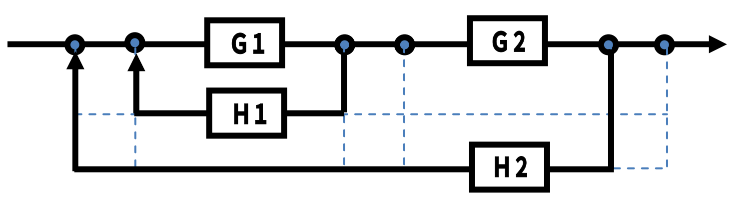


構造定数と指標
ｓｔ（１）＝１　ｌｏ（１）＝１
ｓｔ（２）＝２　ｌｏ（２）＝１
ｄｓ１２＝１　ｄｌ１２＝０
Ｄｅｔ１＝１

特性式の分母　（１＋Ｇ１Ｈ１）（１＋Ｇ２Ｈ２）　→　（１＋Ａ）（１＋Ｂ）型

パターン３

Ｄｅｔ１＜＝０
包含

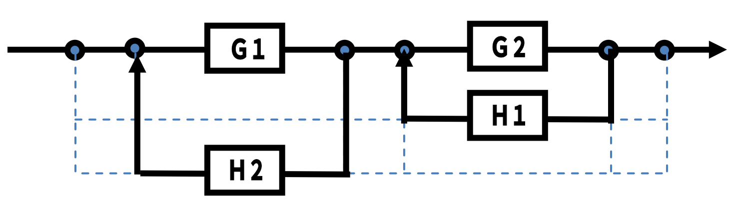


構造定数と指標
ｓｔ（１）＝１　ｌｏ（１）＝１
ｓｔ（２）＝２　ｌｏ（２）＝２
ｄｓ１２＝１　ｄｌ１２＝１
Ｄｅｔ１＝０

特性式の分母　１＋Ｇ１Ｈ１＋Ｇ１Ｇ２Ｈ２　→　１＋Ａ＋Ｂ型

パターン４

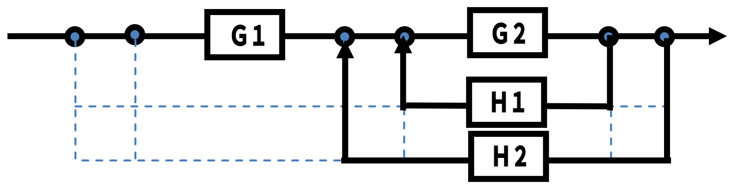
Ｄｅｔ１＞　０
独立



構造定数と指標
ｓｔ（１）＝２　ｌｏ（１）＝１
ｓｔ（２）＝１　ｌｏ（２）＝１
ｄｓ１２＝１　ｄｌ１２＝０
Ｄｅｔ１＝１

特性式の分母　（１＋Ｇ１Ｈ２）（１＋Ｇ２Ｈ１）　→　（１＋Ａ）（１＋Ｂ）型

パターン5

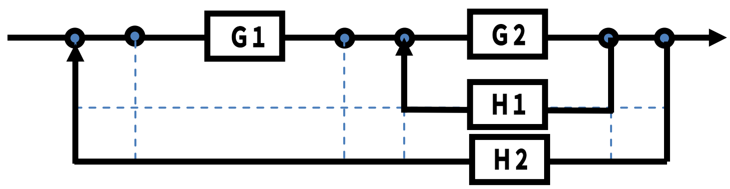


構造定数と指標
st（1）＝2　lo（1）＝1
st（2）＝2　lo（2）＝1
ds12＝0　dl12＝0
Det1＝0

特性式の分母　1＋G2H1＋G2H2　→　1＋A＋B型

パターン6

Det1＜＝0
包含

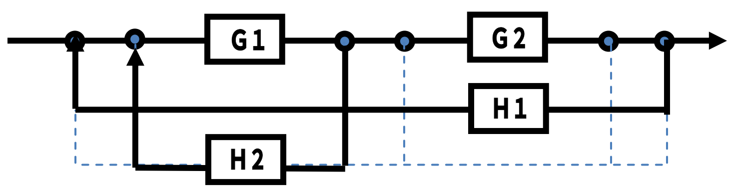


構造定数と指標
st（1）＝2　lo（1）＝1
st（2）＝2　lo（2）＝2
ds12＝0　dl12＝1
Det1＝－1

特性式の分母　1＋G2H1＋G1G2H2　→　1＋A＋B型

パターン7

Det1＜＝0
包含

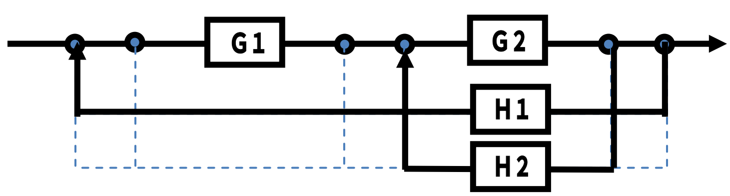


構造定数と指標
st（1）＝2　lo（1）＝2
st（2）＝1　lo（2）＝1
ds12＝1　dl12＝1
Det1＝0

特性式の分母　1＋ G1H2 ＋G1G2H1　→　1＋A＋B型

パターン8

Det1＜＝0
包含

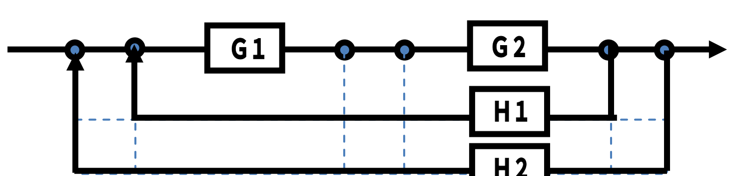


構造定数と指標
st（1）＝2　lo（1）＝2
st（2）＝2　lo（2）＝1
ds12＝0　dl12＝1
Det1＝－1

特性式の分母　1＋G2H2＋G1G2H1　→　1＋A＋B

パターン9

Det1＜＝0
包含



構造定数と指標
st（1）＝2　lo（1）＝2
st（2）＝2　lo（2）＝2
ds12＝0　dl12＝0
Det1＝0

特性式の分母　1＋G1G2H1＋G1G2H2　→　1＋A＋B型

(7) ブロック線図に対応した特性式の計算

パターン1 (包含)

$$\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}=\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1+G1H2\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H1+G1H2\end{pmatrix}$$

特性式の分母　1＋G1H1＋G1H2
→　1＋A＋B型

パターン2 (独立)

$$\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1+G2H2\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & (1+G1H1)(1+G2H2)\end{pmatrix}$$

特性式の分母　(1＋G1H1) (1＋G2H2)　→　(1＋A) (1＋B) 型

パターン3 (包含)

$$\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}=\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H1+G1G2H2\end{pmatrix}$$

特性式の分母　1＋G1H1＋G1G2H2
→　1＋A＋B型

パターン4 (独立)

$$\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1 & 0\\ 0 & 1+G1H2\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1+G2H1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & (1+G1H2)(1+G2H1)\end{pmatrix}$$

特性式の分母　(1＋G1H2) (1＋G2H1)　→　(1＋A) (1＋B) 型

パターン5 (包含)

$$\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\left[\left\{\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right]=\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\left\{\begin{pmatrix}G2 & 0\\ 0 & 1+G2H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}$$

$$=\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1+G2H1+G2H2\end{pmatrix}=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G2H1+G2H2\end{pmatrix}$$

特性式の分母　1＋G2H1＋G2H2
→　1＋A＋B型

パターン6（包含）

$$\left[\left\{\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}=\left\{\begin{pmatrix}G2 & 0\\ 0 & 1+G2H1\end{pmatrix}\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G2H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G2H1+G2H2\end{pmatrix}$$

特性式の分母　1＋G2H1＋G2H2
→　1＋A＋B型

パターン7（包含）

$$\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}=\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H2\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H2\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H2+G1G2H1\end{pmatrix}$$

特性式の分母　1＋G1H2＋G1G2H1
→　1＋A＋B型

パターン8（包含）

$$\left[\left\{\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}=\left\{\begin{pmatrix}G2 & 0\\ 0 & 1+G2H2\end{pmatrix}\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G2H2\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G2H2+G1G2H1\end{pmatrix}$$

特性式の分母　1＋G2H2＋G1G2H1
→　1＋A＋B

パターン9（包含）

$$\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}=\left\{\begin{pmatrix}G1G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1G2H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1G2H1+G1G2H2\end{pmatrix}$$

特性式の分母　1＋G1G2H1＋G1G2H2
→　1＋A＋B型

１０．主軸要素数＝3の場合のブロック線図の形の判別

(1) 主軸要素数＝2の場合と同様、ブロック線図の構造を指標Detの符号で分類し、対応する特性式の形がどのようになるか確認した。

(2) 主軸要素数＝3の場合に生成されるブロック線図のパターンは２１６通りあり、全てを 記載すると膨大になるので結果のまとめと代表的なパターンを示す。(1つのループについて生成される図形のパターンは添付図－5に示すように 6種類あり、これらを組み合わせた合計２１６パターンのブロック線図が生成される。)

(3) 主軸要素数＝2の場合にはループが2つしかないので、ループ同士の関係を表す指標は1つとなるが、主軸要素数＝３の場合はループが3つになるので、ループ同士の関係を表す指標はDet1、Det2、Det3の３つ必要となる。

(4) 指標については、要素数＝2の場合同様、
- o Det[i]＜＝0の場合は2つのループが包含関係にある
- o Det[i]＞0の場合は2つのループが独立の関係にある

と定義して、ブロック線図の構造の特定に適用することができる。

(5) なお、構造の特徴をより明確にするため、包含関係にあるループのうち、出発点が 等しく、かつループ距離も等しい場合 (到達点も等しい場合) を包含関係と区別して表示した。

(6) 表中に＜ 0、＝0、＞0 を記載すると見にくいので、
- o Det[i] ＜ 0 の場合を ○･･･2つのループが包含関係にある場合
- o Det[i] ＝ 0 の場合を ◎･･･包含関係のうち出発点とループ距離が等しい場合
- o Det[i] ＞ 0 の場合を △･･･2つのループが独立の関係にある場合

として表示した。

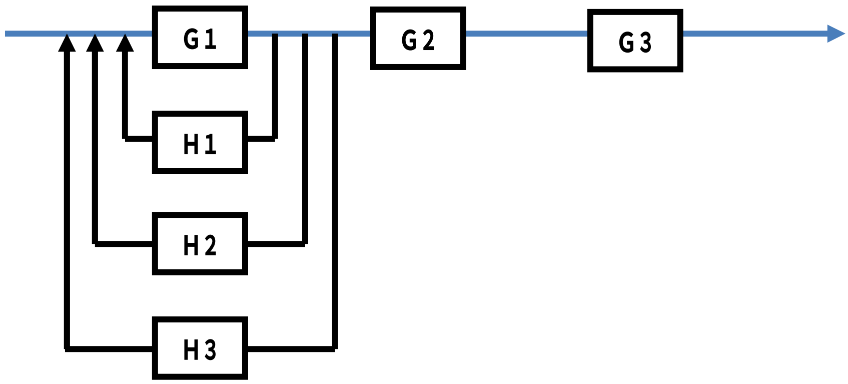
(7) その結果、表VI－2に示すように、２１６パターンのブロック線図は１５種類の構造的特徴に分類されることがわかった。次ページ以降に、ブロック線図の形と対応する特性式の計算結果を記載する。

表VI－2

No	Det1	Det2	Det3	パターン数
1	◎	◎	◎	6
2	○	○	○	24
3	◎	○	○	17
4	○	◎	○	17
5	○	○	◎	17
6	△	○	○	18
7	○	△	○	18
8	○	○	△	18
9	○	△	△	12
10	◎	△	△	12
11	△	○	△	12
12	△	◎	△	12
13	△	△	○	12
14	△	△	◎	12
15	△	△	△	9
			合計	216

(8) １５種類の構造的特徴のそれぞれについて代表的な形のブロック線図を示す。特性式の型は次の計算のところに示す。

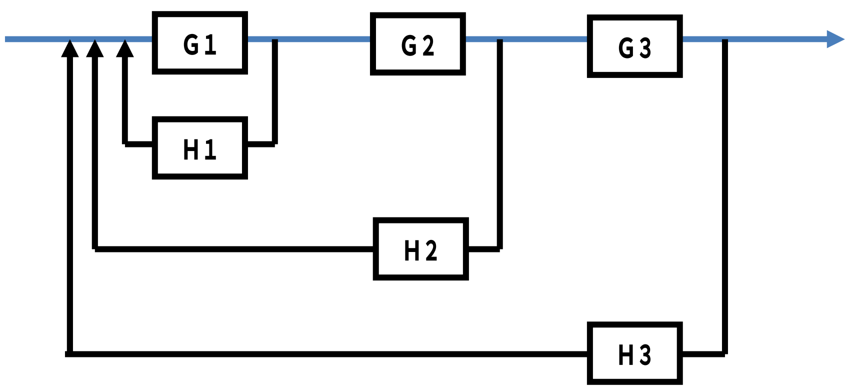
No.1 ◎◎◎ (6パターン)



st (1) =1 lo (1) =
1st (2) =1 lo (2)
=1st (3) =1 lo
(3) =1
ds12=0 dl12=0
ds23=0 dl23=0
ds31=0 dl31=0
Det1=0 ◎
Det2=0 ◎
Det3=0 ◎

特性式の分母 $1+G1H1+G1H2+G1H3$

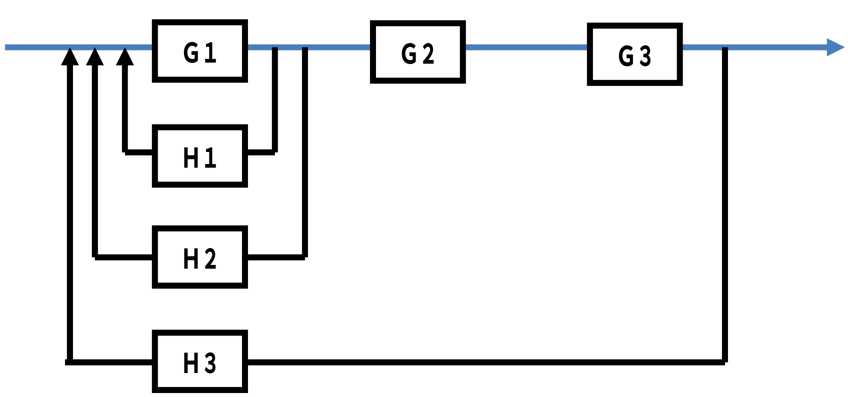
No.2 ○○○ (24パターン)



st (1) =1 lo (1) =
1st (2) =2 lo (2)
=2st (3) =3 lo
(3) =3
ds12=1 dl12=1
ds23=1 dl23=1
ds31=2 dl31=2
Det1=0 ○
Det2=0 ○
Det3=0 ○

