



RIETI Policy Discussion Paper Series 10-P-016

設計比較優位説のプロセス的基礎

藤本 隆宏
東京大学



Research Institute of Economy, Trade & Industry, IAA

独立行政法人経済産業研究所

<http://www.rieti.go.jp/jp/>

設計比較優位説のプロセス的基礎

藤本隆宏（東京大学）

要 旨

本論では、筆者が理論的・実証的な考察を加えてきた「設計の比較優位仮説」に対して、設計プロセス論的な基礎を付け加えることを試みる。具体的には、公理系設計論をベースに、設計行為を、不確実性下で製品機能・製品構造の連立方程式を解く、2 段階設計プロセスによって近似する。第 1 段階は、構造・機能の因果知識が不完全な中での暫定設計解の導出、第 2 段階は、その暫定設計解から、最適設計解へと漸近する試行錯誤のプロセスである。このモデルによって、アーキテクチャ、組織能力、設計調整プロセス、市場ニーズなどを定式化することにより、統合型の組織能力の遍在する日本の設計拠点が、インテグラル・アーキテクチャの製品のリードタイム競争で優位性を持つこと、および、同じインテグラル型でも、科学的知識の獲得を必要とするタイプの製品では日本企業が設計の競争優位を持てるとは限らないことを論理的に推測し、シミュレーションによって確認する。

RIETI ポリシー・ディスカッション・ペーパーは、RIETI の研究に関連して作成され、政策をめぐる議論にタイムリーに貢献することを目的としています。論文に述べられている見解は執筆者個人の責任で発表するものであり、(独)経済産業研究所としての見解を示すものではありません。

1 はじめに

筆者はこれまで、経済系の貿易論と工学系の設計論を結合した「設計（アーキテクチャ）の比較優位論」を模索してきた（藤本、2001a、2003a、2003b、2007、他）。具体的には、設計調整能力に優れた現場、そしてそうしたタイプの現場の多い国・地域・産業・企業は、設計調整を多く要する製品を得意とし、それを輸出する傾向がある、と主張した。

「設計調整能力の豊富な国は設計調整集約的な産業で比較優位を持つ」という本説の論理構成は、標準的な新古典派貿易論が説く「労働力（資本）の豊富な国は、労働（資本）集約的な産業で比較優位を持つ」という有名な命題と基本的な論理構造は同型である。また、開発現場の相対生産性から比較優位を論じる「比較設計費」の考え方は、リカード貿易論の「比較生産費」概念を設計領域に援用したものといえる。

また、技術格差貿易説・プロダクトサイクル説・雁行形態論に対しては、「技術後発国の現場群の組織能力と当該製品のアーキテクチャ変化の間の適合関係が、産業の国際移転の速度ないし時差に影響する」という命題でこれらの説を修正できる。規模の経済・製品差別化・企業の非同質性を導入した「新・新貿易論モデル」に対しては、初期生産の立地は偶然決まるという命題に代えて、「設計立地が初期生産立地を決める」という論理を導入して『設計の比較優位論につなげれば、設計立地と生産立地のその後の関係は「新・新貿易論モデル」に従う、という形で「微細な産業内貿易」を説明しても良い。新貿易論ならば「設計地で累積的に生産が続く」という結論になろうし、新・新貿易論ならば、「当初の設計地で生産が続くか否かは当該現場の生産性および当該財の輸送費・販売固定費次第だ」というさらにリアルな結果を得る。このように「設計の比較優位説」は、従来のどの貿易論も全否定するものではなく、むしろ補完的に使うことで、現代の貿易現象をよりよく説明できる枠組みではないかと筆者は考える。

現場に戻った設計比較優位説へ：しかしながら、これまで展開してきた「設計の比較優位」概念は、製品全体の「マクロ・アーキテクチャ」（調整負荷）と、産業全体の「ものづくり組織能力」（調整能力）の適合関係を論じている点において、経済学的にはミクロであっても、設計論や組織論としてはマクロ的である。そこで本論では、ものづくり現場論のレベルに立ち戻り、「設計プロセス」の視点から、組織能力とアーキテクチャの適合関係について、ミクロ的な考察を加えてみたい。なお、以下の考察の多くの部分は、藤本・大隈（「設計立地の比較優位に関する試論—枠組・実証・シミュレーション—」RIETI Discussion Paper 07-J-025）に依拠している。

2 設計プロセスとリードタイム

2-1 連立方程式モデルとしての設計プロセス

連立 1 次方程式でのモデル化: 次に、設計プロセスが、どのようにしてパフォーマンス、とりわけ設計のリードタイムや生産性に結びつくかを、簡単なモデルを用いて検討する。ここでは、すでに概要を紹介した、①公理系設計論をベースとした連立方程式モデル、②試行錯誤で構造設計案のサーチと機能評価を繰り返す問題解決モデル、そして③両者を結合した「2 段階設計モデル」の 3 アプローチについて、設計プロセスとリードタイムを結ぶ論理を示してみたい。

このロジックを簡単に説明するため、工学領域で展開されてきた設計論、とりわけ公理系設計論の枠組 (Suh, 1990, 2001) の考え方を取り入れる。すなわち、設計パラメータ間の関係を (単純化のため) 線形と仮定し、製品機能パラメータ群を従属変数、製品構造パラメータ群を独立変数とする連立一次方程式を解くのが製品設計活動だと考えてみよう¹。

まず、ある組立製品 (例えば自動車、PC など) を考えてみよう。そして、顧客が要求する主要な機能要素 (自動車ならば燃費、安全性、乗り心地など) を代表する機能パラメータが n 個あるとしよう²。一方、製品の構造は、例えば自動車であれば 10 にも 100 にも 1000 にも分解できるが、できるだけ機能との対応関係が明確になるように、 n 個の構造要素 (部品など) に分解することは不可能ではなかろう。そして、各構造要素の本質的な特性を示す構造パラメータ (部品寸法など) もひとつずつ、合計 n 個見つかるとうしよう。

以上の定式化が出来たとするならば、この製品に関する構造設計プロセスを、「 n 個の機能要件パラメータ (n 本の方程式) を同時に満たす n 個の未知数 (構造パラメータ) の値を求める、連立 1 次方程式の解を求めるプロセス」として近似的に示すことができる。これを機能ベクトル= \mathbf{y} 、構造ベクトル= \mathbf{x} 、機能・構造対応係数マトリックス= \mathbf{A} で表現するならば、当該人工物の構造・機能間の因果関係は、以下のように示せる (大隈・藤本, 2006)。

¹ ここでは、所与の機能パラメータから構造パラメータを導く製品設計に議論を絞るが、工程設計に関しては、同様の定式化を行い所与の構造パラメータ (あるいは機能パラメータ) から工程パラメータを導くという形に変数を入れ替えれば、同じ枠組みが使える。

² 機能パラメータ n をいくつに設定し何を選ぶかは、実際には難しい問題だが、あくまでも利用者にとって意味のある機能要素、利用者が当該製品を購入するときに考慮する主要な機能に絞るべきである。この分析の趣旨から考えるならば、設計者の論理で、利用者が理解しにくい工学的・技術的な機能パラメータを選ぶべきではない。ひとつの目安としては、消費者の許容価格に影響を与える機能パラメータ (たとえばヘッドニック価格モデルで有意な係数を伴う機能変数群) を選ぶというのが一手である。

$$\mathbf{y} = \mathbf{A} \cdot \mathbf{x}$$

\mathbf{y} : 機能要素 (機能パラメータ) のベクトル

\mathbf{x} : 構造要素 (構造パラメータ) のベクトル

\mathbf{A} : 機能・構造の対応関係を示すマトリックス

(1)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{m1} & & & a_{mm} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

ここで、正方行列である係数行列 \mathbf{A} 中の係数 a_{ij} は、機能パラメータ y_i に対する、構造パラメータ x_j の正味の影響を表す。つまり、全体として係数行列 \mathbf{A} は、当該人工物に関する工学的な因果知識を表現している。たとえば、機能パラメータ y_1 の値は、

$$y_i = a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \cdots + a_{in} \cdot x_n$$

という形でリニアに近似される。

仮に a_{ij} が工学的に既知で正確であれば、構造設計者は、構造パラメータ $\{x_1, x_2, \dots, x_n\}$ の値を主体的に決めることにより、その結果としての機能パラメータの値 $\{y_1, y_2, \dots, y_n\}$ を事前に知ることが出来る。逆に、顧客や機能設計者から、要求機能の値 $\{y_1^*, y_2^*, \dots, y_n^*\}$ が与えられているなら、構造設計者は、この連立方程式を解くことによって、設計目標とする構造パラメータ $\{x_1^*, x_2^*, \dots, x_n^*\}$ を事前に決められる。前述の行列表現とするなら、 $\mathbf{x}^* = \mathbf{A}^{-1} \mathbf{y}^*$ である (ただし \mathbf{A}^{-1} は逆行列、 \mathbf{y}^* は外から与えられる要求機能パラメータのベクトル、 \mathbf{x}^* は設計者が選択する最適な構造設計パラメータのベクトル)。

ただし、後述のように、現実の設計活動では、このような完璧な因果知識は存在せず、設計者は、市場や技術の不確実性に直面しつつ、現場で試行錯誤を繰り返す必要がある。設計プロセスの多くの部分は、逆行列や逆関数を求める計算過程ではなく、むしろ、前述のように、問題解決サイクルというヒューリスティックで、よりリアルに説明できることも忘れてはならない。この点については「2段階設計モデル」の項で再説する。

アーキテクチャの行列表現 : この表現を用いるなら、純粋なモジュラー型に対応するマトリックス \mathbf{A} は、対角線の要素以外はゼロとなる対角行列となる。一方、純粋なインテグラル型は、全ての要素が非ゼロの密行列で表現できる。

図1 公理系設計論を援用したアーキテクチャの表現

モジュラー型		インテグラル型
$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & & \\ \vdots & & \ddots & \\ 0 & & & a_{mm} \end{bmatrix}$	$A =$	$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{m1} & & & a_{mm} \end{bmatrix}$

