TraFST

解説/Review
特集：品質・信頼性・安全性確保に関わる横幹的体系

超設計のパラダイムとメソドロジー

高橋　武則*1

Paradigm and Methodology of Hyper Design

Takenori TAKAHASHI*1

Abstract– This paper discusses Hyper Design which is a new design paradigm and shows its methodology concretely. A scientific design is carried out using the expression, however, it is approximate expression. The approximate expression is often a multivariate multinomial expression made by the data which is acquired by an experiment plan. When the specific variable (hyper factor) is focused in this expression, another multinomial expression based on focused variable can be constituted. After it is constituted, the necessary role is assigned for focused variable, design is enabled. An engineer has the freedom to decide which variable must be focused and decide which role must be assigned for it. This paper discusses a design as free creative activity of engineer from this point of view. Hyper Design discussed in this paper is capable to explain various types of design in a comprehensive manner, and at the same time, it is applicable to wide area of actual practices.

Keywords– Hyper Factor, Hyper Structured Function, Hyper Design, Hyper Optimization, Paradigm of Design

1. はじめに

設計には芸術家による感性的設計や職人による経験的設計もあるが，本稿は技術者による科学的設計について議論する．椿ら [15] は，「科学者は，『使った，効いた，だからよいことだ』という論法を嫌うことから出発している．どのようになっているからよいのだという，プロセスの正当性がなければ納得しないのである．」と述べている．これは科学的設計の本質を端的に示している．前述の論法は，医薬の世界では「三た論法」(対照を置かないで「使った，治った，故に効いた」という論法）と呼ばれて問題視されている．これはもの造りの世界でもしばしば通用しているが，科学的設計においてはタブーの論法である．本稿は数式と図表による設計プロセスの明快な論理的可視化が科学的設計の必須要件と考える．

科学の分類には様々なものがあるが，オーソドックスには自然科学，社会科学，人文科学（人文学）に分類される．科学的設計はその数理面・技術面では自然科学（理学と工学）との関連が深く，実際には理学と工学の密接な連携が不可欠である．工学はモノを作るが理学はモノを作らない．しかし理学抜きでは科学的設計は不可能である．科学的設計は数理的に見たらその中身の本質は模型化（モデリング）と最適化で構成されており [5][16]，前者は実験による数式作成のことで後者は数理計画法による求解のことであり，いずれも理学の守備範囲である．一方，設計目的や取り上げる因子・水準は工学の守備範囲であり，両者が連携することによって科学的設計が成立する．本稿はＱＭ（Quality Management）の立場から科学的設計全体を俯瞰する．具体的には，超設計 [8][9][10] と言うパラダイム（思考規範）に立脚して多種の設計を統一的に整理し，必要なメソドロジー（技法）を明らかにし，そのうえで設計の新しい様々な可能性について議論する．

Ｑ（Quality）には時制があり，どの時制のＱに焦点を合わせるかでＱＭの取り組み方が異なり，Ｑの時制の観点からＱＭの歴史を整理することができる．時制とは動詞における時間的関係を示す文法範疇のことで，その分類には様々なものがあるが典型的な範疇として過去・現在・未来の３分類がある．同様にＱにおいても時制が存在し「過去のＱ」，「現在のＱ」そして「未来のＱ」の３つがある．この観点から見ると，ＱＭの歴史はまさにＱの時制のパラダイムシフトの歴史であると言うことができる．

*1慶應義塾大学大学院健康マネジメント研究科　神奈川県藤沢市遠藤 4411
*1Graduate School of Health Management, Keio University, 4411 Endo, Fujisawa, Kanagawa

Received: 17 July 2019, Accepted: 12 August 2019.

ＱＭの初期の段階では「Ｑとは規格適合」というパラダイムのもとでＱＡ（Quality Assurance：品質保証）に軸足が置かれ，検査が重視された．Ｑは検査で保証するというわけで，中でも抜取検査が重要なテーマであった．

しかし，やがて，検査は過去のＱ（作られてしまったＱ）を選別しているわけで，Ｑは工程で保証すべきであるという現在のＱ（作りつつあるＱ）へと軸足を移行した．これはＱの時制のパラダイムシフトである．この時代は工程管理・工程改善が重要なテーマとして注目された．しかしながら，この段階では「Ｑとは規格適合」というパラダイムにはシフトはなかった．

ライバル商品が競合する状況下では，工程でしっかりと作り込み，検査でも手抜かりなく規格をきちんと満たした文句のない製品が市場でまったく売れないという事態がしばしば発生する．このとき，「Ｑとは規格適合」というパラダイムでは説明ができないことが露呈した．そこで，「Ｑとは顧客要求適合」というパラダイムシフトが起きた．これは言い換えると，「ＱとはＣＳ（Customer Satisfaction：顧客満足）である」ということである．この段階で重視されるのは未来のＱ（これから作るＱ）であり，正確にはこれから作るべきＱと言うことである．

未来のＱの場合にはさらに細かく分類すると以下のような 3 つに分類される．

近未来のＱ【設計】:間もなく作るＱ

中未来のＱ【Ｒ＆Ｄ】:少し先に作るＱ

遠未來のＱ【企画】:だいぶ先に作るＱ

上記の順番では企画（何を作るか，何を作るべきか）から始まり，次にＲ＆Ｄ（どういうメカニズムで作るか）を行い，その上で設計（どういう条件で作るか）を行う．なお，設計においてはそれに先行する研究・開発を必要とする場合が多い．三者の関係は以下のとおりである．

研究　：原理や法則を明らかにすること．

開発　：原理や法則を用いて目的の実現に必要なメカニズム (機構:仕組み，仕掛け) を創造すること．

設計　：目的を実現するために機構の条件（因子と水準）を決定すること．

しかしながら，紙数の都合により研究と開発については割愛して設計に焦点を合わせて議論する．

本稿は，「超設計とは特別な設計因子である超因子を用いて柔軟に構成した関数に基づいて数理計画法を用いて行う自由で創造的な条件の最適化である」というパラダイムのもとで設計を議論し，それに必要なメソドロジーを示す．このパラダイムのもとでは様々なタイプの設計を俯瞰することができるので，これを用いて設計を統一的に説明するとともに設計の可能性を広げる．

超設計は，関数それ自体を研究するわけではなく，有用な（設計目的を実現する）ものを作ることである．したがって，超設計において重要なことは，用いる関数は真の関数である必要はなく，設計目的に対して必要な区間において致命的な不適合のない（十分な適合性を持つ）近似式で構わないということである．設計に用いる近似式に関して重要なことは以下に示す LOF と LOI がともに致命的レベルでなければ構わないということである．

＊ LOF (Lack of Fit):適合の欠如（あてはまりの悪さ）
・自由度調整済み寄与率 R^{*2} で評価するとよい．
・誤差にも考慮をしたうえで，R^{*2} の値は 0.5 以上を確保したい．

＊ LOI（Lack of Independence）:独立性の欠如
・トレランス（=1/VIF）で評価するとよい．
VIF：Variance Inflaction Factor
・誤差にも考慮をしたうえで，トレランスの値は 0.5 以上を確保したい．

これらがいずれも致命的でなければ設計したもの（解）の多くはその実現確認で成功（解の受容れ可）するであろうし，仮に失敗（解の受容れ不可）したとしても事後の回帰調整で対応できることが多い．この考えに立てば実験を計画する際には最適計画 [17] を用いることが望ましい．

なお，具体的な説明の際には例として紙ヘリコプターを用いる．それは Box[1] に登場するタイプではなく，著者が開発した複葉型紙ヘリコプター [7] の下翼を取り外して単葉型に退化させた（上翼のみの）単葉型紙ヘリコプターである．

2. 超設計の背景

2.1 設計における誤差と LOF

超因子とは，設計因子の中で特別に指定された因子のことで，それは特別な役割を持つ因子である．そして，超因子に注目して超構造関数を構成し，これに基づいて行う設計が超設計である．これによって多種多様なタイプの設計を超因子・超設計という新たな視点から統一的に整理することができる．そして，この延長線上に存在する新たな設計の可能性を見出すことができる．

設計という創造の中身の実体は，数理計画法を活用する最適化である．ただし，近似式に基づく創造でありかつ誤差を伴うために，最適化によって得られた最適解はあくまでも仮説でしかない．もし，近似式の LOF が大きな場合や誤差のばらつきが大きな場合には，得られた解（最適値）は LOF and/or 誤差の影響で実現しないことが発生する．このため，最適解の実現確認は不可欠である．LOF と誤差の影響により実現確認で，予測値（設

計値）と実現値（データ）が多かれ少なかれずれることは避けられない．もしこのずれが受容れ難いレベルの場合には，最後に回帰による調整を行うことで，何としても設計目的を果たすということが実践型の設計アプローチでは必要になる．

本稿では主に誤差（確率誤差 ε）のある実実験の場合を取り上げるが，誤差のないシミュレーションの場合も若干議論する．なお，誤差のある場合に用いる数式は通常は推定式となるが，本稿では本質を簡潔に議論するために推定式の表現は用いずに通常の数式の表現を用いる．

実実験では誤差（純粋誤差）$\boldsymbol{\varepsilon}$ を無視することはできない．そして実験データから真の模型 $\boldsymbol{\pi}(\mathbf{x})$ を獲得することは困難で，近似式 $f(\mathbf{x})$ を模型化することになる．そこで，LOF を次のように表現する．

$$\begin{aligned} \mathrm{LOF} &= \pi(\mathbf{x}) - f(\mathbf{x}) \\ \mathbf{x} &= (x_1.x_2,\cdots,x_p) \end{aligned} \tag{1}$$