特性式の分母 $1+G1H1+G1G2H2+G1G2G3H3$

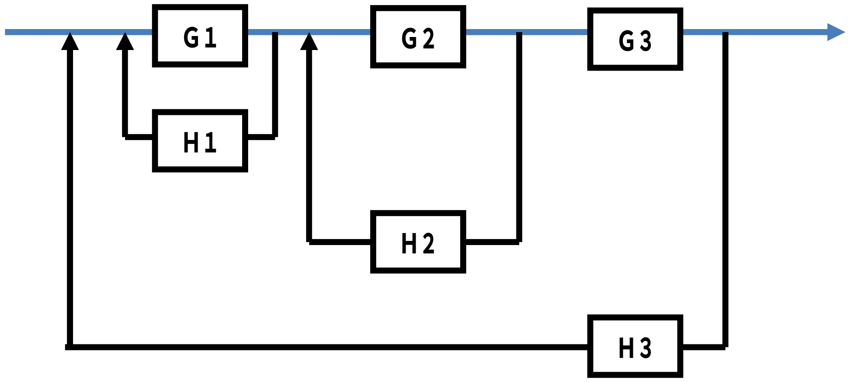
No.3 ◎○○ (No. 4：○◎○、No. 5：○○◎も含む51パターン)



st (1) =1 lo (1) =
1st (2) =1 lo (2)
=1st (3) =3 lo
(3) =3
ds12=0 dl12=0
ds23=2 dl23=2
ds31=2 dl31=2
Det1=0 ◎
Det2=0 ○
Det3=0 ○

特性式の分母 $1+G1H1+G1H2+G1G2G3H3$

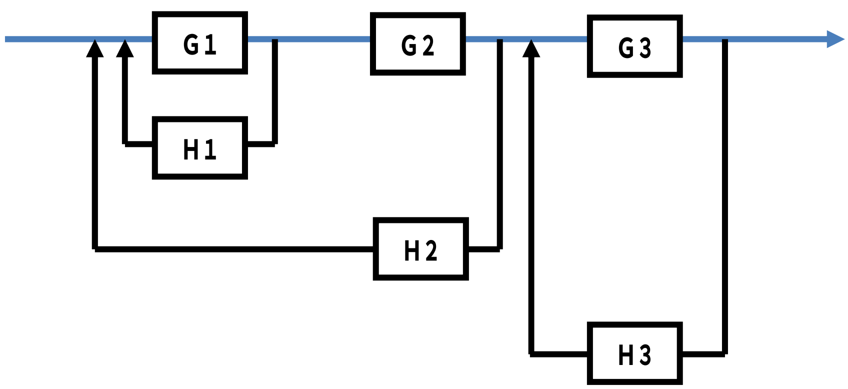
No.6　△○○　(No. 7：○△○、No. 8：○○△も含む54パターン)



st (1) ＝1　lo (1) ＝
1st (2) ＝2　lo (2)
＝1st (3) ＝3　lo
(3) ＝3
ds12＝1　dl12＝0
ds23＝1　dl23＝2
ds31＝2　dl31＝2
Det1＝1　＞0　△
Det2＝−1＜0 ○
Det3＝0　　　○

特性式の分母　$(1+G1H1)(1+G2H2)+G1G2G3H3$

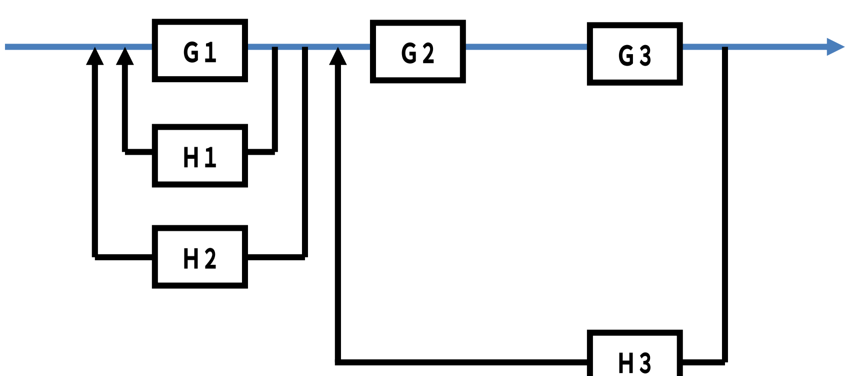
No.9　○△△　(No. 11：△○△、No. 13：△△○も含む36パターン)



st (1) ＝1　lo (1) ＝1
st (2) ＝2　lo (2) ＝2
st (3) ＝3　lo (3) ＝1
ds12＝1　dl12＝1
ds23＝1　dl23＝−1
ds31＝2　dl31＝0
Det1＝0　　　○
Det2＝2＞0　△
Det3＝2＞0　△

特性式の分母　$(1+G1H1+G1G2H2)(1+G3H3)$

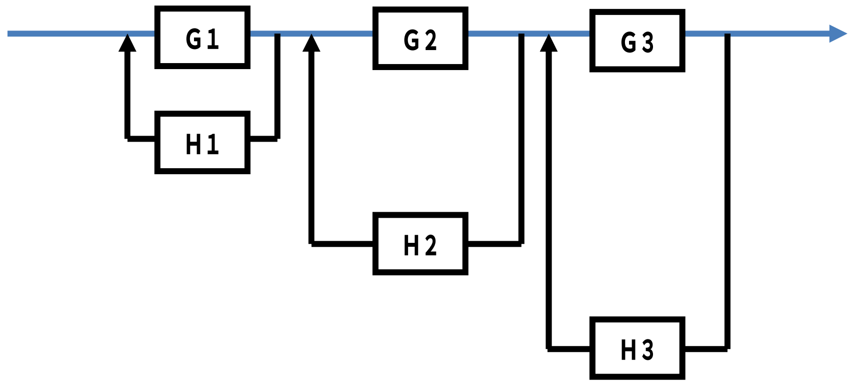
No.10　◎△△　(No. 12：△◎△、No. 14：△△◎も含む36パターン)



st (1) ＝1　lo (1) ＝
1st (2) ＝1　lo (2)
＝1st (3) ＝3　lo
(3) ＝2
ds12＝0　dl12＝0
ds23＝2　dl23＝1
ds31＝2　dl31＝1
Det1＝0　　　◎
Det2＝1＞0　△
Det3＝1＞0　△

特性式の分母　$(1+G1H1+G1H2)(1+G2G3H3)$

No.15　△△△　(9パターン)



st(1)＝1　lo(1)＝1 st(2)＝2　lo(2)＝1 st(3)＝3　lo(3)＝1
ds12＝1　dl12＝0
ds23＝1　dl23＝0
ds31＝2　dl31＝0
Det1＝1＞0　△
Det2＝1＞0　△
Det3＝2＞0　△

特性式の分母　$(1+G1H1)(1+G2H2)(1+G3H3)$

(9) 15種類の構造的特徴のそれぞれに関する代表的な形のブロック線図の特性式を計算する。
(特性式の分母の形も合わせて示す。)

No. 1 ◎◎◎

$$\left\{\begin{pmatrix}G1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}G3 & 0\\0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1 & 0\\0 & 1+G1H1\end{pmatrix}\begin{pmatrix}1 & H2\\0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2G3 & 0\\0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1 & 0\\0 & 1+G1H1+G1H2\end{pmatrix}\begin{pmatrix}1 & H3\\0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2G3 & 0\\0 & 1\end{pmatrix}=\begin{pmatrix}G1 & 0\\0 & 1+G1H1+G1H2+G1H3\end{pmatrix}\begin{pmatrix}G2G3 & 0\\0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\0 & 1+G1H1+G1H2+G1H3\end{pmatrix}$$

・・・1+A+B+C 型

No. 2 ○○○

$$\left[\left\{\begin{pmatrix}G1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\0 & 1\end{pmatrix}\right]\begin{pmatrix}G3 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\0 & 1\end{pmatrix}$$

$$=\left[\left\{\begin{pmatrix}G1 & 0\\0 & 1+G1H1\end{pmatrix}\begin{pmatrix}G2 & 0\\0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\0 & 1\end{pmatrix}\right]\begin{pmatrix}G3 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\0 & 1\end{pmatrix}$$

$$=\left[\left\{\begin{pmatrix}G1G2 & 0\\0 & 1+G1H1\end{pmatrix}\begin{pmatrix}1 & H2\\0 & 1\end{pmatrix}\right\}\begin{pmatrix}G3 & 0\\0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H3\\0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1G2 & 0\\0 & 1+G1H1+G1G2H2\end{pmatrix}\begin{pmatrix}G3 & 0\\0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H3\\0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\0 & 1+G1H1+G1G2H2\end{pmatrix}\begin{pmatrix}1 & H3\\0 & 1\end{pmatrix}=\begin{pmatrix}G1G2G3 & 0\\0 & 1+G1H1+G1G2H2+G1G2G3H3\end{pmatrix}$$

・・・1+A+B+C 型

No. 3 ◎○○

$$\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2G3 & 0\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1+G1H2\end{pmatrix}\begin{pmatrix}G2G3 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}=\begin{pmatrix}G1G2G3 & 0\\ 0 & 1+G1H1+G1H2\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\ 0 & 1+G1H1+G1H2+G1G2G3H3\end{pmatrix}$$ ･･･1＋A＋B＋C 型

No. 6 △○○

$$\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\left\{\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1+G2H2\end{pmatrix}\right\}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1G2 & 0\\ 0 & (1+G1H1)(1+G2H2)\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\ 0 & (1+G1H1)(1+G2H2)\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\ 0 & (1+G1H1)(1+G2H2)+G1G2G3H3\end{pmatrix}$$ ･･･(1＋A)(1＋B)＋C 型

No. 9 ○△△

$$\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right]\left\{\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}\right\}$$

$$=\left[\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right]\begin{pmatrix}G3 & 0\\ 0 & 1+G3H3\end{pmatrix}$$

$$=\left\{\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\begin{pmatrix}G3 & 0\\ 0 & 1+G3H3\end{pmatrix}$$

$$=\begin{pmatrix}G1G2 & 0\\ 0 & 1+G1H1+G1G2H2\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1+G3H3\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\ 0 & (1+G1H1+G1G2H2)(1+G3H3)\end{pmatrix}$$ ･･･(1＋A＋B)(1＋C) 型

No. 10 ◎△△

$$\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\left\{\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}\right\}$$

$$=\left\{\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\left\{\begin{pmatrix}G2G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}\right\}$$

$$=\begin{pmatrix}G1G2G3 & 0\\ 0 & (1+G1H1+G1H2)(1+G2G3H3)\end{pmatrix}$$ ･･･(1＋A＋B)(1＋C) 型

No. 15 △△△

$$\left\{\begin{pmatrix}G1 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H1\\ 0 & 1\end{pmatrix}\right\}\left\{\begin{pmatrix}G2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H2\\ 0 & 1\end{pmatrix}\right\}\left\{\begin{pmatrix}G3 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & H3\\ 0 & 1\end{pmatrix}\right\}$$

$$=\begin{pmatrix}G1 & 0\\ 0 & 1+G1H1\end{pmatrix}\begin{pmatrix}G2 & 0\\ 0 & 1+G2H2\end{pmatrix}\begin{pmatrix}G3 & 0\\ 0 & 1+G3H3\end{pmatrix}$$

$$=\begin{pmatrix}G1G2G3 & 0\\ 0 & (1+G1H1)(1+G2H2)(1+G3H3)\end{pmatrix}$$ ･･･(1＋A)(1＋B)(1＋C) 型

１１．主軸要素数=３の場合のブロック線図の形と特性式の形との対応関係

(1) A、B、Cを個々のループに対応する主軸要素とフィードバック要素の積とすると、ブロック線図の形と特性式の分母の形の間には以下のような対応関係があり、特性式の分母は4つの形に分類できることがわかった。

o ブロック線図の形が◎◎◎、○○○、◎○○、○◎○、○○◎の場合
(No．1，2，3，4，5)
･･･ １＋A＋B＋C
(例) １＋G１H１＋G１H２＋G１H３

o ブロック線図の形が△○○、○△○、○○△の場合 (No，6，7，8)
･･･ (１＋A) (１＋B) ＋C
(例) (１＋G１H１) (１＋G２H２) ＋G１G２G３H３

o ブロック線図の形が○△△、△○△、△△○、◎△△、△◎△、△△◎の場合
(No，9，１０，１１，１２，１３，１４)
･･･ (１＋A＋B) (１＋C)
(例) (１＋G１H１＋G１G２H２) (１＋G３H３)

o ブロック線図の形が△△△の場合 (No．１５)
･･･ (１＋A) (１＋B) (１＋C)
(例) (１＋G１H１) (１＋G２H２) (１＋G３H３)

(2) 但し、上記の対応関係から外れる場合も存在する。

a．具体的には

o ブロック線図の形が△○○、○△○、○○△の場合
･･･ 6パターンが １＋A＋B＋C の形になる。

o ブロック線図の形が○△△、△○△、△△○ の場合
･･･ 9パターンが (１＋A) (１＋B) ＋C の形になる。

o ブロック線図の形が◎△△、△◎△、△△◎の場合
･･･ 6パターンが １＋A＋B＋C の形になる。

b．これらの例外はループの出発点と到達点が交差する場合に発生している。

c．しかしながら、数的には合計２１パターンで全体の約１０％であり、発生する条件も明らかであり、特性式の形も4つのパターンの範囲内であるので、個別に判別条件を設定すれば対応可能であることから、ループの出発点とループ距離を用いてブロック線図の形と特性式の形を対応付ける方法は有効と考えられる。

１１．ブロック線図の形と特性式の形との対応関係を表にまとめると以下のようになる。

表VI－３

Ｎｏ	Det1	Det2	Det3	パターン数	対応する特性式の分母の形
1	◎	◎	◎	6	１＋Ａ＋Ｂ＋Ｃ
2	○	○	○	24	１＋Ａ＋Ｂ＋Ｃ
3	◎	○	○	17	１＋Ａ＋Ｂ＋Ｃ
4	○	◎	○	17	１＋Ａ＋Ｂ＋Ｃ
5	○	○	◎	17	１＋Ａ＋Ｂ＋Ｃ
6	△	○	○	18	(１＋Ａ) (１＋Ｂ) ＋Ｃ，　１＋Ａ＋Ｂ＋Ｃ
7	○	△	○	18	(１＋Ａ) (１＋Ｂ) ＋Ｃ，　１＋Ａ＋Ｂ＋Ｃ
8	○	○	△	18	(１＋Ａ) (１＋Ｂ) ＋Ｃ，　１＋Ａ＋Ｂ＋Ｃ
9	○	△	△	12	(１＋Ａ＋Ｂ) (１＋Ｃ)，　(１＋Ａ) (１＋Ｂ) ＋Ｃ
10	◎	△	△	12	(１＋Ａ＋Ｂ) (１＋Ｃ)，　１＋Ａ＋Ｂ＋Ｃ
11	△	○	△	12	(１＋Ａ＋Ｂ) (１＋Ｃ)，　(１＋Ａ) (１＋Ｂ) ＋Ｃ
12	△	◎	△	12	(１＋Ａ＋Ｂ) (１＋Ｃ)，　１＋Ａ＋Ｂ＋Ｃ
13	△	△	○	12	(１＋Ａ＋Ｂ) (１＋Ｃ)，　(１＋Ａ) (１＋Ｂ) ＋Ｃ
14	△	△	◎	12	(１＋Ａ＋Ｂ) (１＋Ｃ)，　１＋Ａ＋Ｂ＋Ｃ
15	△	△	△	9	(１＋Ａ) (１＋Ｂ) (１＋Ｃ)
			合計	216	