行列 A が、対角線以外はゼロの対角行列であるなら、それに対応するアーキテクチャは純粋なモジュラー型であり、それは「ある機能パラメータ y_i は、ただ一つの構造パラメータで x_i で決まる」ということである。つまり、この人工物に関する因果知識は、 n 本の互いに独立な単独方程式 $y_i = a_{ii} \cdot x_i$ (一般には $y_i = f(x_i)$) に完全に分解できる。あとは、個々の式を解けばよい。

一方、行列 A がゼロでない数字で埋まった密行列であるなら、それに対応するアーキテクチャは純粋なインテグラル型であり、解を求めるには、本格的な連立方程式を解く必要がある。むしろ、現実の人工物のアーキテクチャは、対角行列と密行列の間にあるものが多く、それらが、アーキテクチャ・スペクトルのさまざまな位置に展開することになる。

連立方程式を解く (インテグラル型): このように、ある製品あるいはその部位がインテグラル型なら、その設計プロセスは式が n 本で未知数が n 個の連立方程式を解く問題で近似される。人工物が複雑になり、 n が大きくなれば、その解の収束に要する試行錯誤数は急増する³。上記の例の場合、 n 人の構造設計者 (X_i) と n 人の機能設計者 (Y_i) は、すべてがすべてと連絡を取り合い、互いのパラメータを調整する必要があり、その作業負荷は大きい⁴。

このため、インテグラル・アーキテクチャの設計プロセスにおいては、設計の調整負荷の軽減 (連立方程式の計算量の節約) が、モジュラー型製品の場合 (後述) 以上に重要になる。これに有効なルーチンとしては、①調整 1 回あたりの負荷を減らすことと、②設計解の収束に至る調整の回数を減らすこと、この 2 つが考えられる。調整 1 回あたりの負荷を減らすルーチンとしては、当事者間の事前情報共有、コロケーションによる距離の短縮、

³ 連立 1 次方程式の数値解法としては、Jordan の掃き出し法、Gauss の消去法、Cramer の公式 (逆行列) などが知られるが、比較的計算量の少ない消去法や掃き出し法でも、変数が n 個の連立方程式の計算量 (加減乗除回数) は n^3 に比例するとされる。ただし、個々の加減乗除動作をひとつずつカウントする計算量の式で設計の調整負荷を代理させるべきかどうかは議論の余地がある。

⁴ 同様に、 A が三角行列なら、各パラメータを担当する設計者の間に階層性が現れ、相互調整というより

コミュニケーション技術の導入などがありうる。調整回数削減のためのルーチンとしては、収束条件の緩和（機能目標の引き下げ）といった消極的な方策もあるが、例えば複数の 1 対 1 コミュニケーションを 1 回で済ませる大部屋方式などもある。

概して、「統合型組織能力」を構成するチームワーク重視の設計ルーチンが、インテグラル型設計の調整負荷低減には有効であることが示される（Clark and Fujimoto, 1991; 延岡、1996）。

単独方程式を解く（モジュラー型）：これに対し、開発される人工物が純粋にモジュラー型であれば、その設計は、互いに独立の n 本の単独方程式を解く問題で近似される。

一般に、個々の設計者（設計タスク）が、個々の構造要素（部品など）の設計パラメータを収束させるという役割を担っているとするなら、機能・構造設計要素間の相互依存性が小さいモジュラー寄りの製品は、各タスクの収束作業を他と比較的独立的に行うことができる（Alexander, 1964; Simon, 1969; Ulrich, 1995）。

仮に、各構造設計要素 x_i を担当する構造設計者 X_i が n 人、各機能設計要素 y_i を担当する機能設計者（例えば実験エンジニア） Y_i が n 人いる設計組織がこの設計を担当するとしても、構造設計者 X_i は、他の構造設計者 X_j とコミュニケーションをとる必要は無く、対応する機能パラメータを担当する実験エンジニア Y_i とペアで仕事をすれば担当分の構造設計は完了する⁵。モジュラー型の設計プロセスは、コミュニケーション節約的である。

計算量としての設計リードタイム：以上、公理系設計論を応用した「連立方程式のアナロジー」から、ある人工物の設計に関して、設計パラメータを収束させるのに要するリードタイムは（他の条件が一定のとき）、インテグラル型で長く、モジュラー型で短くなる可能性が高いことが容易に推定できる。

実際、連立方程式を解くのに要する時間計算量を、設計リードタイムあるいは設計工数と見立てるならば、本格的な連立方程式を解くインテグラル型と、単独方程式の束を解けばよいモジュラー型では、計算量の差は歴然としている。特に設計要素の数（ n ）の多い複雑な人工物の場合、計算量の差は広がる。本格的な連立方程式を解くための計算量は（掃き出し法や消去法など効率的な解法を採ったとしても）、 n の 3 乗に比例することが知られているが、これに対し n 本の単独方程式を解く計算量は n に比例するのみである。

リアリティの限界：このように、設計プロセスを「連立方程式を解くプロセス」に擬えるというアイデアは、設計プロセスの分析に多くの本質的な洞察をもたらす。

は、一方的な指示に近い関係が設計者間に現れる。

⁵ これは、Cusumano and Selby (1995) が記述したマイクロソフト社の開発組織、すなわちプログラマーとテストエンジニアがペアで仕事をする体制（異なるペア間は比較的独立）と整合的である。

しかし反面、設計現場を観察した上でいうなら、目標の人工物に関する構造・機能関係の因果知識、つまり行列 A の中身が完全であることを前提に、机上で数式を解くように設計解を得るというアナロジーは、日々苦心する多くの設計者からみれば、違和感のある像でもあろう。実際の設計プロセスの、少なくともある部分は、因果知識つまり行列 A の中身が不完全であることを前提に、試行錯誤で切り抜けるものであろう。

その意味で、設計者の完全情報を前提に、「連立方程式モデル」で実際の設計プロセスを模写することには、一定の限界がありそうだ。この点を補う工夫のひとつが、たとえば後述の「2段階設計モデル」である。

2-2 問題解決モデルとしての設計プロセス

漸進設計と並行設計：人工物の構造・機能に関する因果知識が不完全であると仮定し、限定的合理性下での問題解決プロセスとして設計プロセスを描写する、という考え方 (Fujimoto, 1989; Clark and Fujimoto, 1991) は、すでに示した。設計に関するものづくり能力とは、第一義的には、組織の問題発見・問題解決能力であると本稿では主張する。

これを、モデル化する方法としては、大きく分けて、「漸進設計」(コーディネーション法) と「並行設計」(コンペティション法) という2つのアプローチがありうる⁶。

①「漸進設計」は、1つの人工物の構造設計案に対して、設計者が相互に調整しつつ徐々に構造設計を改善し、機能の向上を目指すプロセスであり、比較評価される代替案は「既存設計案」と「新規(微修正)設計案」である。

②「並行設計」では、複数の独立した構造代替案を並行開発し、その製品機能を競わせ、勝った方を採用する。常に複数の代替的な新規設計案が機能性を競う。いわば設計案のトーナメントだが、多数の代替案から一気に選抜するか、1対1のコンペを1回戦、2回戦と繰り返すかなど、いくつかの方法がある。以下、具体的に見ていこう。

漸進設計(「蜘蛛の巣モデル」の例)：「漸進設計」(コーディネーション法)における設計案のサーチとシミュレーションは、実際には多様な方法で行われうるが、最も複雑なインテグラル型製品の場合、構造設計要素を1回に1箇所ずつ修正し、修正前の既存設計と比較し、機能が改善されれば修正案を採用、さもなければ既存案を採る、というインクリメンタル(漸進的)な方法が、最も手堅い。

ここで、公理系設計の定式化に従い、機能設計要素の目標が2つ (y_1^*, y_2^*)、構造設計要素が2つ (x_1, x_2)、各々の構造設計パラメータに関する専門設計担当者も2名 (X_1 と X_2 、例えばボディ設計者とエンジン設計者) という単純な例を考えるなら、設計者 X_1 と X_2 は、

互いに相手からの設計変更情報 \mathbf{x}_j に応じて相互に自らの担当部分 \mathbf{x}_i を修正し、それを全体構造に組み込んでシミュレーション（試作・実験）を行い、その結果としての機能目標 $\mathbf{y}^* = (\mathbf{y}_1^*, \mathbf{y}_2^*)$ に近づけるよう協調して修正を繰り返していく設計現場を想定できる。

しかし、2人が同時に設計変更を行えば、互いの設計変更の前提条件が変わってしまうので、かえって混乱を招く恐れがある。そこで、碁や将棋のように、設計変更ができるのは交互に1手ずつ、という設計順序（シーケンス）のルールを決め、設計者 \mathbf{X}_1 と \mathbf{X}_2 が交互に設計変更を進め、その都度機能向上の有無をチェックする、というのが、時間はかかるがもつとも無難である。

これをモデル化した例としては、たとえば、経済学で「蜘蛛の巣モデル」と呼ばれる、以下のようなものがある（大隈・藤本、2006）。

第0期：設計の初期値 $\mathbf{x}^0 = (\mathbf{x}_1(0), \mathbf{x}_2(0))$ が与えられる（たとえば既存製品）

第1.1期：まず設計者 \mathbf{X}_1 が、相手設計者の初期値 $\mathbf{x}_2(0)$ 、機能構造の因果知識 $\mathbf{A} = \{\mathbf{a}_{ij}\}$ を与えられたものとして、ひとつの機能目標 \mathbf{y}_1^* に対する局所的な解を求め、これを $\mathbf{x}_1(1)$ とする⁷。