このとき超設計の基本的模型構造は以下のようになる．

$$\begin{aligned} y &= \pi(\mathbf{x}) + \varepsilon \\ &= f(\mathbf{x})+\mathrm{LOF}+\varepsilon\,, \quad \mathbf{x}=(x_1.x_2,\cdots,x_p) \end{aligned} \tag{2}$$

椿 [13] は統計模型を芝居に見立てて誤差 $\boldsymbol{\varepsilon}$ をアドリブになぞらえている．この見立てに従えば，LOF は削ぎ落し（捨象）になぞらえることができる．$\boldsymbol{\varepsilon}$ と LOF を適度におさえたものが優れた統計模型（見事な芝居）である．実践的な設計を行うためには以下の点が重要である．

① $\boldsymbol{\varepsilon}$ を必要なレベルまで小さくする．
※誤差なしシミュレーションの場合は無視する．

② クリティカルな LOF が生じない工夫を行う．

③ LOF と ε を分離する工夫を行う．

①は改善活動の範疇なので，紙数の都合により本稿ではその議論を割愛して②に焦点を合わせて議論する．その際に，③を配慮した実験計画（Resolution IV の計画+同一条件での複数の繰り返し）のもとでの F 検定や純粋誤差のもとでの予測区間を利用した t 検定を活用する．

2.2 テイラー展開に基づく近似式と工学的設計

1) テイラー展開に基づく近似式

実験を行って近似式（模型）を作成しようとする場合に適切な近似模型は 1 次模型か，積項模型か，あるいは 2 次模型かということについて実験の前に明確な情報がないことが少なくない．しかし，テイラー展開のメカニズムを応用すると近似式を用いた合理的なアプローチが可能になる．それは実験計画（DOE: Design of Experiment）に基づいて収集したデータで近似模型を作成し，それを用いた数理計画法による最適化という形の設計である．ここでは，分かり易い説明のために一般形の最小形として 2 次の積項と 2 次項を示すことができる 2 変数関数 $\boldsymbol{\pi}(x_1,x_2)$ の場合を取り上げる．なお，この場合の真の 2 変数関数は未知であるとする．これに関する点 (a,b) の近傍でのテイラー展開を以下に示す．

$$\begin{aligned} \pi(x_1.x_2) &= \pi(a,b) + \frac{\partial\pi(a,b)}{\partial x_1}(x_1-a) + \frac{\partial\pi(a,b)}{\partial x_2}(x_2-b) \\ &\quad + \frac{\partial^2\pi(a,b)}{\partial x_1\partial x_2}(x_1-a)(x_2-b) \\ &\quad + \frac{1}{2}\frac{\partial^2\pi(a,b)}{\partial x_1\partial x_1}(x_1-a)^2 + \frac{1}{2}\frac{\partial^2\pi(a,b)}{\partial x_2\partial x_2}(x_2-b)^2 \\ &\quad + \cdots \\ &= c_0 + c_1(x_1-a) + c_2(x_2-b) + c_{12}(x_1-a)(x_2-b) \\ &\quad + c_{11}(x_1-a)^2 + c_{22}(x_2-b)^2 + \cdots \end{aligned} \tag{3}$$

点 (a,b) そのものならば $\mathbf{y}$ は定数 $\boldsymbol{\pi}(a,b)$ となる．シミュレーションの場合にはこれが真値となり，実実験の場合にはこれに確率誤差 ε がつく．実験水準はこの点 (a,b) から離れるわけであるが，その離れ具合で以下のようになる．このとき，最終的にものを作る場合には，極端に広い水準をとることはありえないことを考慮する．

① ごく近傍ならば 1 次項までで近似ができる．

② 少し離れたならば積項までで近似ができる．

③ 更に離れた場合は 2 次項までで近似ができる．

④ かなり離れた場合は 3 次項を要する場合もある．

このことを踏まえて戦略的に設計を行うことができる．

もの作りの設計は長らく実実験により行われてきたが昨今はシミュレーションを用いる場合が少なくない [18]．しかしながら，本稿では原則として実実験の場合に焦点を合わせて説明を行い，シミュレーションについての説明が必要な場合に限り両者の違いを明らかにして，シミュレーションの場合のためのアプローチ方法の概要を示す．

2) 設計における量的因子に関する 4 種類の解

設計は有用なものの創造のための条件決定に関する関係者の合意形成とみることができる．それ故に，厳密な解にこだわる必要はない．量的因子に関する以下の 4 種類の解を状況によって柔軟に使い分けることが望ましい．

1) 実験点解：解の選択対象は実験点のみである．
2) 格子点解：量的模型（数量化 I 類）を用いて実験点の条件の組合せから解を選択する．
3) 内挿解：量的模型を用いて内挿の範囲で求解する．
4) 外挿解：量的模型を用いて外挿の範囲で求解する．

【注】 質的因子の場合は 1) と 2) しかない．したがって量的因子は質的に扱ってはならないし，質的因子は工夫して量的因子に直すべきである．

・ 量的因子：水準が連続尺度の因子
・ 質的因子：水準が名義尺度の因子

なお，次の 5 点を踏まえて求解することが重要である．

① 原則として内挿解を基本とするのが合理的である．
② 安全を考えて格子点解（DOE の解）は必ず用意する．
③ 格子点解の実現失敗に備え実験点解も用意する．．
④ 外挿解は試す価値があるので挑戦した方がよい．
⑤ 最後の切り札は実験点解である．

上記の① ～ ③の順番は解のレベルの順番でもある．低順位の解を確保した上で高順位の解を試すのがよい．特に外挿解は統計的にはタブーとされているが，これはチャレンジと考えればよい．解が実現すればとても良い解となるし，実現しない場合でも新たな情報が手に入る点で貴重な挑戦と言えよう．ただし，もし内挿解が実現しないという場合には②や③を用いるほかに，後ほど第 6 章で議論する「事後に行う回帰調整」という選択肢がある．

3. 超設計とは（基盤関数，超因子，超構造関数）

3.1 基盤関数

実験データで推定された近似式のことを基盤関数と呼ぶ．これは超設計のおおもとの関数で，この式では全ての設計因子は独立変数（説明変数）として対等である．

$$y = f(\mathbf{v}), \quad \mathbf{v} = (v_1, \cdots, v_{p+1}) \tag{4}$$

この関数をそのまま用いて望む y の値を実現する設計因子の水準を決定するのが通常の設計である．しかし，超因子というものを用いて進化させた設計が超設計である．逆に言えば超設計を退化させた設計（超因子のない設計）が通常の設計なのである．

3.2 超因子と超構造関数

設計因子の一つないし複数を超因子 $(H_A, H_B, \cdots)$ という特別な因子に指定した上で，これに注目して構成した多項式を超構造関数と呼ぶ．以下に最も基本的で易しい構造である超因子が H のみ（ただ一つだけ）でかつ超因子 H に関する超構造関数の次数が 1 次の場合を示す．

$$y = F(H; \mathbf{x}) = \lambda_0(\mathbf{x}) + \lambda_1(\mathbf{x})H, \;\; \mathbf{x} = (x_1, \cdots, x_p) \tag{5}$$

式 (4) から式 (5) への移行は以下の数理的な組換えである．

$$\mathbf{v} = (v_1, \cdots . v_{p+1}) \;\rightarrow\; (H, \mathbf{x}), \;\; \mathbf{x} = (x_1, \cdots, x_p)$$

これが 0 次の（超因子がないという）場合が通常の設計である．H に関して高次にすることや複数の超因子を指定して多変数の高次式することも可能である．本章では最も簡単な場合を用いて超設計の本質を説明する．なお，超因子に指定されなかった他の設計因子は超構造関数においては係数関数 $\lambda_0(\mathbf{x})$ と $\lambda_1(\mathbf{x})$ となる．ここで改めて超設計を超因子と超構造関数の観点から定義を行うと以下のようになる．

> 「超設計とは超因子を指定して超構造関数を構成し，これに関する最適化で設計目的を実現することである．」

超設計（最適化）の結果として係数関数の値が決定し，そのことにより最終的に超構造関数が確定する．

3.3 超因子の 4 つの役割と超設計の種類

上記の説明では超因子を一般形の変数 H で示したが，超因子の代表的なタイプには以下の 4 つがある．それぞれの特徴を表すために超因子の変数記号を使い分ける．

① 入力因子 M:これで y を制御する（望む値にする）．
② 撹乱因子 Z:これがばらつくと y が大きくばらつく．
③ 組織因子 U:この水準ごとに y に関する式が異なる．．
④ 描写因子 W:これと y とを組みにすることで対象の形態（形状や状態）が描写できる．したがって両者には因果関係はないので， W を W_1 とし，y を W_2 とした方が描写因子の意味が分かり易いかもしれない．

以下に各々の超構造関数とその意味を紹介する．なお, 係数 (定数項と傾き) には意味を示す名前を付けている．

① **入力因子による制御設計**：その都度自由に望む出力を実現する設計（構造：切片パートと傾きパート）

$$\begin{aligned} y = F(M; \mathbf{x}) \;&=\; b_0(\mathbf{x}) + b_1(\mathbf{x})M \\ &b_0 \text{ :intercept, } b_1 \text{ :slope} \end{aligned} \tag{6}$$