１２．まとめ

(1) 主軸要素数=２の場合同様、ブロック線図の形が包含関係を使って判別できれば、対応する特性式の分母の形も４種類に決める事ができる。

(2) 計算式の形が決まれば、出発点とループ距離によって定まる個々のＡ，Ｂ，Ｃの式を代入することにより、生成されたブロック線図の特性式の分母が計算できる。
(VIIIのフィードバックループの部品の特定と問題出題の項に方法を記述した。)

１３．以上の検討により、包含と独立の関係を使えば、ループ構造と特性式を分かりやすい形で結びつけることが可能となり、また複雑な判別条件を設定して計算順序を決めなくても、ブロック線図に対応する特性式の分母を計算することが可能である見通しがついた。

１４．この考え方に基づきブロック線図の等価変換関する演習問題出題教材プログラムを試作した。
(概略を次のVIIとVIIIに概略を述べる。)

【Ⅶ. 計算機による等価変換の問題演習教材】

【アウトライン】

１．教材試作に使う言語の選択

ＧＵＩが充実しており、効率的に画面設計ができる Visual　Basic を用いた。

２．出題方式

信州大学のＣＡＩ教材のスタイルにならい、問題を１０問連続して出題し、学習者が全て正解すれば演習が終了し、途中で間違えればそこで終了して初めからやり直す方式とした。

３．教材の使用環境

（１）最終的には社内のネットワーク上で任意に使用できることを目標としている。

（２）当面は、時間的に余裕がある自宅での学習が可能なようＣＤで配布し、自宅のパソコンにインストールきる形式とした。市販の Visual　Basic　の教科書に添付された　自由配布の Visual　Basic　 6.0 上で動作するプログラムを作成したので、誰でも自由に使用が可能である。

（３）このため、今回は社内システム上で動作させるためのサーバとの連携部分は作成していない。

【プログラムの全体構成】・・・添付図－１　　のフローチャート参照

１．合計１０題の問題を連続して出題し、問題を解くに従って易しい問題から難しい問題にレベルが変化する方式とした。

２．ブロック線図に含まれる要素数が多いほど「難しい」と考え難易度を以下のように設定した。

ａ．主軸要素数＝１・・・易しい

ｂ．主軸要素数＝２・・・標準

ｃ．主軸要素数＝３・・・難しい

３．１０問の問題の難易度は以下のように構成した。

ａ．１問目は易しいレベル

ｂ．２問目から６問目は標準レベル

ｃ．７問目から１０問目は難しいレベル

４．出題プログラムの概略構成

（１）Visual　Basic　が備えるＧＵＩである Command　ボタンを学習者が押すことにより演習が進行する設定とし、Command　ボタンは

ａ．問題を次に進めるボタン

ｂ．回答を送信するボタン

の 2 つを画面上に表示した。

（２）Visual　Basic　では Command　ボタンごとに実行するプロシージャが定義されるためプログラムは大きく分けて

ａ．次に進めるボタンには問題出題に関するプログラム部分が対応し
ｂ．解答送信ボタンには答え合わせを行うプログラム部分が対応する
構成となった。

（３）問題の進行はカウンターを用いて以下のように制御した。
ａ．問題に正解すればカウンターがカウントアップして次の問題を出題する。
ｂ．難易度の切り替えは、予め設定したカウント数になったときに対応するプログラムの部分に処理を切り替えることで実施する。
ｃ．不正解であればカウントを中止して演習を停止する。
ｄ．10 問連続して正解であれば演習は終了する。

（４）カウンターについて
ａ.プログラムは、問題出題プロシージャと答え合わせプロシージャの 2 つのプロシージャから構成されている。
ｂ．このため、カウンターは Public 変数として定義し、2 つのプロシージャから共通で参照可能として問題出題と答え合わせを連携して進行させた。
・・・Visual　Basic　特有のカウント制御

（５）問題出題部分
ａ．主軸要素数＝１の場合
o ブロック線図のパターンが１つしかないので、常に同じ問題が出題される。
o このため、画面に表示するブロック線図は固定の画像とし、正解も定数としてプログラム内部に保持した。
ｂ．主軸要素数＝２と３の場合
o ブロック線図の画面への表示
・主軸要素は常に同一なので固定した画像として表示した。
・フィードバックループについては、出発点とループ距離が異なる部品を用意し、乱数を用いてランダムに部品を選んで組み合わせ、主軸上に表示した。
・描画するループ数は主軸要素数と同数として試作した。
o 正解について
・指標Ｄｅｔ１～Ｄｅｔ３を用いてブロック線図中のループ相互の包含関係が判定し、等価変換した結果の特性式の形を特定した。
・特定した式の形に合わせて各ループのフィードバック効果を特性式に代入して正解を作成した。
o 回答欄について
・多項式である特性式の項数に対応する数の Text　Box を画面に表示させ、回答欄に表示した。
・このときに、1 つの Text　Box を空白として表示し、学習者は空白部分に対応する文字式を回答する方式とした。

（６）補足

ａ．今回は試作であったので、問題出題部分についてはバグなどの発見のしやすさも考慮してあえて難易度のレベルごとに独立にプログラムを作成した。

ｂ．このため３つのプログラムが結合する形となっており冗長な記述となっている。

ｃ．但し、ブロック線図の画面上の配置や構成を定義する変数などは各プログラムで共通としてあるので、主軸要素数＝３のプログラムは、主軸要素数＝２と主軸要素数＝１の構成を含んでおり、統合して簡素なプログラムとすることは可能である。
・・・今後統合を検討する。

（７）答え合わせ部分

ａ．学習者が入力した文字式（文字列）について、計算機が作成した正解の文字式（文字列）と照合して答え合わせする方式とした。

ｂ．文字列照合の結果
ｏメッセージボックスがポップアップして結果を表示する方式とした。

ｃ．正解の場合
ｏカウンターを１つ進めて次の問題を出題する準備をする。
ｏ次に進むボタンをクリックすると、次の問題が表示される。

ｄ．不正解の場合
ｏカウンターを停止し、演習問題を終了する。

【ブロック線図の描画】

1．予め用意した図形を部品化として組み合わせて表示することで、問題１問ごとに異なるブロック線図を描画する方法をとった。具体的には以下の通り。

（１）主軸要素は常に同じ画像なので固定的に画面に表示する

（２）フィードバックループについては、予め用意した部品（基本的な図形）の中から乱数を用いて主軸要素数と同数の部品をランダムに選択し、組み合わせて作図する

2．描画した図形の見やすさを考えて、フィードバックループの図形は主軸要素の下方（Visual Basic の Picture　Box の画面ではＹ軸方向）のみに描くこととし、図形が重複しないようフィードバックループは１つ１つＹ軸方向に一定の間隔をあけて描くこととした。
この一定の間隔に分けて描かれるフィードバックループのＹ軸方向の位置を「段」と呼ぶ。

3．主軸要素の図形の構成

（１）主軸を表す直線上に主軸要素数に応じた数のブロックの図形を配置した。

ａ．図形を表示するためのデータはプログラム中に記述した。

（２）ブロック中には、主軸要素の特性を以下のように表記した。

ａ．主軸要素数＝１の場合・・・Ｇ１

ｂ．主軸要素数＝２の場合・・・Ｇ１，Ｇ２

ｃ．主軸要素数＝３の場合・・・Ｇ１，Ｇ２，Ｇ３

4．フィードバックループの構成

（１）主軸要素数と同じ数のフィードバックループを描画することとした。

（２）１つのフィードバックループには１つのフィードバック要素を表すブロックを配置した。

（３）フィードバックループの距離は１，２，３の３パターン設定した。

ａ．主軸要素数＝１の場合
　ｏ　ループ距離＝１　のみ設定可能

ｂ．主軸要素数＝２の場合　ループの出発点が１と２の２つ設定できるので
　ｏループ距離＝１　と　２　が設定可能

ｃ．主軸要素数＝３の場合　ループの出発点が１と２と３の３つ設定できるので
　ｏループ距離＝１　と　２　と　３　が設定可能

（４）出発点の位置より大きいループ距離を設定すると、ループの到達点が図形の範囲を超過するので、ループ距離は必ず出発点の位置の数値以下となるようにした。

（５）フィードバック要素の特性は、主軸要素と区別するため Ｈ ij で表すこととした。

ａ．添字ｉはフィードバックループが配置される画面上の段数を表す。

ｂ．添字ｊはループの出発点の番号とした。

ｃ．その結果、設定できるフィードバック要素は以下の表のようになった。

主軸要素数＝1

	段数	出発点＝1
ループ距離＝1	1	H11

主軸要素数＝2

	段数	出発点＝1	出発点＝2
ループ距離＝1	1	H11	H12
ループ距離＝1	2	H21	H22
ループ距離＝2	1	―	H12
ループ距離＝2	2	―	H22

主軸要素数＝3

	段数	出発点＝1	出発点＝2	出発点＝3
ループ距離＝1	1	H11	H12	H13
ループ距離＝1	2	H21	H22	H23
ループ距離＝1	3	H31	H32	H33
ループ距離＝2	1	―	H12	H13
ループ距離＝2	2	―	H22	H23
ループ距離＝2	3	―	H32	H33
ループ距離＝3	1	―	―	H13
ループ距離＝3	2	―	―	H23
ループ距離＝3	3	―	―	H33

５．フィードバック要素の部品化

（１）主軸要素数＝２の場合

ａ．１段目に表示可能な距離＝１のループは　Ｈ１１　と　Ｈ１２　の２パターン。

ｂ．同じく１段目に段に表示可能な距離＝２のループは　Ｈ２２　の１パターン。

ｃ．２段目も同じであるので、各段には３パターンの選択肢がある。

（２）主軸要素数＝３の場合

ａ．１段目に表示可能な距離＝１のループは　Ｈ１１　と　Ｈ１２　Ｈ１３　の３パターン。

ｂ．同じく１段目に表示可能な距離＝２のループは　Ｈ１２　と　Ｈ１３　の２パターン。

ｃ．同じく１段目に表示可能な距離＝３のループは　Ｈ１３　の１パターン。

ｄ．２段目、３段目も同じであるので、各段には６パターンの選択肢がる。

（３）描画段数、出発点、ループ距離に対応したフィードバックループの図形データを作成し部品とする。

ａ．主軸要素数＝２の場合を図―　　に示す。

ｏ出発点とループ距離で３パターン定まるが、描画位置が２パターンあるので合計６パターンのデータを作成する。

ｂ．主軸要素数＝３の場合を図ー　　に示す。

ｏ出発点とループ距離で６パターン定まるが、描画位置が３パターンあるので合計１８パターンのデータを作成する。

（４）フィードバックループへの番号付け

ａ．主軸要素数を　ｉ　として、選択した出発点　ｓｔ（ｉ）とループ距離　ｌｏ（ｉ）を

ｏ　ｉｎｄｅｘ＝１００＊ｉ＋１０＊ｓｔ（ｉ）＋ｌｏ（ｉ）

として指数　Ｉｎｄｅｘ　に変換する。

・・・ｉは主軸要素数であるとともに、ループを描画する段数を定める。

ｂ．フィードバックループの部品に、予めその構成に対応した指数ｉｎｄｅｘで番号付けしておく。

ｏ主軸要素数＝２の場合６パターン

ｏ主軸要素数＝３の場合１８パターン

６．これらの部品がランダムに選ばれてブロック線図を以下のように構成する。

（１）主軸要素数＝１の場合

ａ．主軸要素　＋　１段目のフィードバックループ

（２）主軸要素数＝２の場合

ａ．主軸要素　＋　１段目のフィードバックループ　＋　２段目のフィードバックループ

（３）主軸要素数＝３の場合

ａ．主軸要素　＋　１段目のフィードバックループ　＋　２段目のフィードバックループ　＋　３段目のフィードバックループ

７．生成されるパターン数

（１）主軸要素数＝１の場合は１パターン

（２）主軸要素数＝２の場合

ａ．１段目と２段目に各３通りのパターンの選択肢があるので、これらを組み合わせると合計９通りのパターンを生成可能。

（３）主軸要素数＝３の場合

ａ．各段に６通りのパターンの選択肢があるので、３つの段を組み合わせると合計２１６通りのパターンを生成可能。

（４）合計２２６通りのパターンが生成できる問題数となる。

８．描画パターンの選択

（１）ループを規定するのは出発点とループ距離であるので、乱数で出発点とループ距離を選択する（主軸要素数＝１の場合は選択はなく常に同じ問題を出題する)。

ａ．主軸要素数＝２の場合は９通りの組み合わせの中から２パターンを選択する。

ｂ．主軸要素数＝３の場合は２１６通りの組み合わせの中から３パターンを選択する。

（２）出発点位置よりループ距離が大きい組み合わせは成立しないので、この場合は

ｏ　ループ距離　＜＝　出発点位置

になる組み合わせができるまで選択を繰り返す。

（３）主軸要素数をｉ　として、選択した出発点　ｓｔ（ｉ）とループ距離　ｌｏ（ｉ）を

ｏ　ｉｎｄｅｘ＝１００＊ｉ＋１０＊ｓｔ（ｉ）＋ｌｏ（ｉ）

として指数　Ｉｎｄｅｘ　に変換する。

９．図形の描画

（１）主軸要素は常に同一の図形として描画する。

（２）フィードバックループについては、主軸要素数に対応する数だけランダムに選択され、主軸要素の図形に追加して描画する。

（３）図形は Visual　Basic　のＧＵＩである　Form　上に設定した Picture　Box　に表示する。

【特性式の分母の作成】

１．出題したブロック線図に対して正解を作成しなければばらない。

２．主軸要素数=１の場合は１パターンしかないので予めプログラム中に定数として設定する。

３．主軸要素数=２，３の場合

（１）指標 Det1（主軸要素数＝２の場合はこの指標だけ使用)、Det2、Det3 によりループ構造が特定されると、表ー　　に示すように対応する特性式の形も特定される

（２）主軸要素数=２の場合、ブロック線図に対応する特性式の形は以下の２パターンになる。

ａ．　１＋Ａ＋Ｂ
ｂ．（１＋Ａ）（１＋Ｂ）

（３）主軸要素数=３の場合、ブロック線図に対応する特性式の形は以下の４パターンになる。

ａ．　１＋Ａ＋Ｂ＋Ｃ
ｂ．（１＋Ａ）（１＋Ｂ）＋Ｃ
ｃ．（１＋Ａ＋Ｂ）（１＋Ｃ）
ｄ．（１＋Ａ）（１＋Ｂ）（１＋Ｃ）

（４）Ａ，Ｂ，Ｃはフィードバックループの形に応じた主軸要素とフィードバック要素の積で表され、予め用意したフィードバックループの部品に対応する文字式である。

（５）例えば主軸要素数=３で､ブロック線図の形が　◎◎◎　（全てのループの出発点が等しく、また全てのループのループ距離が等しい　３つのループが重なった状態）である場合、

ａ．対応する特性式の形は　１＋Ａ＋Ｂ＋Ｃ
ｂ．ループの出発点が１であると仮定するとＡ，Ｂ，Ｃには出発点　ｓｔ（ｉ）＝１，ループ距離　ｌｏ（ｉ）＝１　に対応する部品の特性
　ｏ　Ａ＝Ｇ１Ｈ１１
　ｏ　Ｂ＝Ｇ１Ｈ２１
　ｏ　Ｃ＝Ｇ１Ｈ３１
　が対応する。
ｃ．これをＡ，Ｂ，Ｃに代入することで、出題したブロック線図に対応する特性式の分母を作成することができる。