第1.2期：第1.1期に設計者 \mathbf{X}_1 が設計変更した構造パラメータ $\mathbf{x}_1(1)$ を所与として、もうひとりの \mathbf{X}_2 が、相手設計者の変更値 $\mathbf{x}_1(1)$ 、機能構造の因果知識 $\mathbf{A} = \{\mathbf{a}_{ij}\}$ をもとに、もうひとつの機能目標 \mathbf{y}_2^* に対する局所的な解を求め、これを $\mathbf{x}_2(1)$ とする。

：

第t.1期：1.1期と同様に、 \mathbf{X}_1 が、相手の前期値 $\mathbf{x}_2(t-1)$ 、機能構造の因果知識 $\mathbf{A} = \{\mathbf{a}_{ij}\}$ 、機能目標 \mathbf{y}_1^* に対する局所的な解を求め、これを $\mathbf{x}_1(t)$ とする。

第t.2期：1.2期と同様に、 \mathbf{X}_2 が、相手の変更値 $\mathbf{x}_1(t-1)$ 、機能構造の因果知識 $\mathbf{A} = \{\mathbf{a}_{ij}\}$ 、機能目標 \mathbf{y}_2^* に対する局所的な解を求め、これを $\mathbf{x}_2(t)$ とする。

設計収束：以下、同様に繰り返し、機能の測定値 $(\mathbf{y}_1(t), \mathbf{y}_2(t))$ が、目標値 $(\mathbf{y}_1^*, \mathbf{y}_2^*)$ に十分に接近したと判断されたら、設計の収束・完了を宣言する。そこまでの所要時間が設計リードタイムとなる。

このモデルは、① 一度に1設計者しか設計変更できない、② 構造要素 i を担当する構造

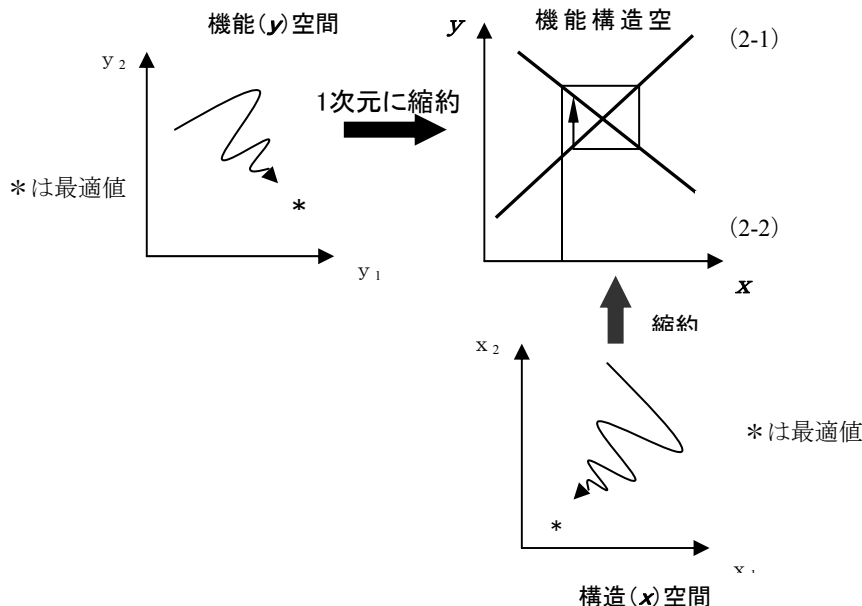
⁶ 以下の議論は、基本的に大隈・藤本（2006）をベースとする。

⁷ 具体的には、単独方程式 $[\mathbf{y}_1^* = \mathbf{a}_{11} \cdot \mathbf{x}_1 + \mathbf{a}_{12} \cdot \mathbf{x}_2(0)]$ を \mathbf{x}_1 について解き、その解を $\mathbf{x}_1(1)$ として $\mathbf{x}_1(0)$ をこれに置き換える。

設計者 X_i は、対応する機能要素 i の目標 y_i^* にのみに集中してベストと考える設計変更を行う、というシンプルなルールに基づいている。例えば、機能目標 y_i^* が {安全、燃費、馬力}、構造要素 x_i が {車台、車体、エンジン} であるとき、すべての構造要素がすべての機能要素に影響することは明らかだが、あえて車台設計者は安全、車体設計者は燃費、エンジン設計者は馬力の改良に集中するように上から指示する。そして、一人の設計者の設計変更中は、他の2人には設計変更を禁ずる。

以上の漸進設計モデルは、経済学では「蜘蛛の巣モデル」(例えば価格・数量調整の部分均衡モデルに用いられる)と同型である(図2)。

図2 漸進設計の「蜘蛛の巣モデル」



資料: 大隈・藤本 (2006) にもとづく。ベクトル x, y がスカラー変数に縮約できることが仮定されている。

以上の漸進設計モデルは、すべての設計パラメータが相互依存するインテグラル型アーキテクチャの人工物を暗に想定している。ゆえに、複数の設計者が良かれと思い、同時に行った設計変更が、相互作用により機能低下を招くなどの混乱を避けるため、1人が設計変更している時は、他の設計者は設計変更を禁じられる。これは、設計者の待ち時間増加、すなわち設計リードタイムの長期化、設計生産性の低下につながり、その面では望ましくないが、漸進設計により確実に製品機能を高めるための安全策ともいえる。筆者らの実態調査によれば、これは、実際の設計活動でもしばしば見られるパターンであるようだ。

大部屋設計による調整速度アップ：これに対し、大部屋でお互いの設計者の動きを可視化し、互いの設計意図が予測可能な状況を作れば、たとえば1度に3人の設計者が、サッカー・チームの同僚のように互いの動きを予測・報告し合いながら、設計変更を同時に破綻なく遂行することは可能かも知れない。その場合、漸進設計の調整スピードが3倍に上がったとみなすことができよう。

例えば、長期雇用・大部屋設計・チーム設計を特徴とする日本の多くの開発チームが示すリードタイムの短さは、こうした、設計者同士の可視化がひとつの要因と考えられる。

モジュラー設計と「蜘蛛の巣モデル」：以上の分析は、すべての設計要素が相互作用を持つ「インテグラル型アーキテクチャ」が前提となっているが、これがモジュラー製品であれば、結果は自明で、設計はすぐに終わる。

すなわち、構造要素 i は、対応する機能要素 i に関して機能完結的であるから、構造設計者 \mathbf{X}_i が、機能要素 i の目標 \mathbf{y}_i^* を達成すべく \mathbf{x}_i の設計変更を行えば、解は1回で最適値 \mathbf{x}_i^* に収束する。慎重を期して、設計者が一人ずつ設計変更をしたとしても、 n 期で設計は収束する。いずれにしても、よほど機能要求水準が低くない限り、インテグラル型製品の設計リードタイムは、同様のモジュラー型製品より、かなり長くなると予想される（後述のシミュレーションで確認する）。

並行設計モデル：試行錯誤的な問題解決アプローチのもうひとつの例は、並行設計モデル（コンペティション型の接近）である（大隈・藤本、2006）。

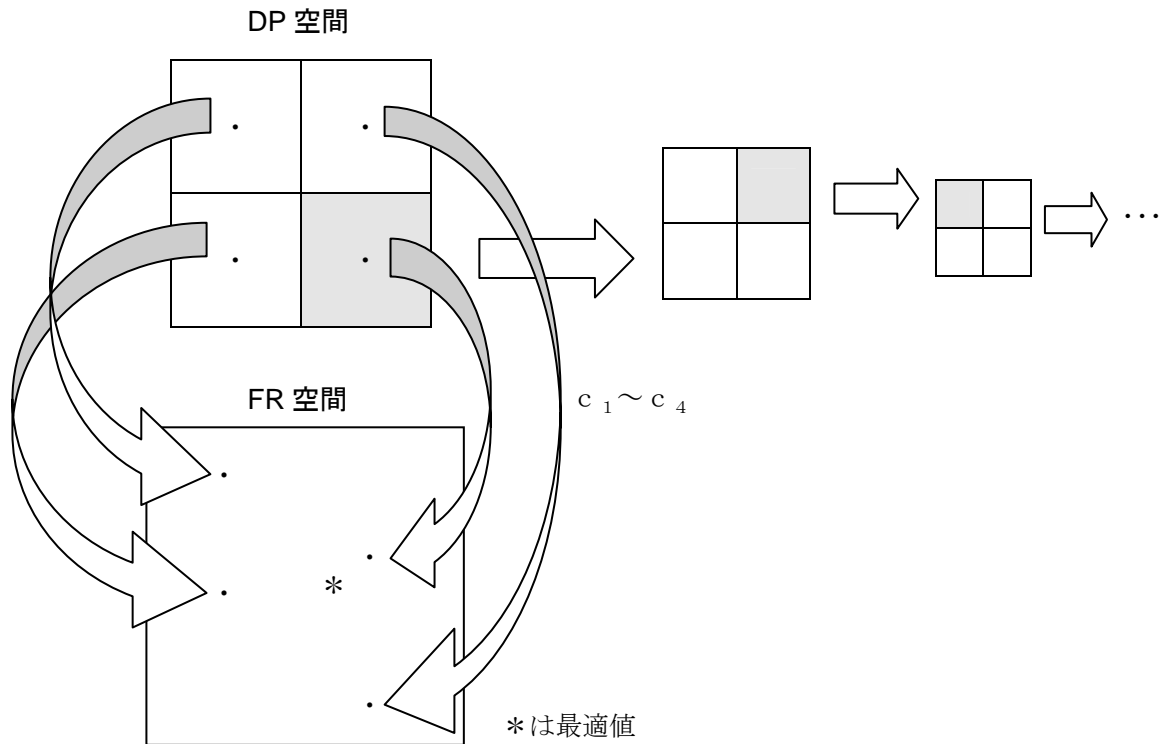
単純化のため、前項と同様の人工物を想定し、2次元の構造設計空間 $(\mathbf{x}_1, \mathbf{x}_2)$ と2次元の機能設計空間 $(\mathbf{y}_1, \mathbf{y}_2)$ でこれを表そう。この設計空間上で、並行開発された複数の構造設計案を競争（コンペティション）させ、要求機能 $(\mathbf{y}_1^*, \mathbf{y}_2^*)$ に最も近い結果を出した設計案を選択する。