② **撹乱因子による頑健設計**：撹乱の影響を減衰して安定を確保する設計（構造：安定パートと乖離パート）

$$\begin{aligned} y = F(Z; \mathbf{x}) \;&=\; s(\mathbf{x}) + d(\mathbf{x})Z \\ &s \text{ :stable, } d \text{ :divergent} \end{aligned} \tag{7}$$

③ **組織因子による連合設計**：組織間で条件共有の合意を形成する設計（構造：平均パートと固有パート）

$$\begin{aligned} y = F(U; \mathbf{x}) \;&=\; a(\mathbf{x}) + p(\mathbf{x})U \\ &a \text{ :average, } p \text{ :particular} \end{aligned} \tag{8}$$

※ 頑健設計はロバストパラメータ設計とも呼ばれ，撹乱因子は誤差因子と呼ばれることが多い．

④ **描写因子による形態設計**：目標の形態（形状，状態）を実現する設計（構造：独立パートと関連パート）

$$y = F(W;\mathbf{x}) \;=\; l_0(\mathbf{x}) + l_1(\mathbf{x})U \qquad l_0 \text{ :unlinked, } l_1 \text{ :linked} \tag{9}$$

上記の 4 つの数式は数学的にはまったく同じである．しかし，超因子の役割（意味）によって第 1 項と第 2 項に役割を与えると，実際の設計は大きく異なる．

たとえば，①の制御設計では傾き部分を急にすることが多く，②の頑健設計では乖離部分を 0 に近付けることになる．③の連合設計では 4.7 節の要約関数で述べるような政略的な設計が行なわれ，④の形態設計では 7 章と 8 章で述べるような様々な応用が存在している．なお，①の M と②の Z を組み合わせた 2 つの超因子の場合が動特性の頑建設計 [7] である．この模型は，4 つの項（定数項，M の項，Z の項，MZ の項）から構成されるが，その解説は割愛する．

3.4 超因子と女王蜂

余談ではあるが，超構造関数の本質は蜜蜂の組織（女王蜂と働き蜂）に似ている．多変数関数は多数の独立変数（以後は因子と呼ぶ）を有しているが，その中から超因子とする因子を指定してそれに関する多項式（超構造関数）を構成すると他の因子はこの式において係数（係数関数）となる．数学的には多変数関数として何ら変わりはないが，設計においては何を超因子にするか，その超因子にどのような役割を与えるかでその後のアプローチは大きく異なる．

すでに紹介した「超因子」は例えて言えば蜜蜂の「女王蜂」である．女王蜂は特異な遺伝子を持ってロイヤルファミリーとして女王蜂になるべき運命で生まれてくるのではなく，他の多くの働き蜂と同様の遺伝子で雌蜂として誕生する．しかし，雌蜂の中の特別な蜂（王台と呼ばれる場所に産み付けられた雌蜂）だけにロイヤルゼリーが与えられることによって女王蜂になるわけである．そして他の雌蜂は働き蜂となり女王蜂とともに蜂の巣を築き運営していく．蜂の巣をシステム（プロセス，機械，製品）と考え，働き蜂を設計因子，女王蜂を超因子と考えれば，超設計の本質的な構造は蜂の巣というシステムの構造と同じであるということが言える．

3.5 設計のレベル（戦術，戦略，政略）

設計は数理的観点からみると数理計画法による最適化であるが，経営的観点からみると設計の本質的構造は作戦と同じである．そして，作戦は大きく戦術と戦略と政略に分類される．この観点から設計と言うものの構造を整理したい．数理計画法による最適化の基本は，所与の条件のもとでの最適化であり，それは戦術である．しかし，様々な条件（前提条件，関数，因子，水準等）を変えたり合意形成方法（ドクトリン，ルール，交渉方法等）を変えたりすることで解は大きく変化する．前者が戦略で後者が政略である．経営という観点から見た設計には戦術と戦略と政略の３レベルがあることを強調したい．

3.6 超設計のもとでの柔軟設計

1) 柔軟設計

超設計において鍵を握る重要な因子である超因子は設計因子の中から選ばれる．そして近似式として模型化された多変数関数（基盤関数）を超因子に関する多項式（超構造関数）に構成した上で設計（最適化）を行う．超因子には多種類の役割があり，どの因子を超因子にするのか，超因子の数をいくつにするのか，超因子にどのような役割を与えるのかは設計者の自由である．事前に超因子とその役割を決めて設計することが多いが，事後に超因子とその役割を決めて設計することも可能である．それどころか，超因子の指定やその役割を事後に自由に何度も変えることも可能である．それは以前にとったデータの再利用が可能なことを意味する．このような設計は柔軟設計 [8][9] と呼ばれ，これは超設計の進化形である．柔軟設計が可能であるということは，超設計というものが数理模型に基づく自由な創造であることを明らかに示している．そのことは超設計が設計の戦略と政略の可能性を広げるものであることを意味している．

2) 柔軟設計の数理

直積実験は内側配置と外側配置の積の実験であるために内側に存在する模型の全ての項と外側に存在する全ての項の積を扱うことができる [4]．変数選択をすることによって模型化がなされ基盤関数が獲得できる．このとき，一方の側あるいは両方の側に積項（交互作用）があれば，両者の積として 3 次ないしは 4 次という高次の積項（交互作用）が扱えるために様々な設計が可能になる．

高次の積項（交互作用）があると次の様に関数構造を自由に形成することができる．最初にある因子に注目してその因子を用いて多項式の関数構造を形成する．このとき多項式の各項の係数部分が残りの因子の関数となっている．このような構造の関数が超構造関数である．最初に指定してそれに関する多項式の関数構造を形成する変数は超変数であるが，設計の場合にはこれを超因子と呼び，係数の関数を形成するその他の変数が設計因子である．超因子は複数を指定することも可能である．一般には，頑健設計（ロバストパラメータ設計）において外側配置の因子を超因子として用いることが多く，そして量的外側因子は入力因子で，質的外側因子は撹乱因子として用いられることが多い．しかし，逆転の発想で，内側の設計因子を超因子として扱うことも可能である．この場合，数学的には外側と内側は双対の関係（互いに

裏返しの関係）になっている．さらに一層柔軟に考えれば，積項を持つ因子であればどれでも超因子とすることができる．すなわち，直積実験のデータによって得られる高次の積項のある式を基盤関数と呼び，これの中の因子を自由に選んで超因子とする設計が柔軟設計である．要は積項（交互作用）があればよいので，用いる実験は直積実験である必要はない．したがって，どのような方法であれ，積項（交互作用）のある式があれば柔軟設計は可能である．

- 既定設計：最初から設計因子に定まった役割を与えて実験を行いそのもとで設計する．
  - ※ 当初の目的に特化して余計なことはしないので折角の可能性を無視するので勿体ない．
- 柔軟設計：設計時に設計因子に自由に役割を与えて設計する．データの再利用が可能である．
  - ※ 多様な試みにより設計の様々な可能性を追究するとともにその限界についても把握することができる．

3) 柔軟設計による可能性の幅広い検討

戦略的設計あるいは政略的設計では因子の役割を自由に変更することになる．ただしそれはしばしば設備変更，設備投資，製作条件の変更をともなうもとで行うことになる．例えば，外気温度が撹乱因子の場合に，恒温設備を導入すればそれを前提条件や設計因子にすることができるし，室内の温度制御の可能な設備を導入すればそれを入力因子にすることもできるのである． 逆に，これまでたいへんな思いをして管理していたある作業を撹乱因子にして頑健設計を行えば，以後はその作業の管理は不要にすることができる．

柔軟設計では事後の（設計時）の因子役割の割り付けは自由なので，これを用いれば様々な可能性を試みて画期的な設計（戦略的設計，政略的設計）が可能になる．

4) 柔軟設計によるデータの再利用

多くの場合，ある目的でとった実験データはその目的のもとでのみ使用されてその役割を終える．しかし，このデータを用いて柔軟設計を行えば，既存のデータで新たな設計を行って新たな可能性を探ることができる．あるいはこれから実験を行うという場合に，既存の実験データを用いて柔軟設計で可能性や限界を把握したうえで，実験の計画を立てることができる．取り上げるべき因子と取り上げるべき水準，前提条件にすべき因子とその固定すべき水準，無視しても問題のない因子などの情報を獲得することができ，これらに基づいて高度な実験計画を立てることができる．

5) 柔軟設計による戦術的最適化と政略的最適化

頑健設計の場合，絶対的な撹乱因子（誤差因子とも呼ばれる）というものは存在しない．たまたまの事情で，制御できないとか制御したくないために撹乱因子としているだけである．設備投資や技術改良・技術革新で撹乱因子は前提条件や設計因子や入力因子にすることができる場合も少なくない．この意味で，設計を戦略的あるいは政略的に解くことを考えるべきである．これまでの設計は与えられた条件のままで戦術的に求解することが多かったがそれは不自由な設計（最適化）である．

4. 超連結関数

4.1 自動車のダッシュボードの計器パネル

設計の本質は自動車の運転に似ている．運転の本質は「2 次元の移動の条件（走る，曲がる，止まる）の決定」である．このために運転席の前のダッシュボードに計器盤であるパネルがあり，速度計，距離計，タコメーター，燃料計，水温計などがある．運転者はこれらから必要な情報を得たうえで刻々と判断を行って適切な運転を行う．自動車の計器パネルの場合にはここに運転に必要な様々な情報（自動車の様々な状態）が表示されている．