（６）実際の出題形式は、Visual Basic の Form 上に特性式の各項に対応する Text Box が配置されているので、これらの Text Box に Ａ，Ｂ，Ｃ を表示することにより問題が出題できる（多項式としてＡ，Ｂ，Ｃの和をとらなくてもよい)。

（６）他のブロック線図の場合も、同じ方法で対応する特性式を求めることができる。

【Ⅷ. 計算機による等価変換の問題演習教材のプログラム 】の【フィードバックループの部品の特定】と【問題出題】の項を参照

【回答方式と答え合わせ】

1．回答方式

（1）Visual Basic の Form 上に回答用の Text Box と回答送信用の Command Box を設定した。

o　Command Box　は送信ボタンとなっている。

（2）学習者が問題出題用の Text Box の中で空白として表示された部分に対応する解答を回答用の Text Box に文字式（文字列）で入力し、送信ボタンを押すと、処理のプロセスが答え合わせのプロシージャに移行し、答え合わせを行う方式とした。

2．答え合わせ

（1）学習者が入力した文字式（文字列）について、計算機が生成した正解の文字式（文字列）と照合して答え合わせする方式とした。

（2）文字列照合の機能

ａ．今回は試作であるので、文字列照合の機能は最小限とした。

ｂ．入力される解答は半角大文字に限定して出題画面にその旨注記した。

ｃ．また、記述の順序も 主軸要素 G ⇒ フィードバック要素 H　の順に限定して文字列照合することとした。

（3）文字列照合に関する補足

ａ.回答は､例えばＧ１Ｈ１２など､必ずアルファベットと数字の組み合わせとなっている。

ｂ．また、文字の出現順序も

ｏ主軸要素・・・アルファベット→数字

ｏフィードバック要素・・・アルファベット→数字→数字

の順番になっている

ｃ．このため、学習者が任意の順番で主軸要素とフィードバック要素に対応する文字列を入力しても､正解と不正解を判別する機能は一般の文字列照合より容易に実現できると考えられるが、今回は試作であるので省略した。

ｄ．また、文字列照合の煩雑さを回避するため、選択肢方式で複数の正解候補を表示し学習者に選択させる方式も考えられるので、今後機能追加する。

（4）文字列照合の結果

ａ．正解となった場合

ｏメッセージボックスがポップアップし「正解です。次の問題に進んでください。」と表示する。

ｏ合わせてカウンターを１つ進め、処理を問題出題のプロシージャに移行させる。

ｂ．不正解の場合

ｏメッセージボックスがポップアップし「不正解です。正解は～～」と表示する。

ｏ合わせて、カウンターを問題数１０より大きな値としてループから抜け出し、それ以上演習が進まないようにした。

【Ⅷ．計算機による等価変換の問題演習教材のプログラム 】

【Ⅷ―１．プログラムの骨格】

Public ICount　As Integer　（Command1_Click()と Command2_Click()双方から参照可能な変数）

Private　Sub　Command1_Click()　　　　（次へのボタンに対応するプロシージャ）

ICount=ICount+1

If ICount > 10 Then　（連続１０問正解なら演習を終了、不正解の場合 ICount=1000 なので演習を終了）
Exit Sub
End If

ICount=1　　　（１問目）
主軸要素数＝１の出題

1<ICount<7　　（２問目から６問目まで）
主軸要素数＝２の出題

6<IConunt<11　（７問目から１０問目まで）
主軸要素数=３の出題

End SuB

Private　Sub　Command2_Click()　　（送信のボタンに対応するプロシージャ）

文字列照合（正解なら Command2_Click を終了し処理を Command1_Click に移行）

不正解の場合
ICount=1000　（ICount=1000 として Command2_Click を終了し処理を Command1_Click に移行）

End Sub

【Ⅷ－２．ブロック線図（主軸要素）の描画】

１．ブロック線図は、Visual　Basic　が備える　Picture　Box　に描画する。

２．基準となるのが全てのブロック線図に固定画像として描画される主軸要素。
Picture　Box への描画は、例えば
ａ．線分は　Picture1.Line (OX, OY)-(OX + 400 - R / 2, OY)
ｂ．円形は　Picture1.Circle (B1 + L + 200, OY), R
などの命令で実行される。

３．ここでは主軸要素数＝２の場合の描画データについて説明する。

（１）基準点の設定；Ｐｉｃｔｕｒｅ　Ｂｏｘでは、画面の左上の隅がＸ軸、Ｙ軸の原点になっているので、主軸要素の基準点を（Ｘ、Ｙ）＝（２０００，１０００）にとった。

```
OX = 2000
OY = 1000
L = 1000
B1 = OX + 1000
B2 = B1 + 2600
B3 = B2 + 2600
R = 50

‘（２）左端分岐点（到達）; 到達点を白抜き丸で表示、円の中心と半径および白抜きを設定
Picture1.FillStyle = 1　（白抜き表示）
Picture1.FillStyle = 1
Picture1.FillStyle = 1
Picture1.Circle (OX + 800, OY), R

'（３）左端線分
Picture1.Line (OX, OY)-(OX + 400 - R / 2, OY)
Picture1.Line (OX + 400 + R / 2, OY)-(OX + 600 - R / 2, OY)
Picture1.Line (OX + 600 + R / 2, OY)-(OX + 800 - R / 2, OY)
Picture1.Line (OX + 800 + R / 2, OY)-(B1, OY)

'（４）左箱；主軸要素Ｇ１が入るブロックを４本の線分で描画
Picture1.Line (B1, OY + 200)-(B1, OY - 200)
Picture1.Line (B1, OY + 200)-(B1 + L, OY + 200)
Picture1.Line (B1, OY - 200)-(B1 + L, OY - 200)
Picture1.Line (B1 + L, OY + 200)-(B1 + L, OY - 200)

'（５）中央分岐点（出発）; 出発点を黒丸で表示、円の中心と半径および塗りつぶしを設定
Picture1.FillStyle = 0　（黒で塗りつぶし）
Picture1.Circle (B1 + L + 200, OY), R
```

```
Picture1.FillStyle = 0
Picture1.Circle (B1 + L + 400, OY), R
Picture1.FillStyle = 0
Picture1.Circle (B1 + L + 600, OY), R
'（６）中央分岐点（到達）；到達点を白抜き丸で表示、円の中心と半径および白抜きを設定
Picture1.FillStyle = 1
Picture1.Circle (B1 + L + 1000, OY), R
Picture1.FillStyle = 1
Picture1.Circle (B1 + L + 1200, OY), R
Picture1.FillStyle = 1
Picture1.Circle (B1 + L + 1400, OY), R

'（７）中央線分（白抜き丸があるので細かく分割）
Picture1.Line (B1 + L, OY)-(B1 + L + 200 - R / 2, OY)
Picture1.Line (B1 + L + 200 + R / 2, OY)-(B1 + L + 400 - R / 2, OY)
Picture1.Line (B1 + L + 400 + R / 2, OY)-(B1 + L + 600 - R / 2, OY)
Picture1.Line (B1 + L + 600 + R / 2, OY)-(B1 + L + 1000 - R / 2, OY)
Picture1.Line (B1 + L + 1000 + R / 2, OY)-(B1 + L + 1200 - R / 2, OY)
Picture1.Line (B1 + L + 1200 + R / 2, OY)-(B1 + L + 1400 - R / 2, OY)
Picture1.Line (B1 + L + 1400 + R / 2, OY)-(B1 + L + 1600, OY)

'（８）右箱；主軸要素Ｇ１が入るブロックを４本の線分で描画
Picture1.Line (B2, OY + 200)-(B2, OY - 200)
Picture1.Line (B2, OY + 200)-(B2 + L, OY + 200)
Picture1.Line (B2, OY - 200)-(B2 + L, OY - 200)
Picture1.Line (B2 + L, OY + 200)-(B2 + L, OY - 200)

'（９）右端分岐点（出発）；出発点を黒丸で表示、縁の中心と半径および塗りつぶしを設定
Picture1.FillStyle = 0
Picture1.Circle (B2 + L + 200, OY), R
Picture1.FillStyle = 0
Picture1.Circle (B2 + L + 400, OY), R
Picture1.FillStyle = 0
Picture1.Circle (B2 + L + 600, OY), R

'（１０）右端線分（黒丸だけなので１本だけでも差し支えなし）
Picture1.Line (B2 + L, OY)-(B2 + L + 1000 - R / 2, OY)

'（１１）Ｇ２表示；ブロックの中に表示するＧ２の文字の大きさと位置を設定
Picture1.FontSize = 12
Picture1.CurrentX = B2 + L / 2 - 200
Picture1.CurrentY = OY - 100
```

```
Picture1.Print "G2"

'（１２）Ｇ１表示；ブロックの中に表示するＧ１の文字の大きさと位置を設定
Picture1.FontSize = 12
Picture1.CurrentX = B1 + L / 2 - 200
Picture1.CurrentY = OY - 100
Picture1.Print "G1"
```

【Ⅷ－３．出発点と距離の決定】

１．主軸要素数=３の場合を例示（主軸要素数=２の場合も同様）。

２．出発点とループ距離の決定

（１）主軸要素数＝３に対応した３つのループを選び出すため、仮の出発点を stt(j)、仮のループ距離を loo(j)として、乱数で１から３までの値を設定する。

（２）ただし、ioo(j)　>　stt(j)　となることがあるので、その場合は loo(j)　<=　stt(j)　となるよう再度、乱数で値を設定する。再試行回は最大１００回まで行う設定とした。

（３）stt(j)　と loo(j)　が　loo(j)　<=　stt(j)　の条件を満たせば

ａ．st(i) = stt(j)

ｂ．lo(i) = loo(j)

として、ループの出発点と距離とする。

（４）このような方法で　３組の st(1),lo(1)と st(2),lo(2), st(3),lo(3)を設定する。

［作成したプログラムは以下の通り］

```
For   i  =  1  To  3
For j = 1   To   100

Randomize
stt(j) = Int(3 * Rnd + 1)
loo(j) = Int(3 * Rnd + 1)

If    loo(j) <= stt(j)    Then
st(i) = stt(j)
lo(i) = loo(j)
End If

Next j
Next i
```

【Ⅷ－４．ループの部品番号の決定】

１．フィードバックループの出発点と距離を設定したら、それを画面上に描画するデータ（部品）との結びつけが必要となる。

２．ここでは、ループが位置する段数をｉとして、段数ｉと出発点 st(i)とループ距離 lo(i)を

index = 100 * i + 10 * st(i) + lo(i)

という式で結びつけて３桁の整数に直し、これを描画データの番号（部品番号）とした。

［作成したプログラムは以下の通り；主軸要素数＝３の場合で例示］

```
For i = 1 To 3
index = 100 * i + 10 * st(i) + lo(i)
'Next i
```

【Ⅷ－５．フィードバックループの描画】

１．index = 100 * i + 10 * st(i) + lo(i)で選択されたフィードバックループを描画する。

２．フィードバックループの選択について
　例えば主軸要素数=３の場合、部品数が１８と多いので、フィードバックループの画像データの選択には、Visual　Basic が備えている Select　Case の機能を用いた。

３．以下に主軸要素数=３の場合の一部分を示す。
　o 部品番号　index =１３１（１段目のループで出発点＝３、ループ距離＝１）の部分のみ。

［作成したプログラムは以下の通り］

```
For i = 1 To 3　　（部品を３つ選択するので１から３までのループにしている）

index = 100 * i + 10 * st(i) + lo(i)

Select Case index　（指定された index に対応するデータを選択）

Case 131　（index＝131 に対応するデータ）

'ループの信号伝達を表す線分と矢線の描画
Picture1.Line (B3 + L + 200, OY + R / 2)-(B3 + L + 200, OY + 600)　　'path
Picture1.Line (B3 + L + 200, OY + 600)-(B3 + L, OY + 600)　　　　　'path
Picture1.Line (B3, OY + 600)-(B2 + L + 1400, OY + 600)　　　　　　'path
Picture1.Line (B2 + L + 1400, OY + 600)-(B2 + L + 1400, OY + R / 2)　'path
Picture1.Line (B2 + L + 1400, OY + R / 2)-(B2 + L + 1400 - 50, OY + R / 2 + 100) '矢印
Picture1.Line (B2 + L + 1400, OY + R / 2)-(B2 + L + 1400 + 50, OY + R / 2 + 100) '矢印

フィードバック要素の箱の描画
Picture1.Line (B3, OY + 600 + 200)-(B3, OY + 600 - 200)
Picture1.Line (B3, OY + 600 + 200)-(B3 + L, OY + 600 + 200)
Picture1.Line (B3, OY + 600 - 200)-(B3 + L, OY + 600 - 200)
Picture1.Line (B3 + L, OY + 600 + 200)-(B3 + L, OY + 600 - 200)

フィードバック要素　Ｈ１３　の表示
Picture1.FontSize = 12
Picture1.CurrentX = B3 + L / 2 - 200
Picture1.CurrentY = OY + 600 - 100
Picture1.Print "H13"

・・・・・・・・・・・・・・・・・・・・・・・・・・・・・

End Select
Next　i
```

【Ⅷ－６．フィードバックループの部品の特定】・・・添付図―６参照

１．主軸要素数=３の場合を例にすると、画面に描画するループは６種類（部品）用意している。
２．これらのループの特性は決まっている（例えば出発点が３でループ距離が２の場合　G2G3H3)。
３．従って、描画段数と出発点、ループ距離を指定すれば、対応する部品の特性を定数として特定できる。
　o Ⅵ－５で述べたような出発点を変数にして主軸要素の数をループ距離によって変えるような
　　複雑な特性の記述は不要となる。

［主軸要素数=３の場合のプログラム］

```
For i = 1 To 3        （ループが描画される段数）

If lo(i) = 1 Then        （ループ距離＝１の場合）
    If st(i) = 1 Then                    （出発点＝１であれば）
        buff(i) = "G1H" & i & "1"        （ループの特性は　Ｇ１Ｈｉ１）
    Else
    If st(i) = 2 Then                    （出発点＝２であれば）
        buff(i) = "G2H" & i & "2"        （ループの特性は　Ｇ２Ｈｉ２）
    Else
    If st(i) = 3 Then                    （出発点＝２であれば）
        buff(i) = "G3H" & i & "3"        （ループの特性は　Ｇ３Ｈｉ３）

    End If
    End If
    End If
Else
If lo(i) = 2 Then        （ループ距離＝２の場合）
    If st(i) = 2 Then                    （出発点＝２であれば）
        buff(i) = "G1G2H" & i & "2"      （ループの特性は　Ｇ１Ｇ２Ｈｉ２）

    Else
    If st(i) = 3 Then                    （出発点＝３であれば）
        buff(i) = "G2G3H" & i & "3"      （ループの特性は　Ｇ２Ｇ３Ｈｉ３）
    End If
    End If
Else
If lo(i) = 3 Then        （ループ距離＝３の場合）
        buff(i) = "G1G2G3H" & i & "3"    （ループの特性は　Ｇ１Ｇ２Ｇ３Ｈｉ３）
End If
End If
```

【Ⅷ－７．問題出題】・・・添付図２－１以下参照

１．主軸要素数=３として、ループ同士が相互に包含関係にある場合を例に説明する。

２．この場合は３つのループが並列結合している状態なので、特性式の分母は１＋A+B＋Cの形になる。

３．A，B，Cには、描画段数と出発点、ループ距離によって決まる部品の特性式が当てはまる。

４．出題は、１＋A+B＋C　の各項をそれぞれ別のテキストボックスに表示する方式で行うので、A，B，Cの式をそのまま表示すればいい。・・・その他の特性式の型については７１～７１－２ページ参照。