そして、勝ち抜いた設計案の枠の中で、さらに細かく分けた設計案を並行開発し、2回戦を行い、機能的に優れたものを選抜する。こうした一種のトーナメント（多段階選抜）によって、最適解に漸近するとみなすのが、並行開発のひとつの定式化の方法である⁸。

具体的な例として、後述のシミュレーションで採用した「4分割法」（図3）を説明する（大隈・藤本、2006）。

⁸ 通常のトーナメントと異なるのは、多段階選抜の仕方である。並行開発モデルが想定するのは、強い地域を選抜して、その地域でまた選抜を行う、という考え方で、例えば高校野球なら、まず東西対抗を行い、仮に東が勝ったら東日本で地区対抗戦を行い、仮に東北が勝ったら東北地方内の県対抗、次は優勝県内の学校対抗…という手順で優勝校を決めるわけである。

図3 並行開発モデルの例（4分割法）



- (1) 構造設計空間 (x) を4分割し、4区画の中心点 (c_1, c_2, c_3, c_4) に対応する、機能設計空間 (y_1, y_2) 上の点 ($y_{c1}, y_{c2}, y_{c3}, y_{c4}$) を測定する。
- (2) これら ($y_{c1}, y_{c2}, y_{c3}, y_{c4}$) のうち、要求機能 (y_1^*, y_2^*) との距離が最も小さかった区画を選抜する。仮に区画 i が選抜されたとしよう。
- (3) 区画 i をさらに4分割し、4小区画の中心点 ($c_{i1}, c_{i2}, c_{i3}, c_{i4}$) に対応する機能設計空間上の ($y_{ci1}, y_{ci2}, y_{ci3}, y_{ci4}$) を見つける。
- (4) (2) と同様、要求機能 (y_1^*, y_2^*) との距離が最も小さかった小区画を選抜する。以下、(3) ~ (4) を繰り返す。
- (5) ある微小区画での機能の測定値 $y_{c..}$ が、目標値 y^* に十分に接近したと判断されたら、その区画の中心点を設計案として採用し、設計の収束・完了を宣言する。そこまでの所要時間が設計リードタイムとなる。

漸進モデルと並行モデルの得失： 一般に並行開発モデルは、設計案を複数用意して競争させ、敗退案は廃棄するので、前述の漸進設計モデルに比べ、資源効率が悪いという弱点を持つ。反面、当該人工物をめぐる技術や市場の不確実性が高く、既存設計案から出発し

でも最適解に到達できないリスクが大きい場合、大きく網をかける並行開発モデルのほうが、かえって安全な場合もある。このため、不確実性の高い製品開発状況、たとえば製品開発の初期において、並行開発法が用いられることが多い⁹。

逆に、漸進開発は、手堅く設計を収束させる方法として多用されており、設計コストも並行開発に比べて安価ですむ可能性が高いが、反面、既存設計から出発しても最適設計に到達しないような複雑なケースでは、うまく機能しない。このように、二つの方法は、状況により、補完的に使い分けるのがよいかもしれない。たとえば、開発の初期に並行開発（コンペティション）、後期に漸進開発（コーディネーション）と使い分けるケースが、医薬品開発などで見られる（桑嶋、2006）。

2-3 2段階設計モデルとしての設計プロセス

連立方程式と問題解決サイクルの結合：次に見るのは、「設計プロセスを連立方程式を解くプロセスに見立てる」という考え方に、「しかし設計者の因果知識は不完全である」という命題を加味した「2段階設計モデル」である（藤本、2005、2007；大隈・藤本、2006）。すなわち、設計の前半では、設計者は、不完全な因果知識（行列 A' ）をもとに、設計の暫定解（初期値）を求め、そこを出発点に、ノウハウ的な現場知識を用いた試行錯誤的問題解決により、究極の解（最適値）へ到達すると考える。そして、そこまでの所要時間が、設計のリードタイムである。つまり、不完全な因果知識にもとづく「連立方程式モデル」と、サイモンの「問題解決ヒューリスティック」を結合した、という意味で、「2段階」の設計モデルである。あくまでも現実を模写した抽象的な概念モデルに過ぎないが、一部の現場の設計リーダーに見せたところ、「意外にリアリティがあり違和感が少ない」との評価を得た。

そこで以下では、この「2段階設計モデル」で設計プロセスを捉えなおし、これにより、前述の「比較設計費説」（藤本、2007）と矛盾しない結果を導き出してみたい。

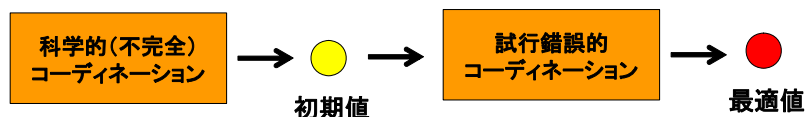
出発点は、例によって公理系設計論ベースの「連立方程式モデル」[$y = Ax$]である。しかし実際には、目標とする人工物の構造 x と機能 y の間の因果関係 (A) は、完全には分かっておらず、よって解析のみに頼って設計解を求めることは、通常は現実的でない。他方、設計者が、技術書や設計標準に首っ引きで事前の机上計算を行い、最終的な設計解になるべく早くに近づこうと努力するのをもまた、日常的な設計活動の一部である。

⁹ 例えば、マツダの初代ロードスターの開発において、FF（前輪駆動）型、MD（ミッドシップエンジン）型、FR（後輪駆動）型の設計が同時並行的に進められ、事後的にFRが採用された例がその典型例である（平井他、2003、pp.66～67）。当時、スポーツカー市場には高い不確実性が存在した。

そこで「2段階設計モデル」は、設計者は、①まず既存の因果知識（例えば公知の科学知識や既存製品の挙動に関する因果知識）を用いた机上計算や思考実験で可能な限り設計パラメータの予測を行い、その結果を「初期値」として設定し、②そこから試行錯誤的に、実物の試作・試験やコンピュータ・シミュレーションを用いて最終解へと接近し、満足すべき機能水準に達したら設計活動を完了する、と想定する¹⁰。つまり、実際の現場での設計活動は、最初は既存の因果知識の机上活用から始まるが、最終的には「限定的合理性下での満足化」（Simon, 1947）、あるいは、試行錯誤的な問題解決プロセス（Clark and Fujimoto, 1991; 藤本・安本, 2000）だと考える。要するに、設計者は、限定された合理性の制約下で、まず不完全な科学的知識によるラフな解析によって「初期値」を与え、そこから現場的知識に頼る試行錯誤によって「最適値」に到達するのである。

実際の設計の多くにおいても、真の因果構造 A のうち一部の変数 ($n' < n$) に関する情報しか分かっていないので、まず既知の不完全な科学的知識 A' を前提に、連立方程式を「科学的」に机上で解き、その暫定解 x' を「初期値」として、次に現場的知識を主に用いた m 回の試行錯誤によって、真の解 x^* に接近するのである（図4）。

図4 2段階の設計コーディネーション



科学的知識と現場的知識：以上でも明らかなように、設計解の求め方には、大きく分けて、(1) 事前の因果情報による解析的方法（連立方程式モデル）と、(2) 方向性を持った試行錯誤的方法（問題解決モデル）がある（図4）。

前者の解析的方法に使われる事前の因果知識は、科学知識、技術知識、伝承など様々なタイプの **know-why** 知識を含むが、ここでは、「科学的知識」でそうした普遍的因果知識を代表させることにしよう。

一方、試行錯誤的方法に使われる文脈特殊的な知識、例えば「どのあたりを設計変更すれば最適設計解に接近しやすいか」に関する、カン・コツ・経験に基づく **know-how** 的な知識を「現場的知識」と呼ぶことにしよう¹¹。

科学的知識による初期値の設定：以上の下準備を踏まえて、設計プロセスを「2段階モ

¹⁰ Pisano (1997) の learning by doing と learning before doing の概念が親和的である。

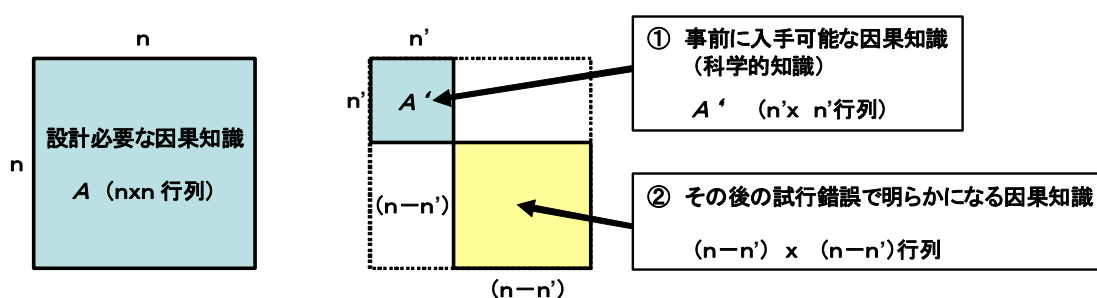
¹¹ こうした知識の類型は、青木 (1995) の「機能的技能」「文脈的技能」をはじめ、過去に類例がある。

デル」で素描してみよう。まず、設計者（研究者・技術者を含む）は、科学的知識を製品設計に活用しようと努力するが、その知識は、通常は不完全である。公理系設計論にしたがって、設計の完了に必要な因果知識の全体を、連立方程式 $[y=A x]$ における A とするならば、設計者が事前に入手できる科学的因果知識 A' は、これとは異なる ($A' \neq A$)。

設計者にとっての科学知識の不完全性は、少なくとも次の 2 つの方法で定式化できる。第 1 に、設計者が入手可能な科学的知識 A' は、設計者にとって不確実な存在であり、その要素である係数 a_{ij} はある確率的ばらつきを持った値をとる、とみなす方法がある。設計者の科学的知識が高ければ、 a_{ij} のばらつきは小さく、 A' と A の乖離も小さいと見る。後述のシミュレーション（大隈・藤本、2006）では、この定式化が採用される。