4.2 コンピュータ画面上の関数パネル

設計の本質は「設計因子とその水準の決定」である．自動車における運転手段は設計因子に当たり，自動車における計器パネルはコンピュータ画面上の関数の表示に当たる．本稿ではこれを関数パネル（関数表示盤：関数一覧）と呼ぶ．その上に表示されるものは設計因子の関数とそれらの合成関数であり，これらの関数をパネル関数と呼ぶ．これらは自動車のパネルとは異なり多重の入れ子構造（マトリョーシカ人形の構造）でかつネットワーク構造（非樹形構造）の多重合成関数となっている．この多重でかつ非樹形構造の連結を超連結構造と呼び，数理計画法を用いて求解（多目的最適化）することが超設計の数理的な本質である．

運転は運転者が一人で条件決定を行うが，設計は多数の関係者が相談して条件決定を行う．このため設計は関係者の合意形成である．そして合意形成のためには，多様な情報である超連結関数を全員に可視化して見せることのできるコンピュータ画面（関数パネル）は重要である．

4.3 必須関数・考慮関数・把握関数

超連結とは，多数の関数の間にある構造が他の多数の関数を高度に入り組んだ形で引用するという複雑な多重構造の状態のことである．設計のため関数パネル（コンピュータ画面）上に登場する多くの超連結関数はそのほ

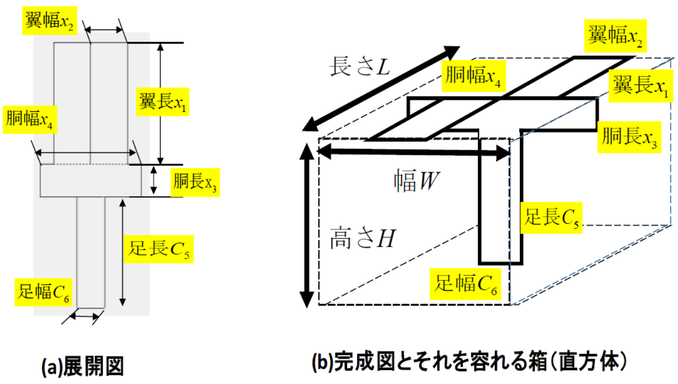

Fig. 1: 単葉型紙ヘリコプター（設計因子は 4 因子）

とんどが多重の超連結の状態になっている．用意した超連結関数が多い場合には，混乱を避けるために関数パネル上には必要最小減のものを表示し，関連する他のものが必要となったら直ぐにそれらをディスプレイ上に表示することが設計ソフトの要件である．もともと超連結関数は設計因子の関数とそれらの合成関数なので，必要なものをその都度呼び出すことはコンピュータを用いれば容易である．このことで漏れなく多面的な配慮ができ，関係者の合意形成も行い易いために総合的な設計が可能になる．

多数の超連結関数は以下のように分類し使い分ける．

＊ 必須関数：定式化に不可欠の関数
　原則として条件を譲歩しない関数

＊ 考慮関数：定式化に取り上げるべき重要な関数
　必要ならば条件を譲歩する関数

＊ 把握関数：定式化に取り上げないが解が求まったらそれを代入して様子を把握する関数（解を受容れることが原則）

定式化は必須関数と考慮関数で行い，もし解が得られない場合には考慮関数の条件を譲歩した再定式化で再度解を求める．必須関数の譲歩は最終手段とする．なお，把握関数はもし解を代入して受容れられないことになった場合には次の定式化で考慮関数に昇格させる．

4.4 超連結関数の具体的な例

以下に紙ヘリコプターを例として多種多様な合成関数で用意される超連結関数を具体的に紹介する．この設計で必要となる主な超連結関数には時間（飛行時間），面積，体積，段差などがある．ここでは設計因子の少ない単葉型紙ヘリコプターとして **Fig. 1** を紹介する．

なお，段差とは接している 2 箇所の寸法のずれ（ギャップ）のことで，これが無ければ（その値が 0 ならば）工数が減って切断作業が楽になるとともにミスやトラブルが減るために総合的に大きな効果となる．また，ここでの階層とは最初の関数が第 1 階層で，それを用いた合成関数が第 2 階層となり，さらにそれを用いた合成関数が第 3 階層となる．なお，この例では，必須関数が時間で，考慮関数は面積・体積・段差であり，それら以外の指標は把握関数である．

4.5 必須関数としての出力

最も重要なものは出力（時間）であり，これが顧客要求を満たさない限り製品としての存在意義はない．しかし，これを満たしても優れた製品とはいえず，同時に次項で示す多種多様な指標を満たす必要がある．

① 時間：

$$y = f(\mathbf{x}) = f(x_1,x_2,x_3,x_4) = c_0 + \sum_{i=1}^{4} c_i x_i + \sum_{i=1}^{4}\sum_{j>i}^{4} c_{ij} x_i x_j \qquad (10)$$

4.6 多種多様な階層構造の指標

指標とは QCDSE（品質，コスト，納期，安全，環境）に関するものでそれらは設計因子の関数である．ゆえに，設計においてはこれらの指標類を考慮関数や把握関数として取り上げなければならない．

② 翼面積 （rotor）： $S_R(x_1,x_2) = 2x_1x_2 \qquad (11)$

③ 胴面積 （body）： $S_B(x_3,x_4) = x_3x_4 \qquad (12)$

④ 全面積 （total）：

$$\begin{aligned} S_T(x_1,x_2,x_3,x_4) &= \kappa\{S_R(x_1.x_2), S_B(x_3,x_4)\} \\ &= S_R(x_1.x_2) + S_B(x_3,x_4) + C_5C_6 \end{aligned} \qquad (13)$$

⑤ 長さ （length）： $L(x_1) = 2x_1 \qquad (14)$

⑥ 幅 （width）： $W(x_2,x_4) = \max\{2x_2,x_4,C_6\} \qquad (15)$

⑦ 高さ （height）： $H(x_3) = x_3 + C_5 \qquad (16)$

⑧ 平面図の面積 （plan figure）：

$$\begin{aligned} S_P(x_1,x_2,x_4) &= \kappa_P\{L(x_1), W(x_2,x_4)\} \\ &= L(x_1) * W(x_2.x_4) \end{aligned} \qquad (17)$$

⑨ 正面図の面積 （front figure）：

$$\begin{aligned} S_F(x_2,x_3,x_4) &= \kappa_F\{W(x_2,x_4), H(x_3)\} \\ &= W(x_2.x_4) * H(x_3) \end{aligned} \qquad (18)$$

⑩ 側面図の面積（side figure）：

$$S_S(x_1,x_3) = \kappa_S\{L(x_1),H(x_3)\} = L(x_1)*H(x_3) \quad (19)$$

⑪ 体積（volume）：

$$V(x_1,x_2,x_3,x_4) = \lambda_V\{S_P(x_1,x_2,x_4),H(x_3)\} = S_P(x_1,x_2,x_4)*H(x_3) \quad (20)$$

⑫ 翼幅と胴幅の差（gap1）：$G_1(x_2,x_4) = 2x_2 - x_4$ (21)

⑬ 胴幅と胴幅の差（gap2）：$G_2(x_4) = x_4 - C_6$ (22)

このように簡単に上げただけでも 13 個の超連結関数があり，この他にも様々な超連結関数を用意することができるがそれについては省略する．ここで主張したいことは「設計は総合的な決定なので様々な指標は無視できない」ということである．なお，常にすべてのものを用いるわけではなく設計目的にとって必要なものを採用すればよい．

4.7 要約関数

コンピュータ画面上に登場する重要な関数にはさらに要約関数がある．これは要約統計量の関数版で，これを用いると質的超因子が多水準の場合を容易に扱うことができ，定式化および求解の結果の理解が容易になる．

撹乱因子が多水準の場合や多因子の場合を扱うにはダミー変数の構造がかなり複雑になり対応が困難になる．その場合には，以下に示す要約関数が有用である．これはデータに関する要約統計量の発展した関数版のもので，これらもまた数理的には多重の合成関数である．代表的なものとして以下の 5 種類の要約関数がよく用いられるが，これらはコンピュータを用いれば容易に扱うことができる．なお，式 (23) の左上添字は撹乱因子の水準を意味し，この式では k 水準の場合を示している．

$$\begin{aligned} f_{Max}(\mathbf{x}) &= Max\left\{{}^1f(\mathbf{x}),\cdots,{}^kf(\mathbf{x})\right\} \\ f_{Min}(\mathbf{x}) &= Min\left\{{}^1f(\mathbf{x}),\cdots,{}^kf(\mathbf{x})\right\} \\ f_{Ran}(\mathbf{x}) &= g_{Max}(\mathbf{x}) - g_{Min}(\mathbf{x}) \\ f_{Ave}(\mathbf{x}) &= \textstyle\sum_{i=1}^{k}\ {}^if(\mathbf{x})/k \\ f_{SD}(\mathbf{x}) &= \sqrt{\textstyle\sum_{i=1}^{k}\ \{{}^if(\mathbf{x}) - f_{Ave}(\mathbf{x})\}^2\ /\ (k-1)} \end{aligned} \quad (23)$$

上記の要約関数を用いると，頑健設計や連合設計において撹乱因子・連合因子の水準が多い場合に以下に示すように最適化のための定式化が簡単かつ明快になる．

撹乱因子における乖離や連合因子における個別の違いは範囲（Range）という形で扱うことができる．水準が異なると最大と最小の差が開き，これはばらつきである乖離や個別の違い（不公平）を表している．これを範囲として扱い，できれば 0 にするかあるいはある値以下にするということが，乖離の減衰や個別の違いの減衰の定式化である．その上で平均パート・共通パートを最適化（最大化・最小化・目標接近化）する．なお，範囲ではなく標準偏差（SD: Standard Deviation）を用いることも選択できるが，多くの人にとって範囲の方が分かり易い．