５．このため、当初行っていた出発点とループ距離からループ構造を判別し、対応する特性を出発点を変数にして記述していく方式に比べ、大幅にプログラムが簡略化されブロック線図の構造との対応も分かりやすくなった。

５．以下に具体的なプログラムを記載する

'==
'包含　パターン　○○○ で全て表せる（以下は全て部分集合）'
　'三重ループ　パターン　◎◎◎
　'包含と二重　パターン　◎○○　○◎○　○○◎
'==

If Det1 <= 0 And Det2 <= 0 And Det3 <= 0 Then　　（包含を表す条件）

Line1.X2 = 4700

'以下の３つの式で全て表せるはず。（プログラム中のコメントの例）

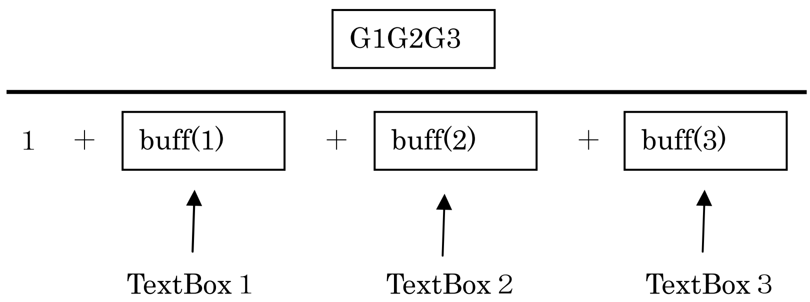
Text1.Text = buff(1)　　・・・　buff(1)の数式を TextBox１に表示

Text2.Text = buff(2)　　・・・　buff(2)の数式を TextBox2 に表示

Text3.Text = buff(3)　　・・・　buff(3)の数式を TextBox3 に表示

End If

出題画面のイメージ



End If

Next i

【Ⅷ－９．答え合わせ】

１．主軸要素数=１の場合は Text Box1 が空白となって問題を表示する。

２．主軸要素数=２、３の場合は、ここでは Text Box3 が空白となって問題を表示するものとした。

Private Sub Command2_Click()

If　ICount = 1 Then　（ICount = 1；主軸要素数=１の場合は Text Box1 に格納される文字式が正解）

ans = Text1.Text

Else

ans = Text3.Text　（主軸要素数=２，３の場合は Text Box3 に格納される文字式が正解）

End If

wordCount = 0　（変数を０に初期化する；この変数で文字列の出現回数をカウントする）

searchPosition = 1　（変数を１に初期化する；この変数で文字列の次の検索開始位置を表す）

Do While InStr(searchPosition, ans, Text15.Text) <> 0

InStr は Visual　Basic が備える文字列検索関数。正解が格納された Text Box である ans　の検索開始位置 searchPosition　から、回答が入力された Text Box15 の文字列を検索し、見つかる間は Loop までを繰り返す。

wordCount += 1　（出現回数を１つ増やす）

searchPosition = InStr(searchPosition, ans, Text15.Text) + Len(Text15.Text)

次の検索開始位置を今見つかった文字列の直後に設定する

Loop

If wordCount > 0 Then　（変数 wordCount　が０より大きければ文字列が発見されている）

MsgBox ("正解です。次の問題に進んでください。")　（正解の場合、処理を Command1_Click()に戻す）

Else

MsgBox ("不正解です。正解は";ans;”です。")

ICount = 1000　（不正解の場合 ICount = 1000　として処理を Command1_Click()に戻す）

End If

End Sub

【Ⅷ－１０．試作結果の評価】

１．次へのボタンを押すと出題が順次進行し、正解の場合は、１問目は主軸要素数=１の問題が出題され、２問目～６問目は主軸要素数=２の問題が、７問目～１０問目は主軸要素数=３の問題が正しく出題され、１１問目以降は出題されない。

２．不正解の場合は出題が停止する。

３．画面に表示されたブロック線図に対する正しい問題出題ができている。
　ａ. 設定したテキストボックスに正しくブロック線図の特性式の分母の各項が表示されている。

４．正しく答え合わせができる。
　ａ．但し、今回は試作なので文字列照合の機能を最小限としているため、既定の書式以外の解答を入力した場合は不正解となる。
　ｂ．この点については、改良がでできるので問題はない。

５．１回の演習で出題される１０問の問題のうち、7 問目以降に主軸要素数=３のパターンを出題するよう設定したが、同じ問題が出題されないよう乱数を使って出題パターンを設定した。実際に使用してみると同じ問題が続けて出ることはほとんどないので、この要請は満たしていると考えられる。

４. 以上のことから、今回の試作プログラムは実用に向けた基本的な機能は備えていると考えられる。

【Ⅷ－１１．出題画面】

１．添付図－７～９に出題画面を示す。
２．図－７が主軸要素数=２、図－８が主軸要素数=３、図－９が主軸要素数=１の場合である。
３．画面の PictureBox には出題されるブロック線図の図面が表示されている。合わせて、試作なので結果確認用のパラメータも表示している。
４．周辺の Form の部分には問題出題用のテキストボックスと回答入力用のテキストボックスを設置した。
５．また、問題を進めるための「次へ」のボタンと「回答」ボタンも設置した。

【Ⅷ－１２．備考】

１．今回の試作では、主軸要素数=３の場合、結果の比較のために

（１）ループが全て独立のパターンについては試作当初行っていた回答生成の方法のままとしてあるが、今回採用した方法は簡潔で分かりやすいので、プログラムの修正簡単に可能である。

（２）添付表―３の No.6～No.14 に示すループ距離＝２の 2 つのループの出発点と到達点が交差している場合については、標準パターンの特性式を適用したままとしているが、これも修正は即座に可能である。

２．問題の出題は、Form の部分に設けたテキストボックスに伝達関数の分母の多項式の各項を表示して行うこととしており、正式版では一部を空白として、その部分を穴埋めする形式とする予定であるが、今回は全ての項を表示している。

（１）ちなみに、主軸要素数=２の場合は１番目のテキストボックスが問題となる部分、主軸要素数=３の場合３番目のテキストボックスが出題部分となるよう設定してある。

（２）機能の検証は当該テキストボックスに表示された文字式を copy して回答用のテキストボックスに張り付け、「回答」ボタンを押すことで可能となっている。

（３）本来は出題部分は空白としておくべきであるが、空白にする処理は大変簡単で即座に実施可能である。

【Ⅷ―１３．課題】

１．今回は試作であったので、文字列照合については必要最小限の機能で作成したので、解答の入力方法に制限を付けた。しかしながら、学習者からみると入力方法に制限があるシステムは使いにくいので、入力条件を設定しそれに見合った文字列照合機能を付加することが必要と思われる。

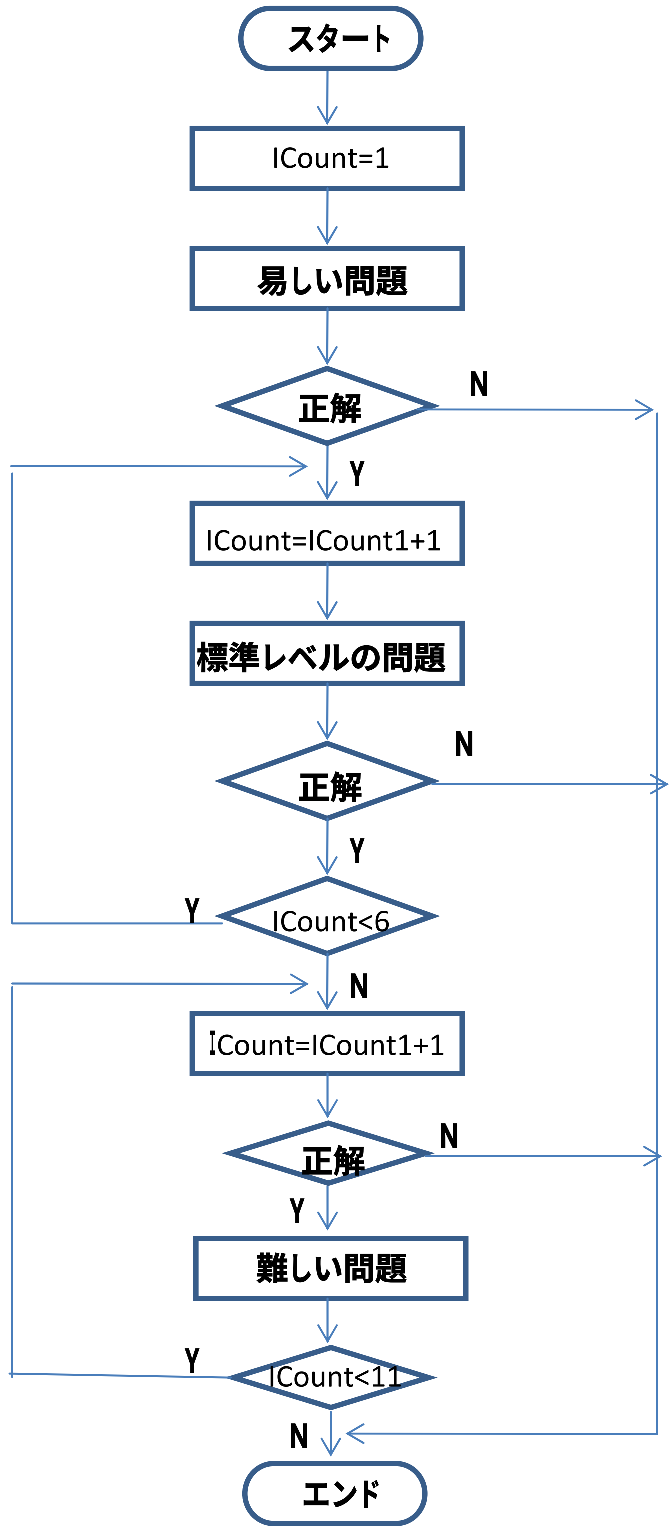
２．今回は、学習者が自宅（会社のネットワーク環境以外の場所）で学習できるよう、個人のパソコン上で動くシステムとして試作したが、他の学習ツールとの連携や会社における研修などの場での使用を想定して、ネットワーク上で稼働できるよう改良することが必要と思われる。

３．今回は、主軸要素数とフィードバック要素数が等しい形のブロック線図のみ取り扱ったが、フィードバックループにフィードバック要素がないパターンや一部のフィードバックループがないパターンなども含め、一般的な形のブロック線図の問題が出題できるよう改良することが必要と思われる。

４．今回は、答え合わせ機能を、回答に対する正解、不正解のみ表示することにとどめたが、より学習者の理解を促進するため、学習者の回答に対して、行列を使った計算方法により、正解を１段階ずつ表示していく記機能を備えることが必要と考えらる。

５．これらの事項については今後改良していきたい。
また、学習者に使ってもらって評価してもらい、改善していきたい。

プログラム全体の構成　　　　　　　　　　　　　　　　添付図ー1



演習問題出題から回答の作成までの概略プロセス　　　　添付図ー2ー1

主軸要素数=3の場合

出発点st (i) とループ距離lo (i) を乱数で3組決定

Ⅷー3参照

↓

Index＝100＊i＋10＊st (i) ＋lo (i) によりループの図形データを呼び出し、画面に描画

Ⅷー4参照

↓

st (i) とlo (i) から、対応するループ特性を生成しバッファに格納

buf (1) ＝ループ1の特性式・・・A

buf (2) ＝ループ2の特性式・・・B

buf (3) ＝ループ3の特性式・・・C

Ⅷー6参照

図ー6参照

↓

分類指標Det1~Det3によりループ構造を特定

○；2つのループが包含関係にある

◎；2つのループが重なっている (包含関係の部分集合)

△；2つのループは独立関係にある

↓

Ⅷー7参照

○◎△で表されるループ構造に対応するループ特性の式を選定

F (A, B, C)

↓

ループ特性式の式F (A, B, C) にバッファの値を代入して正解を作成

A＝buf (1)

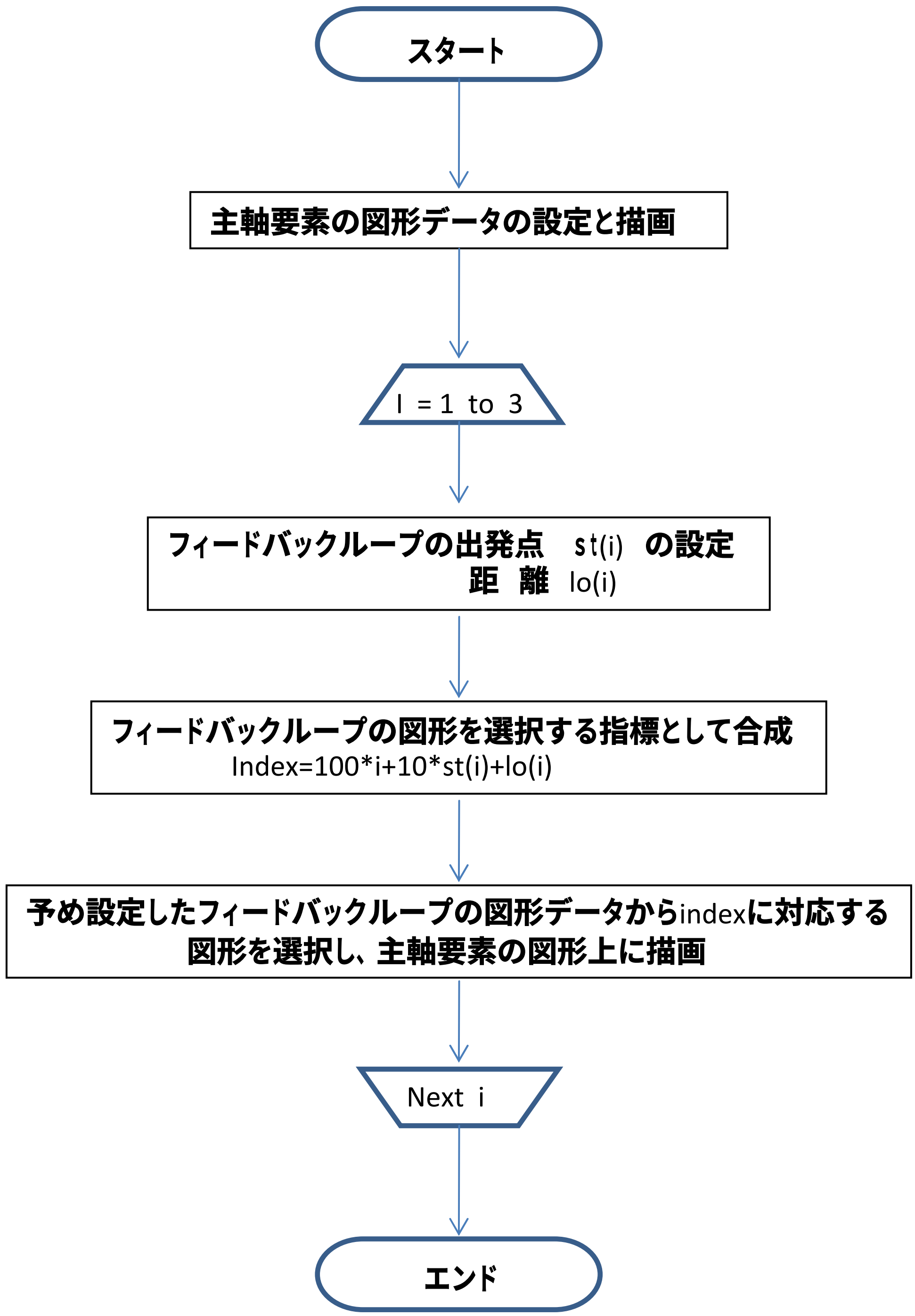
B＝buf (2)