第 2 に、設計者が入手可能な因果知識 A' が、設計に必要な因果知識 A の一部分に過ぎない、という形での、より素朴な定式化が可能である。つまり設計者にとって事前に入手できた科学的な知識 A' は A の一部（単純化のため正方行列と仮定）であり、設計者は、この A' に基づいて、不完全な解を机上計算で得ることが可能とみる（**図 5**）。

図 5 「設計に必要な因果知識」と「事前に入手可能な因果知識」



注:ここでは、事前に入手する因果知識(A')とその後に試行錯誤で見出す因果知識は互いに独立でモジュラー的と仮定している。

科学知識へのアクセスが良いなどの理由で、他者よりも豊富な科学的知識を事前を持つ設計者は、そうでない設計者に比べて、最適設計解に比較的近いポイントに初期値を導出することが可能だろう。このとき、その設計者にとっての、行列 A' の次数 (n') は、 A の次数 (n) に比較的近く、またその設計者の初期値 y と最適値 y^* の設計空間上での距離は、より小さくて済むことが予想される。

こうしたラフな「科学的事前設計」の結果得られる「初期値」が、最適解にどれほど近づいているのかは、この段階において、重要な変数をどれだけ見逃しているかによって異なってくる。

このように、設計者の科学的知識のレベルは、「係数 a_{ij} に欠損値がある (穴が開いている)」

という定式化と、「係数 a_{ij} の真の値が設計者にとって未知であり、あるばらつきを持った値をとる」、という定式化とがありうる。研究目的に応じ、当面は前者を、後述のシミュレーション分析では後者を採用する。

現場的知識による最適値への漸近：「2 段階設計モデル」では、事前の科学的知識を用いた机上計算などにより、設計の初期値が決まれば、その先は、「現場的知識」を用いた試行錯誤、例えば試作品を使った実物実験、CAD（コンピュータ支援設計）に連動したコンピュータ・シミュレーション（CAE）などを用いて、最適の設計解に試行錯誤で最適解に漸近すると考える。とりわけ、実物実験の試行錯誤により、設計者は、真の因果関係 A からの間接的なフィードバックを得ることができる。すなわち、 A を直接観測することは出来ないが、任意の設計解 x に対して「真の因果関係」 A が作用した結果として実現される機能 y を実験等によって測定し、観察された y と機能目標 y^* の間の距離を小さくする方向に構造設計パラメータ群 x を変化させれば、真の因果関係 A を知らなくても、最適設計 x^* に収束しうる。

以上のような「2 段階設計プロセス」による定式化は、むしろ、現実の設計プロセスのごく粗い近似に過ぎないが、少なくとも、実際の設計活動の本質的な部分を捉えているのではないかと筆者は考える。

最適解への接近方法としては、前述のように、「漸進設計（コーディネーション）」と「並行設計（コンペティション）」、あるいはその混合型がありうる。

2 段階設計モデルのリードタイム分析：この 2 段階モデルに関しては、後にシミュレーション・モデルも試みるが、ここでは、より素朴なモデルを使った予備的な分析により、設計プロセスとリードタイムについて、大まかな見通しを示そう。その定式化の大筋は以下のとおりである。

(1) **事前の科学的知識**：設計者がある人工物の設計のために事前に利用可能な科学的知識（次数 n' の正方行列 A' ）は、その設計に必要な因果知識全体（次数 n の正方行列 A ）の一部分のみである（図 5）。

(2) **全体の設計リードタイム**：当該人工物の設計全体にかかるリードタイム（ T ）は、第 1 段階の初期値の導出リードタイム（ T_1 ）と、第 2 段階の初期値から最適値までの到達リードタイム（ T_2 ）の合計である¹²。

(3) **第 1 段階のリードタイム**：第 1 段階では、事前の科学的知識（ A' 次数 n' ）による部分的な連立方程式の解として、初期値（ x' ）が与えられるものとするが、そのリードタイム

¹² 基礎研究から製品開発までを考えると、第 1 段階が長い可能性もあるが、通常の製品開発では、第 2 段階がリードタイムでも工数でも大きなウェイトを占める可能性が高いと予想される。

が「科学的に既知の設計要素」 n' のサイズによって伸縮するという「連立方程式の解」としての定式化は、実際の設計開発活動の描写としてやや理解しにくい。むしろ、初期の科学的設計の期間 (T_1) は、たとえば基礎研究に関する業界内の慣行により一定と考え、その制限時間 (T_1) の中で設計に活用できた科学的知識を A' と考える。

(4) 科学的設計の組織能力：設計者は、科学的設計に振り向けられた制限時間 (T_1) を活用して、できるだけ設計の初期値 (x') を最適設計解 (x^*) に近づけようとする。ある現場 (i) における「科学的知識の活用のうまさ」を s_i とすれば、 $[n' = s_i \cdot T_1]$ と近似することもできよう。この s_i は、科学的設計に関する現場の組織能力とみなせる。

(5) 第2段階のリードタイム：第2段階における試行錯誤的問題解決のリードタイム (T_{2i}) は、科学的設計によって到達した初期値 (x') から最適値 (x^*) までに必要な試行数 (m) と1試行あたりリードタイム (t_j) (ただし j は現場・国など) の積 ($T_{2i} = m \cdot t_j$) で表せる。

(6) 試行錯誤設計の組織能力：国や現場によって、組織的問題解決の組織能力が異なり、よって第2段階のリードタイム (T_{2j}) は現場や地域 (j) によって異なる値をとりうると考える。1試行あたりリードタイム (t_j) は、統合型組織能力が浸透し、事前の情報共有が進んだ現場や国と、そうでないところで、相当に異なるものと想定する。

(7) 初期値の乖離と設計試行数：問題解決の設計試行数 (m) は、設計空間における初期値 (x') と最適値 (x^*) の「距離」に影響される。そしてこの「距離」は、設計者が事前に検討できた科学知識の利用レベル (A' の次数 n' の大きさを代理) が高いほど小さくなる。逆に言えば、事前に科学的に処理できなかった課題数、つまり、必要知識 (n) と科学的知識 (n') の乖離である $n - n'$ が大きいほど、最適解 x^* から遠い初期値 x' から出発せねばならず、それだけ設計試行数 (m) は増える。

(8) 設計収束までの試行数：必要試行数 (m) は、アーキテクチャや収束条件などの諸要因も絡むので一義的には決まらないが、設計要素間の相互作用が密に存在する「インテグラル型」製品の場合、 $[n - n']$ 人の構造設計者が、 $[n - n']$ 人の機能設計者と r 回ずつ局所的な設計調整することで全体設計が収束するとするなら、 $m = r [n - n']^2$ 、つまり必要試行数は $[n - n']$ の二乗に比例すること考えてよかろう。他方、構造設計要素と機能設計要素が一対一対応する「モジュラー型」製品の場合は、すでに示したように、 $m = r [n - n']$ 。つまり必要試行数は、インテグラル型 b の場合よりずっと少なくなろう。

(9) 科学的設計と試行錯誤設計の分離：第1段階の科学的設計によって得られる n' 個の初期設計パラメータ (A' に対応) と、第2段階の試行錯誤設計で解かれる $n - n'$ 個の追加設計パラメータは、互いに独立と仮定する (図 5)。これにより、第2段階の試行錯誤活動によって第1段階のリードタイムが影響を受けないこととなり、全体のリードタイム T は、

単純に第 1 段階の所与のリードタイム (T_1) と、第 2 段階のリードタイム ($T_{2j} = m \cdot t_j$) の単純総和と考えればよくなる。もっともこれは、議論を単純化させるための方便ではあり、後のシミュレーション分析ではこの仮定は取り払われる。

(10) 2 段階設計のリードタイム・モデル : 以上の仮定の下で、「2 段階設計プロセス」において、現場 i が最適設計解にいたる設計リードタイム T_j は、

$$\begin{aligned} T_j &= T_1 + T_{2j} = T_1 + m \cdot t_j = T_1 + r (n - n')^2 t_j \\ &= T_1 + r (n - s_j T_1)^2 t_j \end{aligned}$$

となる。したがって、製品の複雑さ (n, r) や企業の基礎研究ポリシー (T_1) を所与とする、科学設計の組織能力 (s_j) と、試行錯誤設計の組織能力 (それを反映した調整スピード : t_j) が、設計リードタイムに影響する 2 つの潜在的要因ということになる。特に後者 (t_j) は、インテグラル・アーキテクチャと適合的な設計プロセスと整合的であり、「統合型」の設計能力と言える。

J 国現場と U 国現場のリードタイム比較 : ここで、この 2 つの組織能力のプロフィールが異なる J 国と U 国を想定し、「2 段階モデル」によるリードタイム比較を考えてみよう。

まず、日本の設計現場の観察から抽出された「定型的事実」から出発しよう。すなわち、J 国の設計現場、U 国の設計現場には、歴史的な発展経路等の違いから、以下のように異なるプロフィールの組織能力が多く存在するでしょう。

J 国現場の設計能力 [統合型] : 「J 国現場」は、戦後日本の貿易財企業の現場を想定した理念型である。戦後の高度成長期に発達した長期雇用・長期取引慣行の結果、現場が現場特殊なものづくり知識を蓄積し、長期取引・長期雇用ネットワークを通じて、それを組織内・組織間で共有している。また、多能的な設計者・作業者が多く、彼らは相手の仕事の方向性を予想しつつ行動できる (Clark and Fujimoto, 1991; 浅沼, 1997; 藤本・西口・伊藤, 1997; 小池, 1999; Nishiguchi, 1994; 武石, 2003 他)。大部屋で隣接部品の設計者の活動を観察しつつ自分の設計を行うので、同時に複数の設計者が設計パラメータを動かしても、全体の整合性を保つことが比較的容易である。よって、問題解決のための設計調整のスピードが速い。一方、長期取引により依存する分、科学的知識の共有は、長期取引を行なう現場組織間に限定される傾向がある。つまり、現場的知識は「T 字型」、科学的知識は「狭い一の字型」となりやすい。