要約関数を用いるとミニマックス（Minimax）やマックスミニ（Maxmin）の設計が可能になる．それらをさらに変化させて Max-Max, Min-Min, Ave-Max, Ave-Min などのような多様で高度な設計が可能になる．

なお，連合設計において注意すべきことは強く公平（範囲を小さくすること）に拘らないことである．公平さを徹底するとその制約が強く効いて全員が不幸になるという解になることが多い．“味の評価は人それぞれなので，全員に差が出ない公平な味にするにはまずいものにすればよい．”というアイロニー（辛辣な言葉）がある．多少の不公平さを許容することによって平均を良くすることあるいは最小や最大を良くすることが大切である．これは頑健設計においても同様で，頑健性に強く拘ると委縮した解（頑健ではあるがまったく魅力のない解）になることが多い．頑健性を確保したままで平均を自由に調整できるという都合の良い因子（調整因子）は多くの場合存在しない．

5. 可視化の 3 ツールと実実験・シミュレーション

5.1 超設計のための可視化の 3 ツール

1) 特性要因図

特性要因図は，**Fig. 2**(1) に示すところの特性（結果）に影響を与える可能性のある因子（原因）を樹木構造で整理した図である．これは注目すべき因子を漏れなくリストアップできるツールで，その形状から Fishbone Diagram とも呼ばれ，世界中で利用されている．

2) 因子役割図

特性要因図に基づいて特性に強い影響がある因子は，設計する段階ではその役割に注意しなければならない．また，設計においては取り上げるべき結果系のものとして特性の他に費用・生産性ほかの評価指標がある．また実験で因子として取り上げないものにも注意が必要である．前提条件はその水準によって話が大きく変わる．干渉因子はその変動で実験を混乱させる．これらを可視化したものが因子役割図である．

前提条件：因子とその水準を明示し，実験中は管理するか観察して記録を残す．

Table 1: 構造模型表

(1)構造模型表(水準幅が狭い場合)

因子	1次項(主効果)	積項(交互作用)と2次項				
		翼長	翼幅	胴長	胴幅	足幅
翼長	○	×	△	×	×	×
翼幅	○		×	×	×	×
胴長	△			×	?	×
胴幅	△				×	×
足幅	○					×

1次模型

(2)構造模型表(水準幅が中程度の場合)

因子	1次項(主効果)	積項(交互相作用と2次項				
		翼長	翼幅	胴長	胴幅	足幅
翼長	◎	?	◎	×	?	△
翼幅	○		×	×	×	?
胴長	○			×	○	?
胴幅	○				×	×
足幅	○					×

積模型

(3)構造模型表(水準幅が広い場合)

因子	1次項(主効果)	積項(交互作用)と2次項				
		翼長	翼幅	胴長	胴幅	足幅
翼長	◎	◎	◎	×	△	○
翼幅	◎		◎	×	×	△
胴長	○			×	○	△
胴幅	○				×	×
足幅	○					×

2次模型

干渉因子：この因子は前提条件と異なり実験中に水準がばらついて実験を混乱させる．この因子は工夫してその水準を固定するか，実験順番をランダムにしてその影響を確率誤差にするか，あるいは撹乱因子として取り上げてその影響を減衰する．

設計因子：実験に取り上げてその水準を計画的に変更してその影響を見る．

3) 構造模型表

線点図は実験計画の普及に大きな貢献をした [14]．しかしこれは，1 次項（主効果）と積項（交互作用）の表現には向くが，2 次項には対応していない．近年は最適計画がコンピュータソフトで容易に計画できるので，線点図に替わるものとして **Table 1** に示す構造模型表 [12] を用いるとよい．これは 1 次項（主効果）と積項（交互作用）と 2 次項の有無だけでなく，各々の程度（強さ）・確度（確信度合い）についても表現ができるという利点を有している．

構造模型表の代表的な使い方には以下のものがある．

【安全な計画】明かに×以外のものは取り上げる．
実験数は増えるが安全である．

【大胆な計画】？ と×については取り上げない．
リスクはあるが実験数を減らすことができる．

※ 失敗したら不足している項を追加実験で補う．このための計画もコンピュータソフトで簡単に作ることができる．

つまり，戦略的・政略的な実験の計画が可能である．

5.2 変数選択による模型化

通常の場合，実験に際して設計因子として取り上げられるものは多数あり，そのすべてが y に対して効いているわけではない．したがって，何が効いているかを吟味し，そして効いているものによる模型化を行う方法として変数選択を用いるのが合理的である．

実際の取り組みでは先ず特性要因図で因果関係を体系的に可視化する．次にその中の強く効いていそうな因子で設計に使えそうなものを候補として取り上げ，因子役割図を作成する．取り上げないものは前提条件としその固定水準を明示する．そして，第 4 章で示したように，特性（出力）とともに考慮すべき多種多様な指標（設計因子の関数）を用いて構造的な最適化を行う．

なお，**Fig. 2** に，紙へリコプターの場合の特性要因図と因子役割図を例として示している．

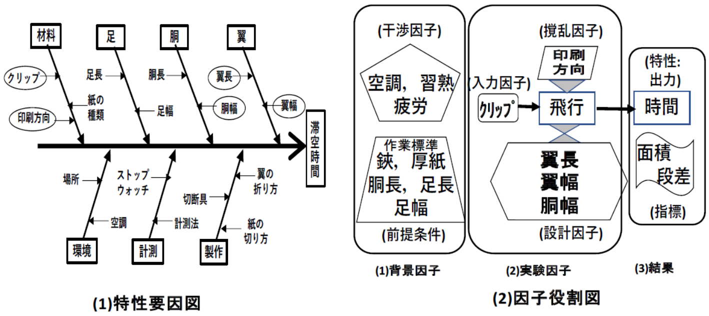

Fig. 2: 特性要因図と因子役割図

5.3 選択指標としての自由度調整済み寄与率

模型化では重回帰分析の変数選択を用いる．その際に問題となるのは選択指標とその基準値である．一般には選択指標として p 値が用いられるが，本稿は寄与率を用いる．その理由は次の通りである．結果である y の変動は原因である設計因子による想定された模型のもとでの変動とそれ以外の LOF（不適合：母数）と確率誤差（変量）による変動から構成される．しかし近年はしばしばシミュレーションによる設計が行われ，その場合には確率誤差がないために分布を前提とした p 値は機能しない．しかし，その場合でも寄与率は有用に機能する．

寄与率とは「データの全情報の中で各要素のもつ情報が占める割合のこと」である．なお，寄与率には寄与率 R^2，自由度調整済み寄与率 R^{*2}，自由度二重調整済み寄与率 R^{**2} の 3 種類があり [2]，自由度調整済み寄与率 R^{*2} か自由度二重調整済み寄与率 R^{**2} を使用することが望ましい．

5.4 真値誤差率と範囲誤差率

確率誤差がある場合の誤差に関する吟味では分散比を用いた F 検定や予測区間を用いた t 検定を用いることができる．しかし，誤差のないシミュレーションの場合にはこれらを使用することはできない．この場合には以下のものを利用する．

誤差率（相対誤差）＝ (実測値 – 真値) / 真値

※本稿ではこれを真値誤差率と呼ぶ．

誤差のないシミュレーションは真値が分かるので，近似値の評価の選択肢として真値誤差率が考えられる．しかし，これは真値の大きさに依存するので使い方に注意が必要である．この場合には以下に示す範囲誤差率（%表示）を用いることが望ましいと考える．

範囲誤差率 ＝ (予測値 – 真値) / 範囲

※範囲＝実験データの最大値 – 実験データの最小値

シミュレーションの場合は確率誤差がないため実験から真値が分かるので，誤差は予測値と真値との差で求められる．そこで，次節では中心点に関する誤差を用いて2次項の必要の有無を検討する場合にはこの誤差を範囲で割った範囲誤差率で判断する方法を採用する．なお，設計後の実現確認では真値誤差率と併用するとよい．

5.5 実現確認における検定と範囲誤差率

1) 実実験の場合の検定による判断

設計で重要なことはその実現を確認することである．このとき確率誤差がある場合とない場合では評価方法が異なる．確率誤差がある場合には以下の予測区間による検定を用いる．

① 設計条件における m 個のデータの平均値の予測区間を求める．

② m 個のデータの平均値が区間外に出現したら失敗（有意：不足の項あり）と判断する．

誤差の無いシミュレーションにはこの方法は適用できないので前節で述べた範囲誤差率を用いる．この値が決められた基準を超えた場合には LOF は無視できないと判断する．その基準値の一つの目安として 3.6%を推奨する．この値は2次曲線の一部分を直線とみなせるかということに関して範囲誤差率の様々なケースについて調べた際に20人中の18人が直線とみなしてよいと答えた値である．ただしこれは感覚的な目安なので客観的な基準を求めることが今後の課題である．

2) シミュレーションの場合の範囲誤差率による判断

初期実験の後に追加実験が行われ，全データで設計がなされる．その際の設計条件で実現確認を行い，得られたデータ（真値）と予測値と範囲で範囲誤差率を計算する．この場合の判断の基準は設計の目的による．範囲誤差率は後述する模型同定のためのものであり，実現確認に関しては真値誤差率の方が実践的である．