C＝buf (3)

↓

項別に表示して問題を出題

ブロック線図の図形描画　(主軸要素数＝3の場合の例)　　　　添付図－2－2

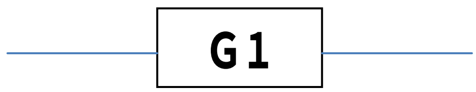


用意した図形（部品）の概念図　1

添付図ー3

主軸要素　・・・　固定画像

要素数＝1



要素数＝2

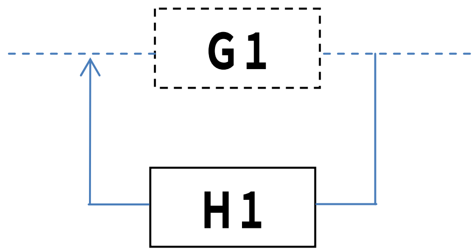
要素数＝3

用意した図形（部品）の概念図　２

添付図ー４

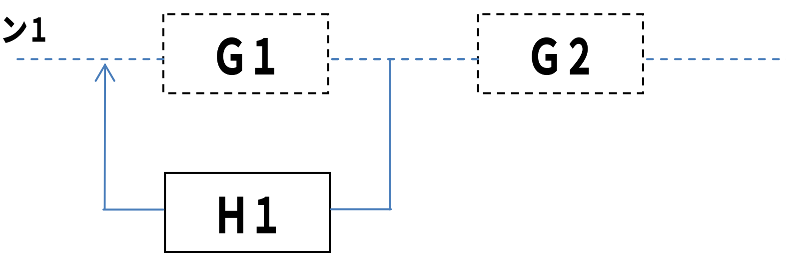
フィードバック要素　・・・　主軸要素上に重ねて描画

要素数＝１　（１種類）



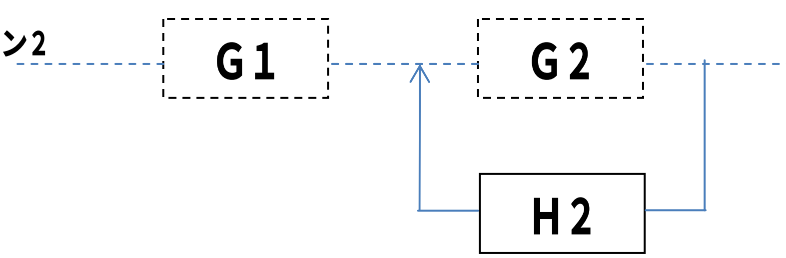
要素数＝２　（３種類）

パターン１



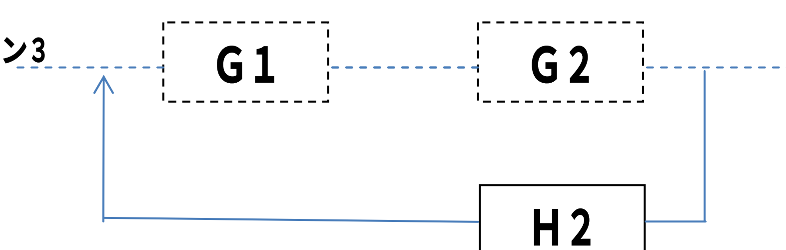
ループ距離＝１
出発点＝１

パターン２



ループ距離＝１
出発点＝２

パターン３



ループ距離＝２
出発点＝２

要素数＝3　（6種類）　　　　　　　　　　　　　　　　　　　添付図ー5

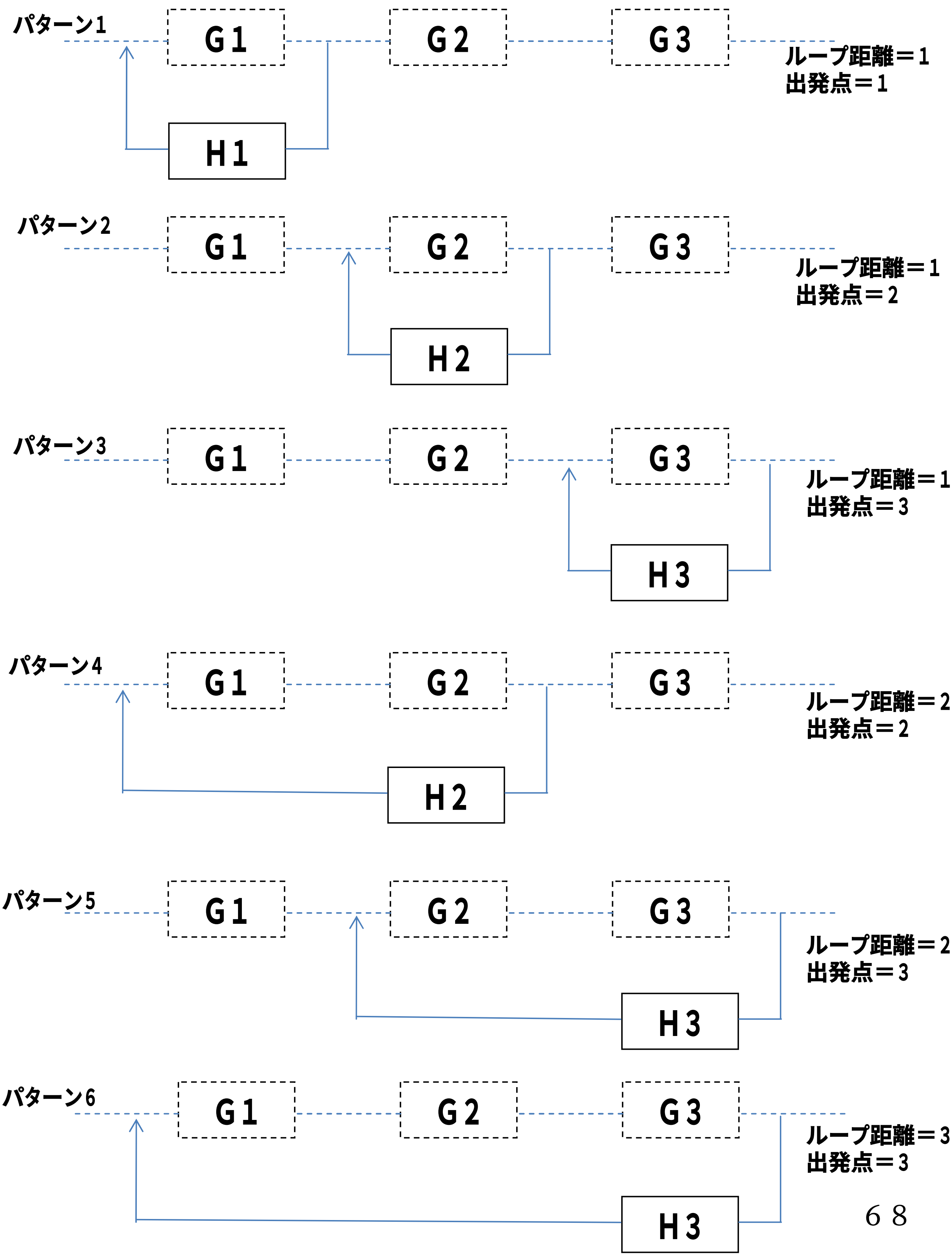


図4，5の補足

予め用意したフィードバックループの特性はループ距離lo(i)と出発点st(i)の値により次の表のように一意に決定する。

主軸要素数=2の場合（図－4参照）

パターン	lo(i)	st(i)	ループ特性
1	1	1	G1Hi1
2	1	2	G2Hi2
3	2	2	G1G2Hi2

主軸要素数=3の場合（図－5参照）

パターン	lo(i)	st(i)	ループ特性
1	1	1	G1Hi1
2	1	2	G2Hi2
3	1	3	G3Hi3
4	2	2	G1G2Hi2
5	2	3	G2G3Hi3
6	3	3	G1G2G3Hi3

i　はループが描画される上下方向の位置（段数）であり、1段目には1つ目のループが
描画され、2段目には2つ目のループ、3段目には3つ目のループが描画されるので、
描画の順番にも等しくなっている。

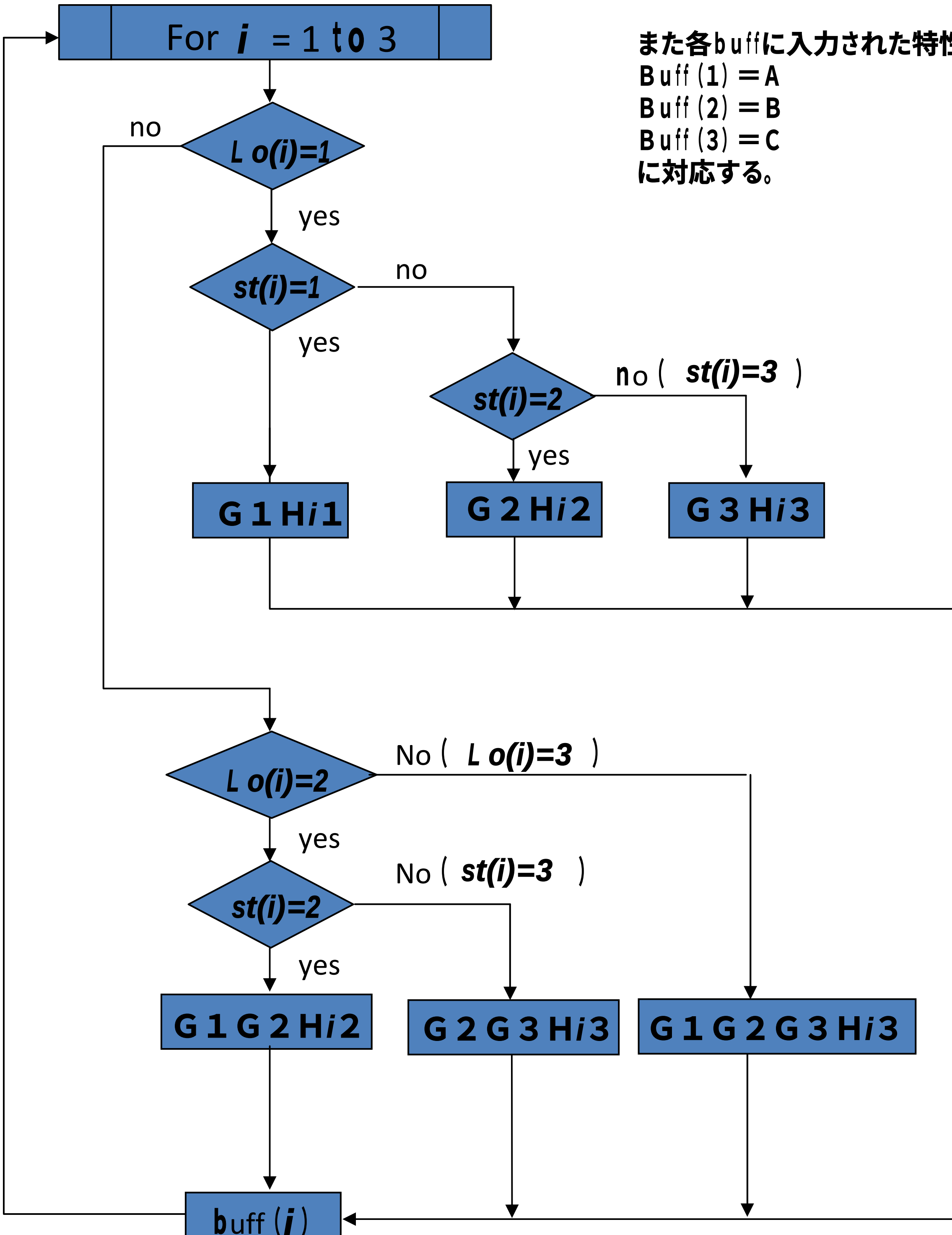
従って、主軸要素数=2のときには、乱数で選定した2つのst(i)とlo(i)によって定まる2つのループが上記主軸要素数=2の場合の表の中から選ばれて主軸上に描画される。
この場合の個々に選定されたループ特性をA，Bとする。

主軸要素数=3の場合には、乱数で選定した3つのst(i)とlo(i)によって定まる3つのループが上記主軸要素数=3の場合の表の中から選ばれて主軸上に描画される。
この場合の個々に選定されたループ特性をA，B，Cとする。

ブロック線図全体の特性は、ブロック線図の形に対応した特性式に個々のループ特性を代入することで求めることができる。

このような方式とすることにより、ループ距離が2または3の時、ループ距離と出発点の関係を判定して、ループの特性式に現れる主軸要素数を決定し、主軸要素Gの添え字とフィードバック要素Hの添え字を描画段数　i　と出発点st(i)を使って記述するなどの複雑な手続きを必要とせずに、対応するブロック線図の特性を記述することができた。

フィードバックループ特性の生成
予め部品として用意したフィードバックループのループ特性はループ距離lo (i) と出発点st (i) により一意に定められ、乱数で選定したlo (i) とst (i) を使って、以下のように書き表すことができる。



また各buffに入力された特性式は
Buff (1) ＝A
Buff (2) ＝B
Buff (3) ＝C
に対応する。

添付表－1

主軸要素数＝3の場合のパターンの詳細

		2つのループ間の関係			特性式の分母の形		特性式の分母の形		生成数
No	パターン	1－2	2－3	3－1	標準パターン	生成数	例外パターン	生成数	合計
1	○○○	包含	包含	包含	1＋ｂｕｆ(1)＋buf(2)+buf(3)	24	-	0	24
2	◎○○	重なり	包含	包含	1＋ｂｕｆ(1)＋buf(2)+buf(3)	17	-	0	17
3	○◎○	包含	重なり	包含	1＋ｂｕｆ(1)＋buf(2)+buf(3)	17	-	0	17
4	○○◎	包含	包含	重なり	1＋ｂｕｆ(1)＋buf(2)+buf(3)	17	-	0	17
5	◎◎◎	重なり	重なり	重なり	1＋ｂｕｆ(1)＋buf(2)+buf(3)	6	-	0	6
6	△○○	独立	包含	包含	(1+buf(1))(1+buf(2))+buf(3)	14	1＋ｂｕｆ(1)＋buf(2)+buf(3)	4	18
7	○△○	包含	独立	包含	(1+buf(2))(1+buf(3))+buf(1)	14	1＋ｂｕｆ(1)＋buf(2)+buf(3)	4	18
8	○○△	包含	包含	独立	(1+buf(3))(1+buf(1))+buf(2)	14	1＋ｂｕｆ(1)＋buf(2)+buf(3)	4	18
9	○△△	包含	独立	独立	(1+buf(1)+buf(2))(1+buf(3))	9	(1+buf(2))(1+buf(3))+buf(1)	3	12
10	◎△△	重なり	独立	独立	(1+buf(1)+buf(2))(1+buf(3))	10	1＋ｂｕｆ(1)＋buf(2)+buf(3)	2	12
11	△○△	独立	包含	独立	(1+buf(2)+buf(3))(1+buf(1))	9	(1+buf(1))(1+buf(2))+buf(3)	3	12
12	△◎△	独立	重なり	独立	(1+buf(2)+buf(3))(1+buf(1))	10	1＋ｂｕｆ(1)＋buf(2)+buf(3)	2	12
13	△△○	独立	独立	包含	(1+buf(3)+buf(1))(1+buf(2))	9	(1+buf(2))(1+buf(3))+buf(1)	3	12
14	△△◎	独立	独立	重なり	(1+buf(3)+buf(1))(1+buf(2))	10	1＋ｂｕｆ(1)＋buf(2)+buf(3)	2	12
15	△△△	独立	独立	独立	(1+buf(1))(1+buf(2))(1+buf(3))	9	-	0	9
				合計		189		27	216