U国現場の設計能力〔分業型〕：「U国現場」は、歴史的にはいわば移民の国である米国の企業を念頭に置いた理念型であり、現場組織は移民を即戦力とするために分業と標準化を重視し、企業内の情報共有は専門家間の狭い範囲に留まる。その反面、U国は市場機構やオープン・ネットワーク（國領、1995）が発達しているため、広い範囲で科学的知識（汎用知識）を共有することに長けている。つまり、U国現場の方が、J国より、オープンな科学者ネットワークに依存した科学的知識の幅が広く、それによって、科学的知識を応用できる設計要素の数（ n' ）の値が大きい。かくして、現場的知識は「I字型」、科学的知識は「広一の字型」である。

要するに、U国の設計現場は、J国の設計現場より、第一段階における「科学的設計」に関する組織能力が高く、その結果、U国現場のほうがJ国現場より、科学的知識を設計に反映させるスピードが速い。すなわち：

$$s_J < s_U$$

一方、J国の設計現場は、U国の設計現場より、第二段階における「試行錯誤設計」の組織能力が高く、その結果、1回あたり平均の組織的試行錯誤のスピードが速い。すなわち：

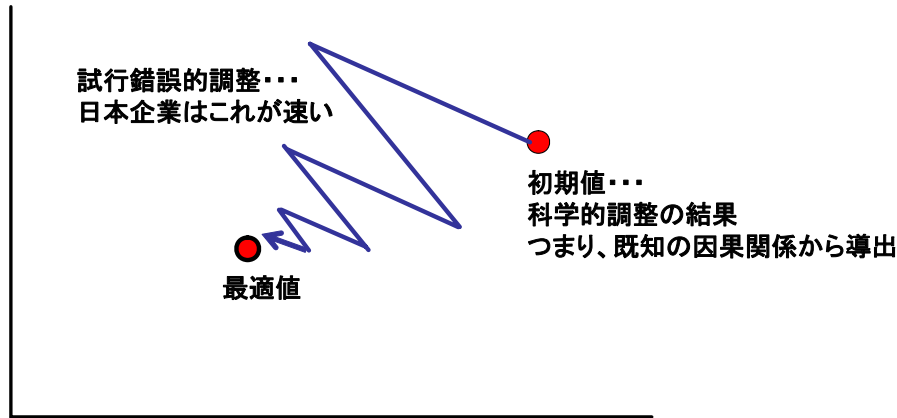
$$t_J < t_U$$

この場合、U国の設計リードタイム [$T_U = T_1 + r(n - s_U \cdot T_1) \cdot t_U$]と、J国の設計リードタイム [$T_J = T_1 + r(n - s_J \cdot T_1) \cdot t_J$]のどちらが短いかは、ケースバイケースで、一概には言えない。

そこで、複雑な製品（ n が十分に大きい製品）に絞り、「通常のインテグラル型」「科学知識集約的なインテグラル型」「モジュラー型」という3つのタイプの製品について、U国現場とJ国現場のリードタイムをざっと比較してみよう。

①通常のインテグラル型製品—J国現場が優位：とくに、製品は複雑（ n や r が比較的大）だが、科学的知識をあまり使わない（ $n' = s_J \cdot T_1$ が比較的小）タイプの製品の場合では、「統合型の設計能力」（ t_j ）におけるJ国現場のリードタイム優位が増幅して現れやすい（**図6**）。通常型の小型ガソリン自動車などは、従来、このタイプであったと考えられる。

図6 「試行錯誤的調整」における日本企業の優位性



②科学知識集約インテグラル型製品—U 国逆転の可能性：しかしながら、科学的知識が設計にとって重要 ($n' = s_j \cdot T_1$ が比較的大) であるような「科学知識集約製品」の場合は、様相が異なる。すなわち、「統合型の設計能力」(t_j) が高いからといって、設計リードタイム上の競争優位は保証されない。

むしろ、科学知識が世界中で等しく利用可能であり、競合する設計現場間で科学設計の組織能力 (s_j) に差が出ないのであれば、出発点の「初期値」で U 国現場が特に有利という展開にはならず、結局、前述の「通常のインテグラル型」の場合と同様、現場間の設計リードタイム (T_j) の差は、単純にこの「統合型の設計能力」(t_j) の現場間の差で決まってくると予想される。この場合は、統合型組織能力を持つ J 国現場が、依然有利である。

しかし、科学知識が世界中で入手可能だとしても、それを設計に活用する能力は、現場・企業・国ごとにばらつく可能性がある。その場合、事前の科学的設計に関する組織能力 (s_j) が概して高い U 国の設計現場は、仮に試行錯誤のスピード (t_j) で J 国現場に負けていたとしても、事前の科学知識の利用 (n' の優位) で他に差をつけ、初期値 (x') を最適値 (x^*) の十分近くにもっていくことで J 国を逆転し、設計リードタイム上の競争優位を築けるかもしれない。これが、後にシミュレーションで検討する「ウサギとカメ」状況である。

③モジュラー型製品—J 国は勝ちにくい：最後に、モジュラー型製品の場合は、これまで繰り返し見てきたように、J 国現場は勝ちにくい。この場合、モジュラー型の場合、すでに述べたように、機能要素 (y_i) と構造要素 (x_i) が 1 対 1 対応であるため、設計リードタイムは、慎重に逐次的に設計調整したとしても $[T_j = T_1 + r(n-n') t_j]$ 、同時並行的に設計調整をすれば $[T_j = T_1 + r t_j]$ で済んでしまう。いずれにしても、第 2 期の試行錯誤的設計の期間 (T_{2j}) は、インテグラル製品に比べてずっと短いので、仮に J 国現場の調整スピ

ードが速い(t_j が小さい)としても、それは設計リードタイム全体には大きくは反映しない。つまり、J国現場にとっては、U国に対し「良くて引き分け」に近い状況である。

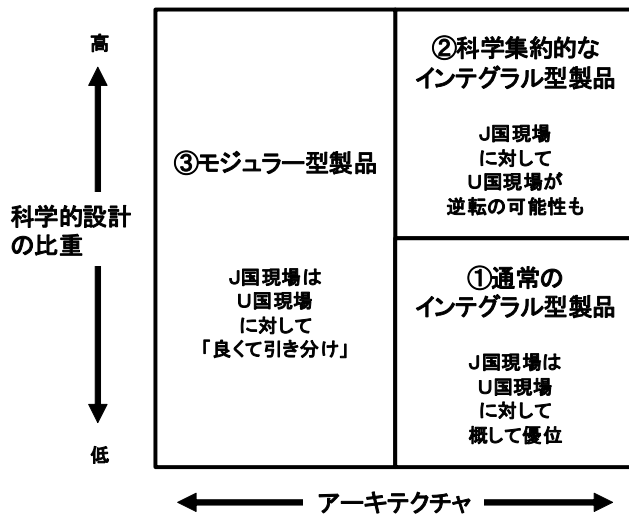
「通常のカモジュラー製品」場合、前述のように、U国現場の科学知識利用の優位性(s_j)は生かないので、U国現場とJ国現場は、いわば引き分けに近い。一方、「科学知識集約的なカモジュラー製品」の場合、U国現場が科学的設計の組織能力を生かして初期値の布石で優位なポイントをとった場合、J国現場に逆転策は残されていない。つまり「良くて引き分け」であり、比較優位論においては、これは[比較劣位]を意味する可能性が高い。

リードタイム分析のまとめ：ここまで、設計プロセスと設計リードタイムの関係について考察してきた。すなわち、開発の第1段階を「科学的設計」、第2段階を「試行錯誤的設計」と見る「2段階設計モデル」を適用して、「統合型ものづくり能力」に富むJ国の設計現場と、事前の科学知識利用の能力に富むU国の設計現場の競争力比較を、リードタイムの観点からみてきた。その予想をまとめるならば、**図7**の通りである。

この図の横軸は、従来の「インテグラル型⇔カモジュラー型」というアーキテクチャ・スペクトルであるが、縦軸には、第1段階の「科学的設計」が競争力全体に与える影響度を取り、その影響が大きい「科学知識集約型製品」と、大きくない「通常の製品」を区別している。

モデルから導かれる設計リードタイム比較によるなら、まず、カモジュラー製品で統合型のJ国現場は競争優位を持ちにくいという結果を得た。しかし、J国現場が設計の比較優位を持ちやすいとここまで主張してきた「インテグラル型」の製品に関しては、確かに「通常のインテグラル製品」ではJ国現場の調整能力が生かされて、リードタイムで優位性を持ちそうだが、「科学知識集約的なインテグラル製品」の場合は、科学知識利用能力の高さゆえに「初期値」の位置取りで優位性を持つU国現場に、リードタイムで逆転される可能性が予想された。J国現場は、インテグラル型製品全般で強いとは限らず、仮に、科学的知識の活用能力において弱点を持つ場合、科学知識集約的なインテグラル製品でもリードタイムで劣位に陥る可能性があることが示唆された。

図7 製品のタイプとリードタイム優位の予想



開発リードタイムと開発工数：本稿ではここまで、設計プロセス論に戻って、組織能力の異なる設計現場が、異なる製品でどのようなリードタイム競争力を生み出しうるかを、簡単なモデルを用いて予想した。ここでリカード的な発想の「比較設計費説」に、以上の分析結果を結合しておこう。