5.6 受容れ不可の場合の回帰調整

実現確認の結果，もし設計（解）が受容れられない場合には回帰調整を行う．初期実験と追加実験を通して効いている設計因子の水準や特性の凡その値が把握できているので，調整に良さそうな因子を一つ選び他の因子の水準は好ましいと思われる水準に固定する．このことにより LOF をおさえることができる．調整因子の水準は望む値が得られそうなものにし，水準数は非線形を想定して4ないし5水準にして繰り返しをとるとよい．このことで LOF とばらつきの両方をおさえることができる．このデータで回帰式を求めたうえで，望む値を実現する条件を見つける．

6. 模型の同定と包括設計法（包括法）CDM

6.1 3 ステージと 6 ステップの概要

包括とはいろいろなものを一つにまとめることをいう．本章では設計として必要なものを一つにまとめたものを包括設計法（CDM: Comprehensive Design Method）と呼び，以後は包括法と表記する．これには大きく分けると3ステージがあり，そのもとで具体的な取り組みには6ステップがある．

[I 準備ステージ] 誤差なしのシミュレーションでは不要

誤差が大きければ以後のステージでの統計的性質が悪く，量産時には工程能力指数（Cp，Cpk）が低くなる．

・ $\mathrm{Cp}=(S_U - S_L)/6\sigma$

・ $\mathrm{Cpk}=\mathrm{Min}\{(S_U-\mu)/3\sigma, (\mu - S_L)/3\sigma\}$

① 誤差管理：誤差を低減したうえで管理する．

[II 本番ステージ]

クリティカルな LOF のない模型を作成し，広く配慮した多目的最適化を行う．

② 同定化：模型構造を明らかにする．

③ 模型化：追加実験を行って基盤関数を確定する．

④ 最適化：数理計画法で最適化（仮決定）する．

[III 事後ステージ]

解は仮説（仮決定）でしかなく，その実現の確認が必要で，問題があれば調整しなければならない．

⑤ 実現確認：最適解の実現を確認する．

⑥ 回帰調整：ずれた解を受容れ可能状態に調整する．

6.2 同定化と実現確認と回帰調整の詳細

以後，前節の中で設計における成功の重要な鍵を握る②同定化と⑤実現確認と⑥回帰調整について説明する．

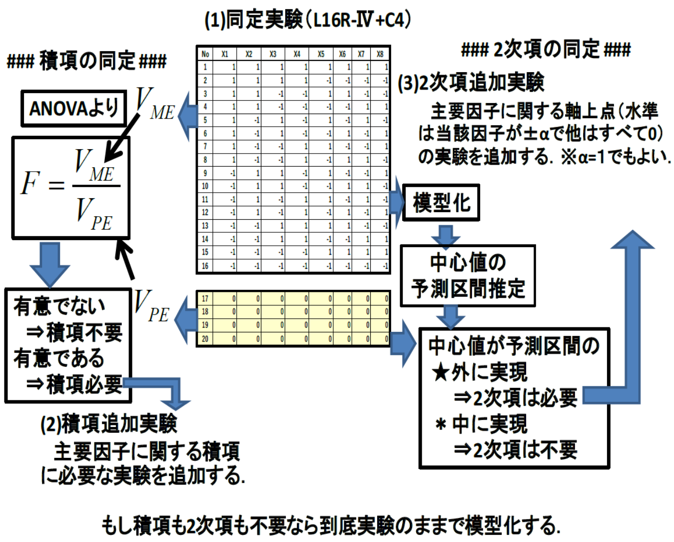

Fig. 3: 積項と 2 次項の判定による模型同定

1) 【同定化】 模型構造を明らかにする.

＊初期実験 ：主要因子の選択と模型構造の同定の実験.

- 8 因子に対しては L_{16}R-IV+C_4 の計画で実験する.（シミュレーションの場合は L_{16}R-IV+C_1）
  - ・ L_{16}R-IV：L_{16}Resolution IV
  - ・ C_4：中心点での 4 回の繰り返し
  - ・ C_1：中心点で 1 回だけ実施

＊因子選抜と構造同定 ：主要因子選抜と模型構造の同定.

- 標準偏回帰係数の絶対値の大きさの情報に基づいて，上位 5 因子以内に絞るとともに積項，2 次項の有無を判断する.
- 実実験は予測区間で，シミュレーションの場合には範囲誤差率で判断する.
  【注】判定の方法については **Fg. 3** を参照されたい. ここでは 2 水準を-1, 1 そして中点を 0 としている.

＊予測区間：判定に用いる両側検定のための予測区間

ID 点（identification point：同定のために用いる点）の平均値（繰り返し数は k）に関する予測区間とは，実験データだけで作成した模型のもとでの平均値の予測区間の式における不偏分散と t 値の自由度をそれぞれ純粋誤差分散 V_{PE} と自由度 ϕ_{PE} に切り替えたものである.

ID 点 $\mathbf{x}_\sharp=(x_{1\sharp},\cdots,x_{p\sharp})$ における k 個のデータの平均値の予測区間

$$\hat{y}_\sharp \pm t(\phi_{PE},\alpha)\sqrt{\left(\frac{1}{k}+\frac{1}{n}+\frac{D_\sharp^2}{n-1}\right)V_{PE}}$$

$$D_\sharp^2=(n-1)\sum_{i=1}^{p}\sum_{j=1}^{p}(x_{\sharp i}-\bar{x}_i)(x_{\sharp j}-\bar{x}_j)S^{ij}$$

$$S=(S_{ij}),\ \ S_{ij}=\sum_{\alpha=1}^{n}(x_{\alpha i}-\bar{x}_i)(x_{\alpha j}-\bar{x}_j)\ \rightarrow S^{-1}=(S^{ij}) \tag{24}$$

2)【実現確認】予測区間と誤差率で解の実現を確認する.

予測区間あるいは誤差率を用いて解の実現を確認する.

＊ 実実験の場合は予測区間による両側検定を行う.

＊ シミュレーションの場合には真値誤差率と範囲誤差率で判断する.

この後に必要な項を得るための追加実験を行う.

3) 【回帰調整】 回帰分析を用いて解を調整する.

＊ それまでの情報で調整因子を選択しその水準を決めて回帰実験を行なって回帰式を求めて調整する.

＊ 他の因子は望ましい水準で固定し，調整因子の水準は非線形に対応するために 4 ないし 5 水準とし，かつ精度を上げるために各水準で繰り返しをとる.

4)【3 次項に関する同定】

時には 3 次を想定する必要があるという場合もあり，この場合には ID 点を増やして同定方法を工夫する．C 点（center point）に加えて上下の四半分点 (quarter point) である Q_L 点，Q_U 点を用いる．これを分かり易く説明するために，**Fig. 4** に 2 次元の場合を示している．各点の定義は以下のようになる.

C 点（中心点）:全ての因子の中央の水準

Q_L 点:全ての因子の下端から水準幅の四半分の水準

Q_U 点:全ての因子の上端から水準幅の四半分の水準

これらの ID 点を用いて以下のように模型の同定を行う.

(1) C 点の予測区間を用いた **2** 次項の判断

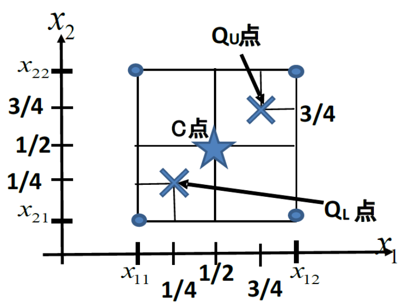

Fig. 4: 3 種類の ID 点（C 点，Q_L 点，Q_U 点）

C 点のみを用いた場合に，これが予測区間内であれば 2 次項を用いなくてもクリティカルな LOF は生じない．もし予測区間を逸脱した場合には 2 次項が必要となる．

(2) C **点と** Q_L **点と** Q_U **の予測区間を用いた 3 次項の判断**

C 点と Q_L 点と Q_U を用いた場合に，以下の判断を行う．

[A] 3 つの ID 点の全てが区間内 ⇒ 線形模型

[B] 3 つの ID 点の 1 つ以上が区間外 ⇒ 2 次以上が必要

[B1] 外れた ID 点が 1 つ⇒ 2 次で十分である．

[B2] 外れた ID 点が複数⇒以下のように判断する．

[B21] 外れの方向が全て同じ側 ⇒ 3 次は不要

[B22] 外れの方向が上下混在 ⇒ 3 次が必要

7. 原形法・区分法・差分法・多頭法

すでに説明した超設計における基本的な超構造関数は以下に示す超因子 H に関する 1 次式であった．

$$y = F(H;\mathbf{x}) = \lambda_0(\mathbf{x}) + \lambda_1(\mathbf{x})H\ ,\ \ \mathbf{x} = (x_1.\cdots, x_p) \qquad (25)$$

しかしながら，現実には様々なタイプの超構造関数が存在する．そこでこれらの様々なタイプの超構造関数のアプローチを **Fig. 5** のように大きく 4 つに分類する [11].