(注) 例外パターンと記したものが、2つのループ距離＝2のループが出発点と到達点を交差する形となっているもの。

これらの特性式の分母に対応する正解の生成方法を71－1ページと71－2ページにまとめて掲載する。

ブロック線図の形に対応した特性式の生成について

st(i)とlo(i)から対応するループ特性を生成しバッファに格納

A，B，Cをst(i)とlo(i)によって定められる3つのループの特性とすると、

(1) 1＋A＋B＋C型の場合

特性式の分母は個々のループの特性の和　1＋A＋B＋C　の形になる。

$$F(A,B,C)=\frac{1}{1+A+B+C}=\frac{1}{1+buf(1)+buf(2)+buf(3)}$$

問題として出題するときは、分母の多項式の各項を出題画面のテキストボックスに対応させて

buf(1)　⇒　text1.text　･･･テキストボックス1に表示
buf(2)　⇒　text2.text　･･･テキストボックス2に表示
buf(3)　⇒　text3.text　･･･テキストボックス3に表示

次のような方式で表示する

$$\frac{1}{1 + \boxed{\mathrm{text1}} + \boxed{\mathrm{text2}} + \boxed{\mathrm{text3}}}$$

Bbuf(1)　　　Bbuf(2)　　　Bbuf(3)

他の特性式の型の場合でも同様な方法で出題を行う

(2)　(1＋A)(1＋B)＋C　型の場合

特性式が以下の形となるため

$$F(A,B,C)=\frac{1}{(1+A)(1+B)+C}=\frac{1}{1+A+B+C+A*B}=\frac{1}{1+buf(1)+buf(2)+buf(3)+buf(1)*buf(2)}$$

出題画面にテキストボックスを4つ設定して以下のように表示する

buf(1)　⇒　text1.text　･･･テキストボックス1に表示
buf(2)　⇒　text2.text　･･･テキストボックス2に表示
buf(3)　⇒　text3.text　･･･テキストボックス3に表示
buf(1)*buf(2)　⇒　text4.text　･･･テキストボックス4に表示

（3）　（1＋A＋B）（1＋C）　型の場合

特性式が以下の形となるため

$$F(A,B,C)=\frac{1}{(1+A+B)(1+C)}=\frac{1}{1+A+B+C+A*C+B*C}$$

$$=\frac{1}{1+buf(1)+buf(2)+buf(3)+buf(1)*buf(3)+buf(2)*buf(3)}$$

出題画面にテキストボックスを５つ設定して以下のように表示する。

buf(1)　⇒　text1.text　･･･テキストボックス１に表示
buf(2)　⇒　text2.text　･･･テキストボックス２に表示
buf(3)　⇒　text3.text　･･･テキストボックス３に表示
buf(1)*buf(3)　⇒　text4.text　･･･テキストボックス４に表示
buf(2)*buf(3)　⇒　text5.text　･･･テキストボックス５に表示

（4）　（1＋A）（1＋B）（1＋C）　型の場合

特性式が以下の形となるため

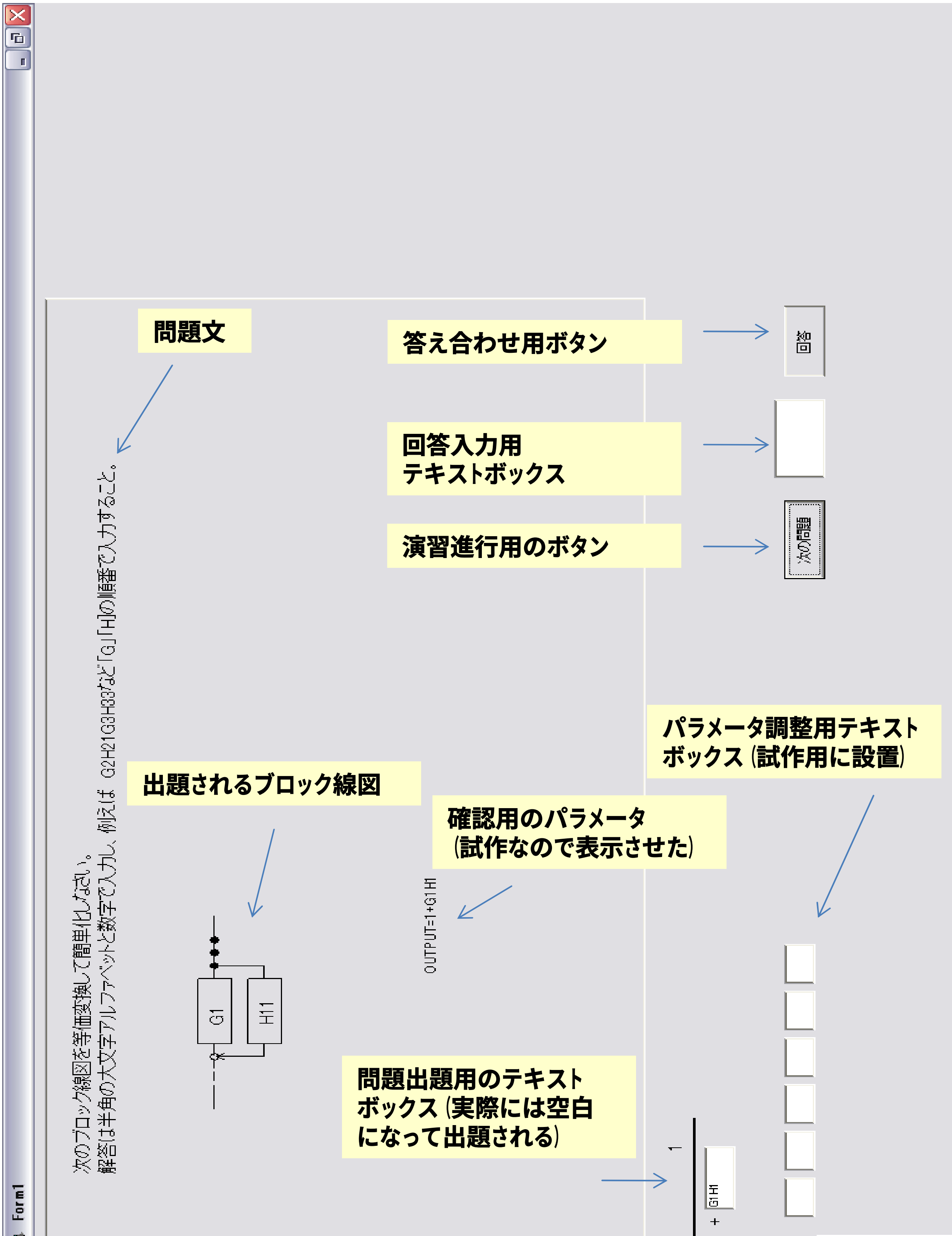
$$F(A,B,C)=\frac{1}{(1+A)(1+B)(1+C)}=\frac{1}{1+A+B+C+A*B+B*C+C*A+A*B*C}$$

$$=\frac{1}{1+buf(1)+buf(2)+buf(3)+buf(1)*buf(2)+buf(2)*buf(3)+buf(3)*buf(1)+buf(1)*buf(2)*buf(3)}$$

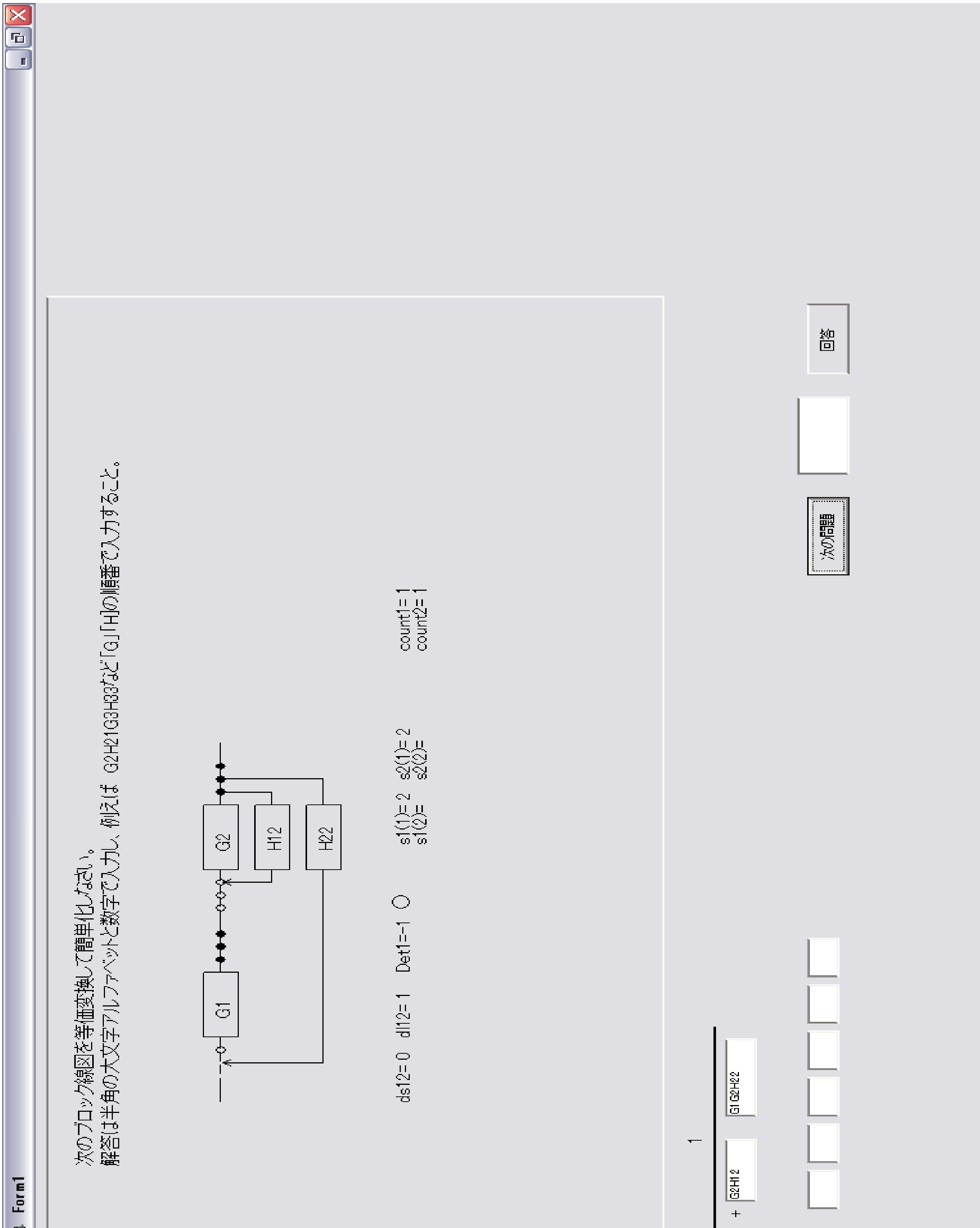
出題画面にテキストボックスを７つ設定して、以下のように表示する。

buf(1)　⇒　text1.text　･･･テキストボックス１に表示
buf(2)　⇒　text2.text　･･･テキストボックス２に表示
buf(3)　⇒　text3.text　･･･テキストボックス３に表示
buf(1)*buf(2)　⇒　text4.text　･･･テキストボックス４に表示
buf(2)*buf(3)　⇒　text5.text　･･･テキストボックス５に表示
buf(3)*buf(1)　⇒　text6.text　･･･テキストボックス６に表示
buf(1)*buf(2)*buf(3)　⇒　text7.text　･･･テキストボックス７に表示

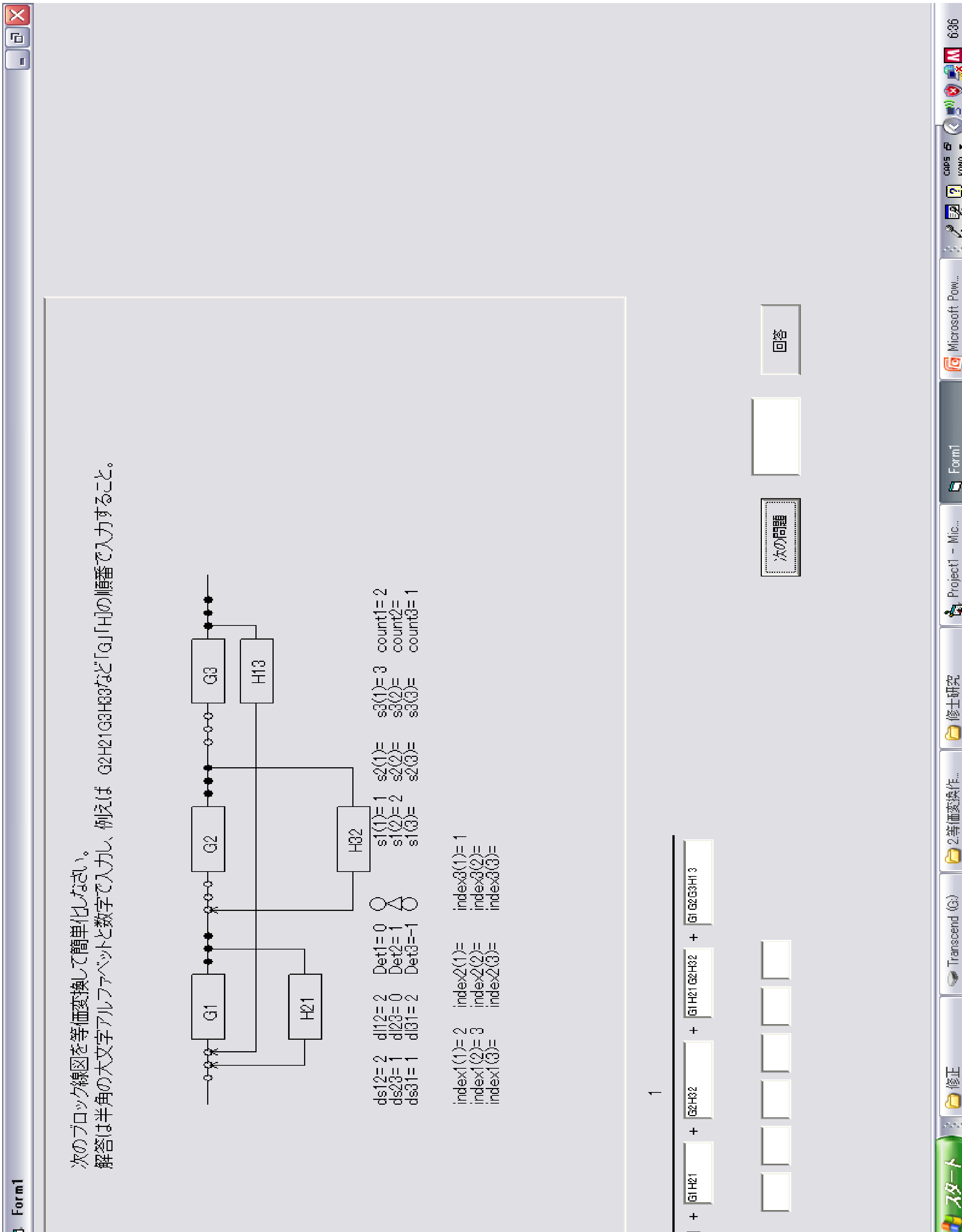
添付図－　ブロック線図の等価変換の問題出題の例（主軸要素数=1の場合）
画面の注釈は主軸要素数=2，3と共通