製品開発における生産性（開発工数）とリードタイムの関係については、さまざまな議論があるが、数理経済学のモデルでは、開発コスト（開発工数 x 時間当たり開発単価）と開発リードタイムの間には負の関係、つまり開発リードタイムを短縮化すれば開発コストは上がる、というトレードオフ関係が想定される傾向があった（伊藤・奥野・清野・鈴木、1988、第18章、19章）。一方、実証分析においては、むしろ開発工数・開発コストと開発リードタイムの間には、正の相関関係、つまり開発リードタイムが短縮化されれば開発工数も低減されるという事例が多く報告された（Clark, Chew and Fujimoto, 1987; Clark and Fujimoto, 1991 他）。

この乖離の原因は、以下のように推定される。すなわち、理論においては、各企業の組織能力を所与としたトレードオフ関係が重視される。具体的には、リードタイム短縮には余計な追加人員投入や開発単価の上昇があるので、開発コスト高につながると見る。

一方、実証研究においては、測定対象の現場群に、通常は組織能力のばらつきがある。ゆえに、通常はそれを反映して、組織的問題解決能力が高い開発現場はリードタイムが短く生産性も高い（逆は逆）、という論理で、正の関係を説明する（Clark and Fujimoto, 1991; 藤本, 2001b）。

本稿では、組織能力の非同質性を前提にした進化論的モデルを採用しており、また実証分析から抽出された「定型的事実」を重視する立場からも、後者の考え方、つまり開発リードタイムは開発工数、開発コストと正の相関を持つと考える。これを反映したモデルは、ごく単純に、「プロジェクト投入人数 (α_j) × リードタイム (T_j) = プロジェクトあたり開発工数 (H_j)」となる (j は設計現場)。実際の関係はもっと複雑だが、ここでは、プロジェクト投入人数は不変で、全員が当該プロジェクトにフル参加すると仮定する。よって、リードタイムが伸縮すれば、開発工数もリニアに伸縮する ($H_j = \alpha_j \cdot T_j$)。

比較設計費説への含意：以上のように、第 j 設計現場の開発工数 (H_j) と開発リードタイム (T_j) の関係を単純化するなら、本稿の設計プロセス分析を、リードタイムを介して、すでに論じた「比較設計費説」(設計の比較優位説) に直結させることが出来る。すなわち、財の平均コストを AC とし、それを生産コスト (PC) と設計コスト (DC) の和と考え、新製品の設計開発の局面では、企業はとりあえず後日の生産コストは所与(仮置き)として、設計コストの比較により設計立地を決めるものとしよう。すなわち、製品設計段階で考慮されるコスト推定は以下の通りである。

$$AC_j = PC_0 + DC_j \quad (PC_0 \text{ は所与の推定値。以下では定数項とみなす})$$

ここで設計コスト (DC_j) に着目するなら、開発の時間単価を W_j 、開発(設計)リードタイムを T_j 、開発工数を H_j とすれば以下のようなになる。

$$DC_j = W_j \cdot H_j = W_j \cdot \alpha_j \cdot T_j$$

ここに、2段階設計モデルのリードタイム推定式を入れれば、インテグラル製品の場合は、前項の分析を援用すれば

$$DC_j = W_j \cdot \alpha_j \{T_1 + r(n - s_j \cdot T_1)^2 t_j\}$$

モジュラー製品の場合は、順番に設計調整する場合は、

$$DC_j = W_j \cdot \alpha_j \{T_1 + r(n - s_j \cdot T_1) t_j\}$$

同時に設計調整する場合は

$$DC_j = W_j \cdot \alpha_j \{T_1 + r \cdot t_j\}$$

したがって、もしも科学的知識の吸収力 (s_j) の現場間の差がでない場合は、 t_j で優位性を持つ J 国の現場が、そうでない U 国現場に比べて、インテグラル製品で常に大きなリードタイム優位(したがって開発工数優位)を持つと推定される。

$$T_j \ll T_U, \quad H_j = \alpha_j \cdot T_j, \quad \text{ゆえに、} H_j \ll H_U,$$

これに比べ、モジュラー型製品のリードタイム優位は相対的に小さいか微小なので、

$$T_j \leq T_U, \quad H_j = \alpha_j \cdot T_j, \quad \text{ゆえに、} H_j \leq H_U$$

仮にこの 2 財（通常のインテグラル財とモジュラー財）しかなく、国境を越えた設計者等の移動に制約があるなら、リカードのロジックに従い、為替や設計者の実質賃金等 (W_j) が調整される結果、設計現場の立地は、インテグラル財では $[DC_J < DC_U]$ で J 国設計現場が優位、モジュラー財では $[DC_J > DC_U]$ で U 国設計現場が優位となりやすい。つまり、前述の比較設計費分析と同様の結果を得る¹³。かくして、比較設計費説と、本稿の設計プロセス分析が結びついた。

科学知識の役割と「ウサギとカメ」現象：しかしながら、「2 段階設計モデル」によれば、

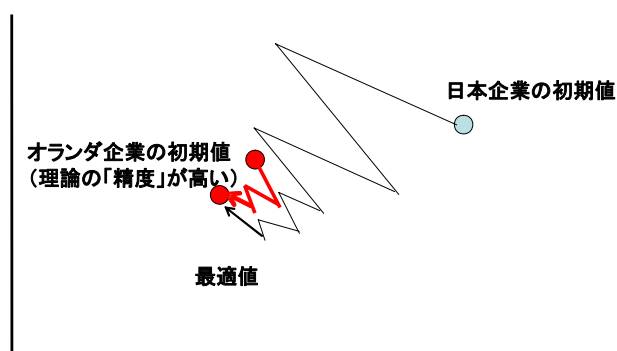
(i) 設計活動に多くの新しい科学知識を必要とするような科学知識集約製品の場合で、(ii) なおかつ、U 国設計現場が、科学知識を設計に活用する吸収能力で J 国現場を上回っている場合 ($s_J < s_U$)、U 国現場は、科学的設計力を生かして、第二段階のスタート地点（初期値）を、J 国現場よりゴール（最適値）に近いところに置く。

確かに、そこから先の問題解決スピードでは J 国現場が勝るのだが ($t_J < t_U$)、設計の第 1 段階における「初期値の置き方」の巧拙でライバル（U 国現場）に差をつけられ、スタート地点がゴールから遠いので（**図 8**）、この距離の差を挽回できないかもしれない。いわば J 国現場が「ウサギとカメ」のウサギ（速いがゴールから離れ過ぎ）の立場になり、開発リードタイム・レースに「利口なカメ」（遅いがゴールに十分接近）である U 国現場に負ける、というシナリオである。

そしてこの状況は、モジュラー型製品でも、インテグラル製品でも、 t_J 、 t_U 、 s_J 、 s_U の値のとり方次第では、生じうる。

¹³ ここではリカードの古典的な議論に合わせるため、組織能力の差がもたらす設計プロセスの違いが設計コストに反映すると考えたが、この部分を、製品開発競争でよく見られる「一番乗りゲーム」、すなわち技術開発・製品開発のリードタイムが短く、結果として特許申請や市場導入で一番乗りを果たした企業が売上げ・利益を総取りするモデル（伊藤・奥野・清野・鈴木、1988 他）に置き換えることもできる。その場合、組織能力の違いが設計プロセスを経由してもたらす競争優位は、より劇的なものとなる。したがって、上記のモデルが予想する結果が逆転することはない。

図8 「科学的調整」におけるオランダ企業の優位性（「ウサギとカメ」現象）



- (1) 製品の「擦り合わせ度」が極端に高い(複雑な連立方程式)
- (2) 日本企業は事前の科学的知識が低く、事後的な試行錯誤に頼る
- (3) オランダ企業は、事前に把握している変数や因果式が多い 科学的調整力

2 段階設計モデルが導き出すこの「ウサギとカメ」状況は、現実にもすでに該当例がみられる。たとえば、きわめて複雑なインテグラル・アーキテクチャ製品として知られる半導体露光装置において、従来セオリー通りに競争優位を持っていた統合型の日本企業（ニコン、キャノンなど）に対して、欧州企業（ASML 社）が競争優位性を高めた、という現象が知られており、これは、単純な「J 国＝インテグラル優位」説では説明できない（中馬, 2004 他）¹⁴。

「インテグラル型製品でも日本（J 国）の現場が強いとは限らない」というこの現象は、この他にも、宇宙航空産業や電子・精密機器産業等における、多様な先端科学知識を要する複雑な製品において少なからず観察され始めている。その理由としては、様々な代替仮説があり得るが、「2 段階設計モデル」を用いた解釈も可能である¹⁵。すなわち、半導体露光装置のように、極めて設計要素（ n ）の大きい、複雑な擦り合わせ製品を想定しよう。その開発の第 1 段階で、統合型の J 国現場は、狭い長期関係ネットワークをベースに科学的知識の探索を行なっているのに対し、U 国設計現場（ここではオランダ ASML 社）は、半導体製造装置企業、半導体材料企業、半導体企業、大学など、幅広い科学的知識の共有が

¹⁴ 半導体露光装置の分野でオランダの ASML 社が近年台頭してきたことは、良く知られた事実である。従来は「これはフィリップス社の電子系とツァイス社の光学系をたくみに組み合わせる ASML 社製品のモジュラー性に起因する」と説明されることが多かったが、中馬（2004）は、ASML 社製品を含めて近年の半導体露光装置は極限的な擦り合わせアーキテクチャ製品であり、したがって説明すべきは、なぜ「超擦り合わせ製品」であるはずの半導体露光装置において日本企業が劣勢になるのかであると主張する。ASML 社の半導体露光装置が実際にモジュラー的なのか超インテグラル的なのかについては、なお議論の余地があるが、ここでは中馬説の事実認識をベースに議論を進めることにする。

¹⁵ この仮定は、オランダ企業は日本企業より幅広い参加者による科学的知識探求のネットワークを持っている、との中馬（2004 他）観察と整合的である。

身上である。したがって、U 現場（企業）の科学的知識の範囲（ n'_U ）は、J 現場（企業）の科学知識の範囲（ n'_J ）より顕著に大きい可能性が高い（ $n'_U \gg n'_J$ ）。つまり、U 現場の科学的設計力の差が大きい場合（ $s_J \ll s_U$ ）、第 1 段階で U 現場が J 現場に大きな差をつける [$(n - n'_J) \gg (n - n'_U)$]。この場合、J 企業は、現場の調整能力の高さ（ $t_J < t_U$ ）にもかかわらず、初期値の不利を最後まで挽回できない。