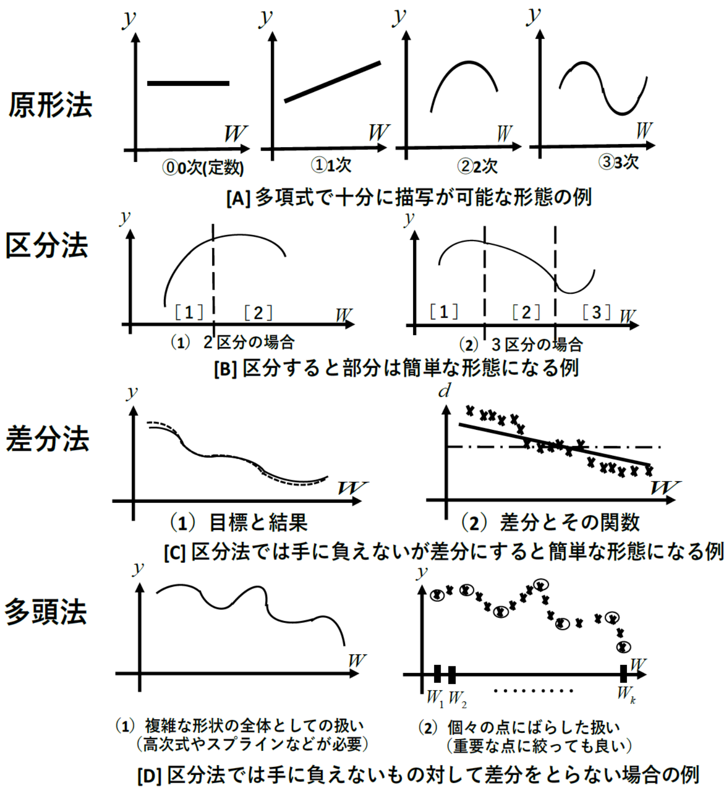

Fig. 5: 様々な超構造関数

(1) 原形法： 元データ y に対して多項式の描写をする．

(2) 区分法： 全体を区分して個々に原形法を適用する．

(3) 差分法： 差分 ($d = y -$ 目標値) に原形法を適用する．

(4) 多頭法： 個々の水準ごとに関数を求める．

7.1 原形法

対象の形状がそのまま多項式によって十分に描写（近似）が可能な場合には，そのまま模型化をすればよい．Fig. 5[A] には典型的な例を 3 次まで示している．次数は 4 次以上であっても構わない．要は多項式で描写できればその係数 (パラメータ) を用いれば設計可能である．これまでの設計の多くは 0 次か 1 次であったが，すでに 2 次 [3][16] や 3 次 [6] の事例が登場している．

7.2 区分法

1) 区分による簡易化

全体は複雑な形態でも区分すると部分は簡単な形態である場合が少なくない．多くの形は分解すると以下のものに帰着でき，高々 3 次で合成が可能である．

＊ 直線：1 次式

＊ 単調な曲線：2 次式

＊ 単調でない曲線

・ 極値（極小値, 極大値：2 次式あるいは 3 次式）

・ 変曲点：3 次式

なお，区分した場合は設計時に同時に定式化を行う．

Fig. 5[B] の (1) は全範囲を 2 次で近似するのはラフ過ぎるので 2 つに区分して各々に 2 次を当てはめればよい．Fig. 5[B] の (2) は 3 次で近似するのはラフ過ぎるので 3 つに区分して各々に 2 次，2 次，2 次とするのがよい．

ただし，区分した場合は隣接する 2 つの区間に同じ点がまたがる．前の区間では上端で，後の区間では下端となる．本来は同じ点なので，両者を等しいとする等式制約をおくか，あるいは両者の差の絶対値がある値以下という不等式制約をおけば対応が可能となる．

2) 事後の全体の合体による復元と模型化

解が得られたら全体を復元する．区分ごとの関数ではあるが模型化ができているので重要な点（始点，終点，極大値，極小値，変曲点，変化の激しい点など）について計算によりその推定値を求める．区分の境界点（隣接する 2 つの区間にまたがる点）は両方の点の平均値を用いる．このようにして推定した点の全体を近似する関数を以下の方法で求めて模型化とする．

＊多項式近似，＊スプライン近似ほか

7.3 差分法

1) 区分法でも手に負えない模型化

形態が複雑な場合には区分数がかなりの数を必要とすることになるので，場合によっては区分法では手に負えなくなる．しかし，もしも目標形態がある場合にはそれとの差分をとるということを試すべきである．目標形態が複雑なものであったとしても，差分は簡単な関数で描写できることが少なくないからである．そのような場合には差分を用いて設計を行う．そして，設計後は推定した差分に目標形態を足し込んで復元すればよい．

運悪く，差分自体がやはり複雑な場合には差分に対して区分法の適用を検討する．あるいは差分に対して次に紹介する多頭法の適用を検討とするとよい．

2) 事後の足し込みによる復元

最適化は差分に対して行われるので，結果が得られたら差分に目標形態を足し込んで復元する．この場合には区分法のように模型化は必要ない．

7.4 多頭法

1) 区分法と差分法が使えない模型化

原形法で困難な場合は目標形態がなければ区分法，目標形態があれば差分法を用いる．もし目標形態がなければ差分をとることができないので，必然的に区分法となるわけであるが，区分法でも困難な場合（多数の区分が必要な場合）には最後に多頭法を検討する．

【注】目標形態がない場合とは，おおよその考えのもとでとりあえず実験を行ってその結果を見ながら設計することを意味する．あるいは，何か（最大，最小，範囲）に関してそれを最大あるいは最小にするという場合も目標形状は存在しない．

目標形態がない場合には差分をとることができない．この場合は複雑な形態を個々の水準ごとの点にばらすしかない．多頭法の多頭とは八岐大蛇（日本神話）やヒュドラ（ギリシア神話）のように，胴（設計因子）は一つ（共通）で頭（点）が多数という意味である．これは複数の関数の集合なのでベクトル関数である．

2) 事後の全体の模型化

多頭法の設計は個別の点に関する同時定式化による設計である．これで設計を行った後で，得られた解のもとでの形態をきちんと模型化する必要がある．区分法のようにそれぞれの区間ごとの関数を用いることはできない．したがって，得られたすべての点を用いて模型化する．その方法は区分法で必要な点を推定した後の模型化と同じように以下のアプローチを用いる．

＊多項式近似，＊スプライン近似ほか

8. 極座標による陽関数化

もともと関数には以下に示す陽関数と陰関数がある．これらは 2 変数の場合で表現すると以下のようになる．

陽関数：$y = f(W)$， 陰関数：$F(W_1, W_2) = 0$

設計に用いる関数は陽関数である必要がある．ただし，陰関数の場合でも円や球などのように係数（パラメータ）で表現できる場合には問題はない．

8.1 2 次元と 3 次元の極座標描写

平面図形と立体図形は，円・楕円や球・楕円体のような典型的な場合（円錐曲線，2 次曲面）には陰関数の係数（パラメータ）を用いての設計が可能である，しかし，実際には典型的な形状ではない場合もあり，その場合には「極座標変換による陽関数化」を用いる．平面の場合には角度 θ と距離 r による (θ, r) で，立体の場合には 2 つの角度 θ, λ と一つの距離 r の (θ, λ, r) で描写できる．

8.2 極座標の原点の決め方

極描写とは極座標を用いて対象を表現することである．この場合に重要なのは原点の決定である．これには代表的なアプローチとして以下の 2 つの方法が考えられる．以下は 2 次元の場合を例として解説する．

＊A 法 その点からの距離の平方和を最小にする点
　※方針：なるべく大きな距離が出ないようにする．

＊B 法 その点からの距離の偏差平方和を最小にする点
　※方針：なるべく距離を同じ長さに揃えたい．

次に 2 つのアプローチの数理について解説する．

1) A 法（距離平方和最小）

n 個の点の距離の平方和は以下のようになる．

$$r_i = \sqrt{(W_{1i} - W_{1*})^2 + (W_{2i} - W_{2*})^2} \tag{26}$$

$$S(W_{1*}, W_{2*}) = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} \left\{ (W_{1i} - W_{1*})^2 + (W_{2i} - W_{2*})^2 \right\} \tag{27}$$

求める値は次の式を解くことで得られる．

$$\begin{cases} \partial S(W_{1*}, W_{2*}) \,/\, \partial W_{1*} = 0 \\ \partial S(W_{1*}, W_{2*}) \,/\, \partial W_{2*} = 0 \end{cases} \tag{28}$$

この解はよく知られているように以下のようになる．

$$W_{1*} = \textstyle\sum_{i=1}^{n} W_{1i} \,/\, n = \bar{W}_1 \,, \quad W_{2*} = \textstyle\sum_{i=1}^{n} W_{2i} \,/\, n = \bar{W}_2 \tag{29}$$

これは平均を求めればよいので簡単に計算ができる．

2) B 法（距離の偏差平方和最小）

ここで求めたいものは以下の最適化の解である．

$$f(W_{1*}, W_{2*}) \;=\; \sum_{i=1}^{n} (r_i - \bar{r})^2 \;\;\rightarrow\;\; \text{最小}$$

$$r_i = \sqrt{(W_{1i} - W_{1\sharp})^2 + (W_{2i} - W_{2\sharp})^2} \qquad (30)$$
$$\bar{r} = \sum_{i=1}^{n} r_i \,/\, n$$

上記のものを書き直すと以下のようになる．

$$f(W_{1\sharp}, W_{2\sharp}) = \sum_{i=1}^{n} \left\{ \sqrt{(W_{1i} - W_{1\sharp})^2 + (W_{2i} - W_{2\sharp})^2} - \left(\sum_{i=1}^{n} \sqrt{(W_{1i} - W_{1\sharp})^2 + (W_{2i} - W_{2\sharp})^2} \right) / \, n \right\} \;\rightarrow\; \text{最小} \qquad (31)$$

これは少し込み入った計算になるが数値計算で求めることができる．この方法は円や球が欠けた形の場合でもその中心・重心が解となる点で優れている．なお，A 法と B 法は各々の特徴を踏まえて使い分けるとよい．