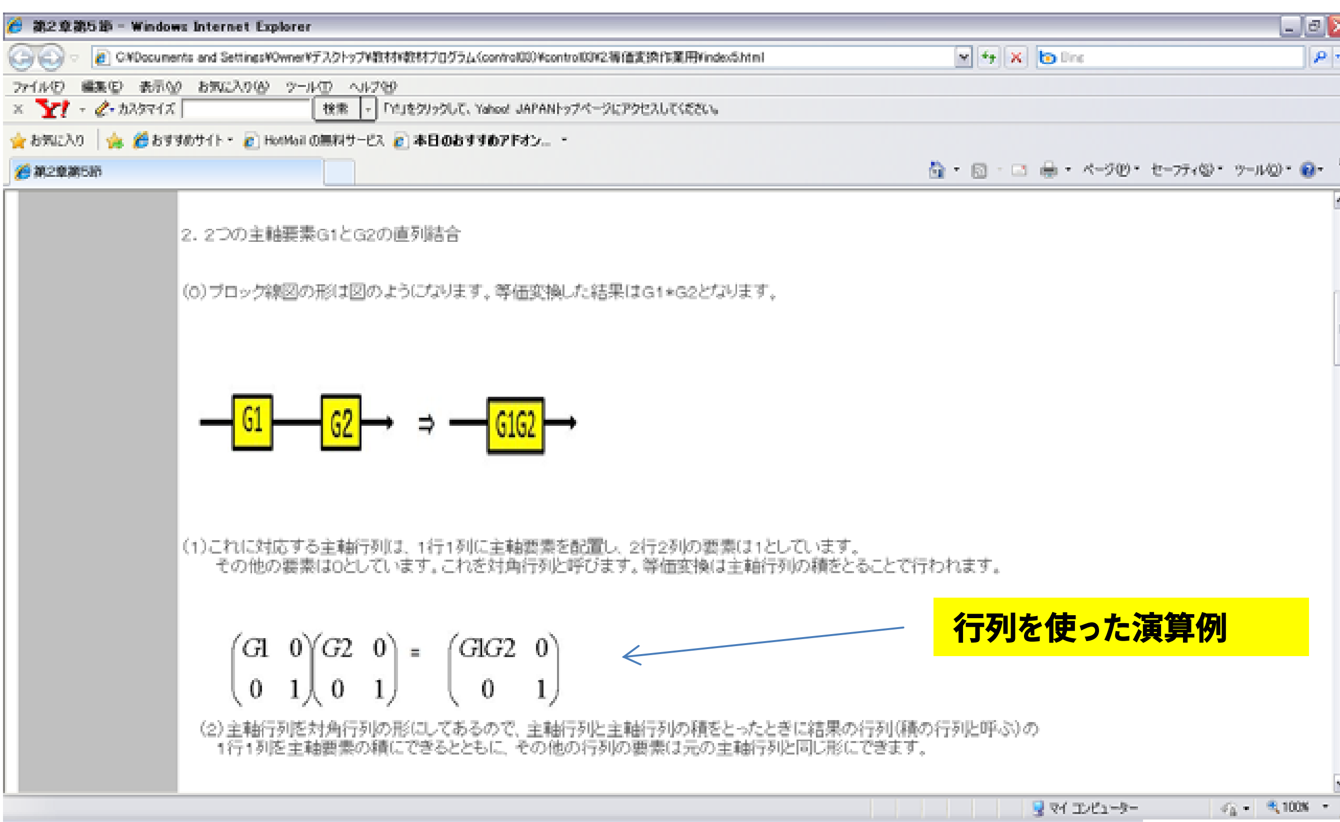
添付図－　ブロック線図の等価変換の問題出題の例 (主軸要素数=2の場合)



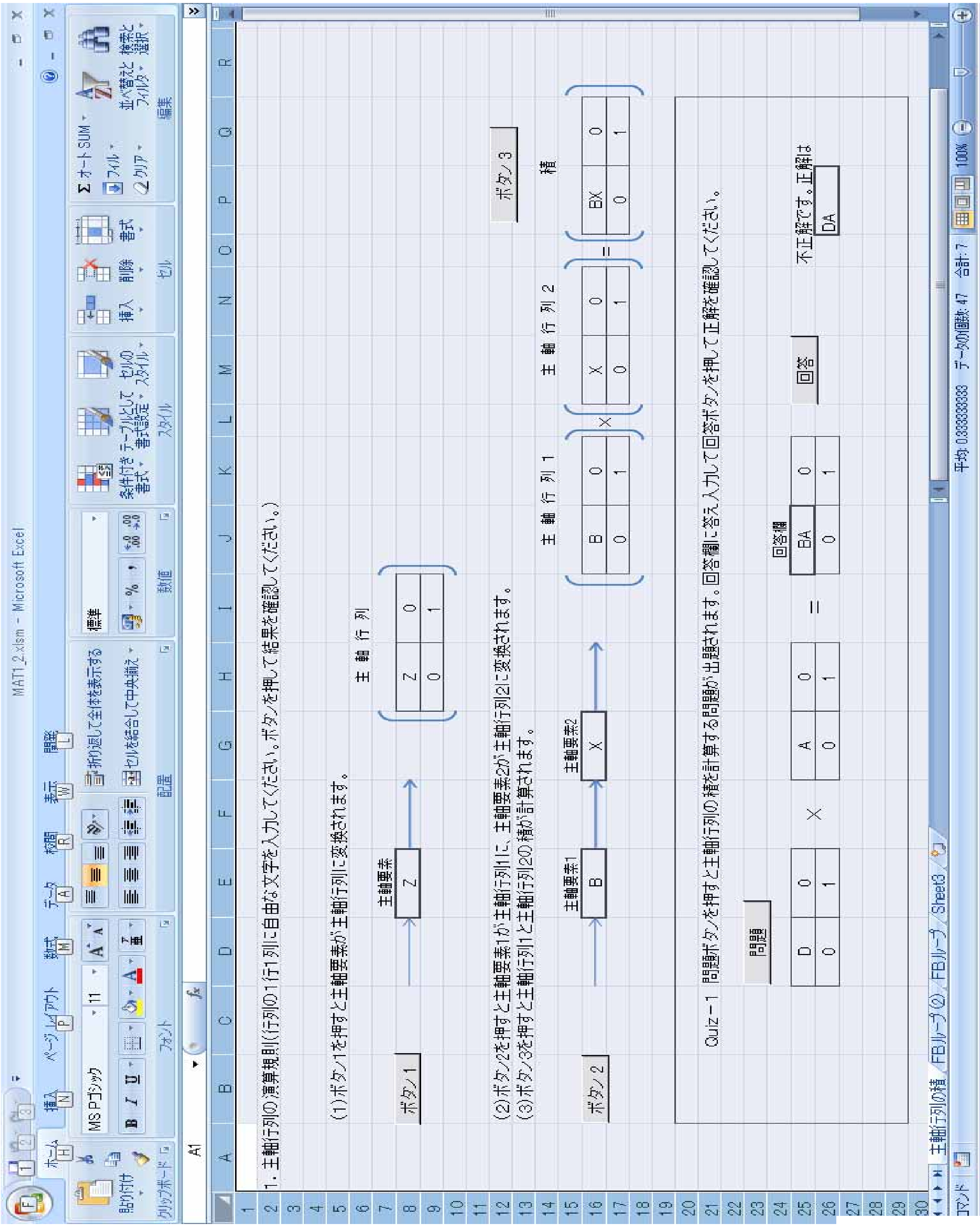
添付図－　ブロック線図の等価変換の問題出題の例（主軸要素数=3の場合）



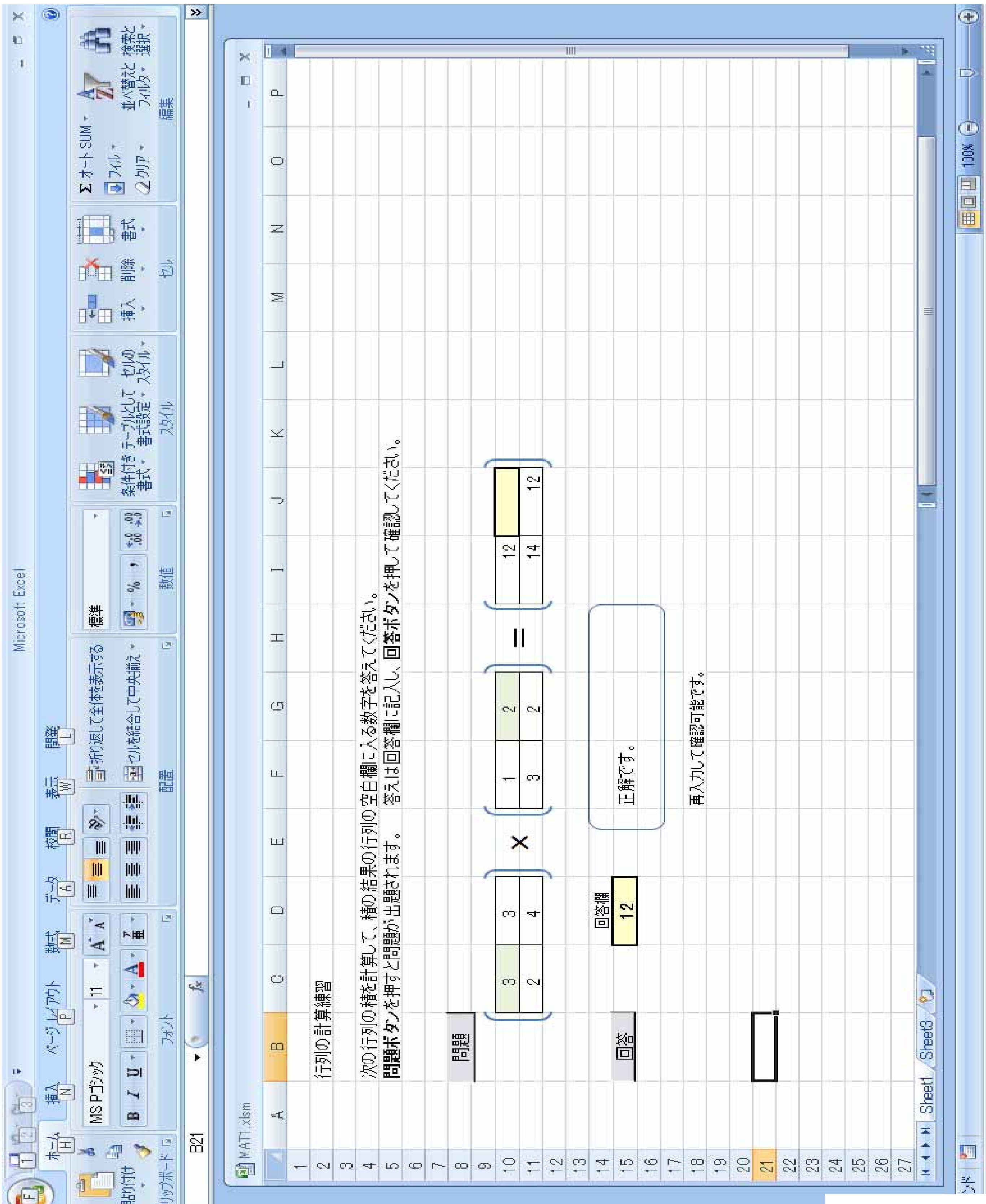
添付図－　　行列を使った等価変換の教材の試作例
（信州大学のＣＡＩシステム上で作りこんだ例）



添付図－　　ブロック線図の等価変換の演習問題の例
（教材中に作りこまずにExcelにリンク；乱数を使って出題の都度問題を変更）



添付図ー　行列計算の復習の例
(教材中に作りこまずにExcelにリンク;乱数を使って出題の都度問題を変更)



【IX. 結論】

1. 従来の連分数を使った方法に代わって、主軸 行列とフィードバック行列を用いた新しい計算方法を考案した。

2. 行列を使った計算方法により、学習者がブロック線図と等価変換のプロセスの対応関係を明確に理解できる学習教材のモデルを試作できた。

2. 行列を使った教材について、パソコン上で稼働する、学習者に対する理解を促進するための簡単なシミュレーションなどを含んだモデルを試作することができた。

3. 計算機を使ったブロック線図の等価変換の演習について

(1) フィードバックループの出発点とループ距離の概念に基づく分類指標を導入することにより、
 a. ブロック線図の構造をループ相互間の包含・独立の関係で分類可能とした。
 b. ブロック線図の形とブロック線図を等価変換した特性式の形とを分かりやすい形で関連付けることができた。

(2) ブロック線図の形とブロック線図を等価変換した特性式の形との関連付けを可能にしたことにより、フィードバックループの等価変換を行う順序に制約されることなく、ブロック線図の形に対応した特性式を構成することができた。

(3) 上記の方法により、演習システムを試作して機能を確認した結果、
 a. ブロック線図の描画
 b. 問題の出題 (出題されたブロック線図に対応する特性式の構成も含む)
 c. 答え合わせ
 などが正しく行われ、ブロック線図の等価変換に関する計算機を用いた演習システムを実用化可能な見通しがついた。

【X. 今後の課題】

1. 今回は学習者が自宅のパソコンで学習可能な教材としたが、今後ネットワーク上での活用を図るためサーバとの連携機能を開発する。

2. 今回は試作の段階にとどまったが、今後学習者に使ってもらって評価してもらい課題を探る。その結果でさらなる改良をはかる。

3. 今回は主軸要素数＝3を上限として教材を試作したが、更に規模拡張が可能かどうか検討する。

【XI. 謝辞】

研究を行うにあたり、長い間常に温かいご指導をいただき、励ましていただいたカワモト・ポーリン・ナオミ先生に心から感謝いたします。

【ＸⅡ．参考文献】

１．電気回路論　（電気学会；昭和４６年３月）
２．ＭＡＴＬＡＢによる制御理論の基礎　（野波、西村；東京電機大学出版局；１９９８年３月）
３．Ｓｃｉｌａｂで学ぶシステム制御の基礎（橋本他；オーム社；平成２０年４月）
４．Ｗｉｎｄｏｗｓを使って制御工学演習　（武内；共立出版；２００２年５月）
５．線形システム　（前田；朝倉書店；２００１年１２月）
６．回路網合成演習　（斎藤、西；朝倉書店；１９８５年１１月）
７．電気工学入門演習　自動制御　（佐藤；学献社；１９９３年３月）
８．例題　演習自動制御入門　（中村、若山；産業図書；昭和６３年３月）