9. 高次の超構造関数

9.1 高次の超構造関数の事例

超構造関数が高次の関数になった場合には高度な設計を行う必要があり，例えば以下の実例がある．

＊2 次の実例：射出成型の事例 [3]

＊3 次の実例：水晶発振子の事例 [6]

これらの場合は超構造関数の偏微分の活用が重要になる．

9.2 超構造関数が 2 次の場合

1) 頑健設計への応用

超構造関数は多変数関数を超因子 H に注目して構成した多項式なのでこれを H で偏微分することができる．

$$y = F(H; \mathbf{x}) = \lambda_0(\mathbf{x}) + \lambda_1(\mathbf{x})H + \lambda_2(\mathbf{x})H^2 \qquad (32)$$

$$\partial y / \partial H = \partial F(H; \mathbf{x}) / \partial H = \lambda_1(\mathbf{x}) + 2\lambda_2(\mathbf{x})H \qquad (33)$$

この関数の絶対値を 0 に近付ければ H の影響は減衰できる．ただし H 自身もその水準が決定され，それは使用時・操業時の H の目標水準（この前後に H の値がばらつく）となる．これは非線形を応用した頑健設計であるが，この場合の H は直積実験で外側に割り付けることもできるし，内側に割り付けることもできる．実務的には，外側に割り付けたら外乱（撹乱因子）で内側に割り付けたら内乱（ばらつく設計因子）である．なお，内側に割り付けた場合には，他の撹乱因子を外側に外乱として割り付けること（複合型頑健設計）もできる．

なお，超構造関数が 1 次式の場合には設計因子がなければ H のばらつきの減衰はできないが，高次式（非線形）の場合には設計因子がなくても H のばらつきの減衰はできる．設計因子があれば，それらに超因子 H が加わってかなりパワフルな減衰が可能になる．

2) 最大，最小，範囲への応用

2 次関数の場合は極値が一つなのでこれを求めた上でその合成関数を用いて設計する．例えば極値を含むある区間の中の最大や最小を求める場合には以下の合成関数（極値の横座標を p と表現）を用いればよい．

$$y = F(H; \mathbf{x}) = \lambda_0(\mathbf{x}) + \lambda_1(\mathbf{x})H + \lambda_2(\mathbf{x})H^2 \qquad (34)$$

$$p = -\lambda_1 / (2\lambda_2),$$
$$p = p(\mathbf{x}),\;\; \lambda_1 = \lambda_1(\mathbf{x}),\;\; \lambda_2 = \lambda_2(\mathbf{x}) \qquad (35)$$

区間 m_L〜m_U での最大と最小は以下の合成関数となる．

$$\begin{aligned} y_{Max}(\mathbf{x}) &= \mathrm{Max}\left\{F(p(\mathbf{x}); \mathbf{x}), F(m_L; \mathbf{x}), F(m_U; \mathbf{x})\right\} \\ y_{Min}(\mathbf{x}) &= \mathrm{Min}\left\{F(p(\mathbf{x}); \mathbf{x}), F(m_L; \mathbf{x}), F(m_U; \mathbf{x})\right\} \end{aligned} \qquad (36)$$

$$y_{Ran}(\mathbf{x}) = y_{Max}(\mathbf{x}) \;-\; y_{Min}(\mathbf{x}) \qquad (37)$$

両者の差である範囲はばらつきを意味し，これを小さくすることは一つのタイプの頑健設計である．設計はこれらを用いて数理計画法で求解すればよい．

9.3 超構造関数が 3 次の場合

超構造関数が 3 次関数の場合 [6] もある．その場合には極大値，極小値そして曲変点が設計の対象となる．

$$y = F(H; \mathbf{x}) = \lambda_0(\mathbf{x}) + \lambda_1(\mathbf{x})H + \lambda_2(\mathbf{x})H^2 + \lambda_3(\mathbf{x})H^3 \qquad (38)$$

3 次関数で極大点と極小点がある場合には各々の横座標は以下の式で与えられる．添字の R は区間の右側の値 p_R，添字の L は区間の左側の値 p_L を意味し，それぞれはいずれも合成関数でその実体は設計因子の関数である．なお，変曲点は二種類の極値の中点なので省略する．

$$\begin{aligned} & p_R, p_L = \left(-\lambda_2 \pm \sqrt{\lambda_2^2 - 3\lambda_3\lambda_1}\right) / (3\lambda_3) \\ & p_R = p_R(\mathbf{x})),\;\; p_L = p_L(\mathbf{x}), \\ & \lambda_1 = \lambda_1(\mathbf{x}),\;\; \lambda_2 = \lambda_2(\mathbf{x}),\;\; \lambda_3 = \lambda_3(\mathbf{x}) \end{aligned} \qquad (39)$$

3 次関数の設計では代表的なものに以下のものがある．

＊極値間左右差，＊極値間上下差，＊極値間勾配

これらは以下の多重合成関数として扱えばよい．

極値間左右差：$p_U(\mathbf{x}) - p_L(\mathbf{x})$

極値間上下差：$F(p_U; \mathbf{x}) - F(p_L; \mathbf{x}) \qquad (40)$

極値間勾配：$(F(p_U; \mathbf{x}) - F(p_L; \mathbf{x})) / (p_U(\mathbf{x}) - p_L(\mathbf{x}))$

多重合成関数は設計の可能性と幅を大きく広げる．

10. おわりに

本稿は超因子というものを指定して超構造関数を構成し，これを用いて数理計画法で最適化するという枠組みで行う設計である超設計のパラダイムとメソドロジーについて議論した．これは様々な設計を統一的に整理することができるとともに新たな設計の可能性を広げるものであることを示した．近年注目されている頑健設計は超設計の中の一部のタイプとして包含しており，従来の設計は超設計の退化した形（超因子がし存在しない場合）として包含している．

超設計は広範で高度な設計目的に対応が可能である．さらには，様々な複雑な形状・状態を極座標変換で描写する方法も示した．そして確実に設計目的を実現するための一括りの設計法である包括設計法についても論じた．

本稿は超設計のパラダイムとメソドロジーの全体を俯瞰する概略的な議論を行った．個々の内容については細かなロジックとメソドロジーが用意されており，それらの詳細には参考文献を参照されたい．

参考文献

[1] Box, G.(1991), “Teaching Engineers Experimental Design with a Paper Helicopter”, Quality Engineering, 4(3), pp.453-459.

[2] 芳賀敏郎，竹内 啓，奥野忠一 (1976)：“重回帰分析における変数選択の新しい規準”，品質，6(2)，pp.35-38.

[3] Miller, A. and Wu, C.F.J. (1996): “Parameter Design for Signal -Response Systems: A Different Look at Taguchi’s Dynamic Parameter Design”, Statistical Science, 11, pp.122-136.

[4] 宮川雅巳 (2000)：「品質を獲得する技術」, 日科技連出版社.

[5] Myers R. H., Montgomery D.C., and Anderson Cook C. M., (2009): Response Surface Methodology: Process and Product Optimization Using Designed Experiments, Wiley, New York.

[6] 高橋星太, 濱口勝重, 高橋武則 (2015): “HOPE 手法を用いた水晶発振子の設計パラメータの最適化”, JSQC 第 109 回研究発表会発表要旨集，pp.39-42.

[7] Takahashi, T. and Saito A. (2005): “Education of Robust Parameter Design by Twin Rotor Paper Helicopter”, Proc. of International Conference on Quality ‘05, Tokyo, CD proceeding, pp.1-12.

[8] Takahashi, T. (2015): “Proposal of Flexible Design and its Application”, Proc. of the Asian Network for Quality Congress 2015 in Taipei, CD proceeding, pp.1-10.

[9] 高橋武則 (2017) : “超構造関数による柔軟設計”, 日本品質管理学会第 113 回研究発表会発表要旨集，pp.157-160.

[10] Takahashi, T. (2017): “Hyper Design based on Hyper Factors”, Proc. of the Asian Network for Quality Congress 2017 in Kathmandu, CD proceeding, pp.1-12.

[11] 高橋武則 (2019): “描写因子による形態設計と極座標描写”, 目白大学研究紀要, 第 17 号, pp.19-34.

[12] 高橋武則 (2019) : “調査・実験の教育ための概念図と特性要因図と構造模型表および変数計量化”, 日本品質管理学会第 119 回研究発表会発表要旨集，pp.149-152.

[13] 椿 広計（2006）:「ビジネスへの統計モデルアプローチ」，朝倉書店.

[14] 椿 広計（2006）： “統計科学の横断性と設計科学への寄与”，「横幹」, 1[1], pp.22-28.

[15] 椿 広計, 河村敏彦（2007）:「設計科学におけるタグチメソッド」, 日科技連出版社.

[16] Wu, C. F. J. and Hamada, M.（ 2009）: Experiments: Planning, Analysis, and Optimization(2nd ed.), Wiley, New York.

[17] 山田 秀（2004）:「実験計画法 —方法論—」，日科技連出版社.

[18] 吉野 睦，仁科 健 (2009): 「シミュレーションと SQC」，日本規格協会.

高橋 武則

早稲田大学法学部卒業，同大学理工学部卒業，早稲田大学大学院理工学研究科修士課程修了，同研究科博士課程修了．東京理科大学工学部助教授，同大学同学部教授．慶應義塾大学大学院健康マネジメント研究科教授．目白大学経営学部教授を経て，2017 年より慶應義塾大学大学院健康マネジメント研究科客員教授．専門は品質管理，特に実験計画法・超設計とTQM（Total Quality Management）．工学博士．