つまり、擦り合わせの「2 段階モデル」において、科学的な設計パラメータ探索量における企業差が、その後の試行錯誤によるパラメータ探索の企業間スピード差を上回る場合、インテグラル型製品でも U 企業が J 企業より競争優位に立つ可能性が出てくる。それも、 n が非常に大きい「極端に複雑なインテグラル型製品」でこそ、こうした逆転劇は起こりやすい、というパラドックスが、ここに存在するのである。

3 シミュレーション分析：「ウサギとカメ」仮説の確認

3-1 シミュレーション・モデルの概要

シミュレーションのねらい：ここまで「2 段階設計モデル」をベースに、設計者の合理的な設計行為を考慮にいたした設計プロセス論の観点から、「設計の比較優位説」を再検討した。簡単なモデルを準備し、諸所に単純化のための仮定を置き、設計プロセスと開発リードタイムの関係について、一連の予想を行った。その結果は、「設計の比較優位説」の予想とおおむね整合的であった。また、設計を「初期値への到達」「最適値への到達」の 2 段階に分けることにより、「ウサギとカメ状況」に関する新たな仮説も得た。

しかし、このような単純なモデルでも、リードタイム競争力の優劣を分ける条件をすっきりと示すことは出来ていない。ある程度、現実の設計行為に近いモデルを考れば、かなり簡単なモデルでも、その結果を予想するのは容易でない。

そこで大隈・藤本（2006 他）は、「2 段階設計モデル」を出来るだけ忠実に再現したシミュレーション・モデルを作成し、これによりリードタイムの動きを分析した。以下、その結果の概要を紹介する（詳しくは、大隈・藤本（2006）を見よ。細かいパラメータの設定は大隈が行った）。

変数の設定：まず、シミュレーション・モデルにおける、各変数の考え方を示そう。考慮するのは、前述の「2 段階設計モデル」

$$[T_J = T_1 + r(n - n')^2 t_J = T_1 + r(n - s_J \cdot T_1)^2 t_J]$$

を構成する諸変数が中心である。すなわち、

- ① 製品の複雑さ：設計要素数 n 、調整繰り返し数 r に反映。
- ② 製品アーキテクチャ：インテグラル型・モジュラー型の 2 類型。

- ③ 科学知識集約度と科学的設計能力：科学的知識 A' 、あるいはそれに関連する設計要素数 n' 、科学知識吸収力 s_j などに反映。
- ④ 最適化の設計プロセス：最適解への接近法としての漸近設計（コーディネーション）と並行設計（コンペティション）の2類型。
- ⑤ 試行錯誤的デザインの組織能力：組織が最適解への試行錯誤的な接近を行なう際の、1回あたりの調整時間（試行スピード） t_j に反映する。
- ⑥ 市場の洗練度（収束条件）：市場がどの程度洗練された製品を要求するか、つまりこの製品に関する顧客の鑑識眼がどのくらい厳しいかを、「市場許容品質」で示す。すなわち、最適解にどこまで接近したときに顧客がその製品に「合格」の判定を下すかで、市場ニーズの洗練度を表現する。

①設計要素数 (n)：所与の製品の複雑さを示す指標には、設計要素の数 (n) と、設計要素間の相互依存関係を示すアーキテクチャの2つがある。

すでに見てきたように、構造設計要素の数 (n) は、公理系設計論においては、機能設計要素の数に等しく、それは要するに、因果行列 A の次数 (n) である。人工物は階層的であるから、構造要素の数は、どこまで階層分解するかによって異なるが、前述のように、まず、顧客にとって価値のある（許容価格に影響する）機能要素の数をまず n とすれば、公理系設計論では、それが構造要素の数となる。つまり、ここで測定しているのは、実はその人工物の利用者（顧客）からみたその製品の複合度だといえよう。いずれにせよ、この数が多くなるほど複雑度は高いと仮定するのは自然であろう。

②製品アーキテクチャ（インテグラルかモジュラーか）：繰り返し述べたように、同じ設計要素数であっても、アーキテクチャがインテグラル型の製品の場合、機能・構造設計要素間の相互依存性が高いので、要素数で同サイズのモジュラー型製品よりも複雑度が高いといえる。言い換えれば A の要素 a_{ij} のうち、ゼロでないものの数を勘定すれば、それがアーキテクチャのインテグラル度を示すといえる。

③科学的知識 (A' のばらつき)：前述の「2段階設計モデル」では、設計に必要な因果知識 A のうち、基礎研究の限られた時間内で設計に反映させられる科学的知識 A' は A の一部に過ぎない、という設定で、この変数を指標化した。しかし、シミュレーション・モデルでは、この形は扱いにくく、また仮定が恣意的な面も否めない。

そこで今回は、すでに示したもうひとつの定式化を採用した。すなわち、設計者が第1段階で、設計の初期値を求めるために用いる事前の科学知識 A' は、真の因果関係（完全知識）を示す行列 A と同じサイズの $n \times n$ 次の正方行列だが、その要素 a_{ij}' は、対応する完全知

識 a_{ij} からずれており、つまり不完全知識だと考える。具体的には、この乖離を確率的なばらつきと考え A' の要素である個々の科学的知識 a_{ij}' は、真の因果関係 a_{ij} から、ある分散 (σ^2) をもったばらつきで乖離していると考え。この分散が小さいほど、設計者が設計の際に事前に持つ因果知識（ここでは科学的知識と想定する）のレベルは高い。

一般に、先端科学知識を多く必要とする「科学知識集約製品」は、発展途上の知識を使うので、科学的因果知識のレベルは低い傾向があろう。また、そうした先端的な科学知識集約製品の場合、その知識を豊富に持つ（上記の分散 σ^2 が小さい）設計者と、そうでない（分散 σ^2 が大きい）設計者の間で、設計パラメータの初期値（暫定解）の出来に大きな差が出るのが予想できる。

④設計プロセス（コーディネーションかコンペティションか）：前述の通り、最適解に接近する設計プロセスには、「漸進設計（コーディネーション）型」と「並行開発（コンペティション開発）型」がある。この 2 タイプのプロセスに要する組織能力の属性は異なる。すなわち、漸進設計型では、各設計要素を担当する専門担当者が、自らの加えた設計変更に関する情報を次の担当者に迅速に伝達したり、試行結果に関する情報を全ての担当者が共有したりする必要がある等、必然的に担当者間での密なコミュニケーションが必要となる。一方、並行設計型では、公平な裁定者がいれば、異なる設計案で競っている各担当者間に必ずしも緊密なコミュニケーションは必要でない。つまり、統合型の組織能力の有無が大きく影響するのは、コーディネーション型の方であると予想される。

⑤組織能力（調整 1 回あたり所要時間 t_j ）：試行錯誤的な設計に関する組織能力、すなわち設計調整 1 回あたり所要時間の長さ (t_j) は、設計担当者間の情報共有の度合いやチームワーク力によって影響を受けると考える。例えば、個室にいる技術者間で 1 対 1 の調整を繰り返すのと、大部屋で互いの動きを見ながら 3 人以上が同時に調整するのでは、同じ製品であっても、調整スピードが大きく異なる可能性がある。これは J 国（日本など）の設計現場に偏在する統合型の組織能力を代理する変数とみなせよう。

⑥市場の洗練度（収束条件）：当該製品の機能に対する市場要求の厳しさを表す指標として、このシミュレーションでは、「製品機能に関する収束条件」を考えた¹⁶。すなわち、初期値から最適値へ向かう第二段階の設計プロセスにおいて、いったいどこまで、現行設計案が発揮する機能 (y') が客の要求する機能目標 (y^*) に接近したら、設計プロセスの終了を宣言するのか。つまり、市場が許容する機能目標からの乖離（設計空間上での距離）が、ある一定の値以下に達した時点で、市場の要求する機能を満たしたものと判定して設計を

終了すると考える。この収束判定基準の値が小さければ、市場が要求する機能要求は厳しく、逆に大きければ緩いと解釈できる。

以上で「2 段階設計プロセス」のシミュレーションのための定式化は一通りできたので、次に、これらの変数とリードタイムの関係に関し、シミュレーションの基本的な結果を示そう。

3-2 シミュレーション結果の素描—諸要因のリードタイムへの影響

まず、シミュレーション結果として算出された平均リードタイムに対して、製品の複雑さ、科学的知識の不完全性、設計プロセスのタイプ、市場要求の厳しさなどが、どのように影響しているかを、一通り確認しておこう（詳しくは大隈・藤本、2006 を見よ）。

製品の複雑さとリードタイム：常識的な予想通り、製品の構造設計要素数（行列 A の次数 n ）が多いほど、平均リードタイムが長いことが示された。

また、これも予想通り、構造要素数（ n ）が同じ人工物であっても、モジュラー型製品よりインテグラル製品の方がリードタイムが長くなり、しかも構造要素数が大きくなると、その差が加速度的に大きくなる傾向が示された。これは、すでに示したモデルの特徴を反映している。すなわち、設計リードタイムは、モジュラー製品の場合は要素数（ n ）にほぼ比例して、またインテグラル製品の場合は要素数の二乗に比例して増加する傾向が見られた。

科学知識集約度とリードタイム：因果知識行列 A' の要素のばらつきで科学的因果知識の不完全性を示すとすれば、因果知識が不完全なほど（科学知識集約的であるほど）、平均リードタイムは長くなる傾向がみられた。とくに、インテグラル製品の場合、因果知識の不完全性が高まるにしたがって、リードタイムは急速に長くなる傾向があった。つまり、因果知識が不完全であることがリードタイムに与える影響は、設計要素数が多い「複雑なインテグラル製品」で特に大きい。これは、科学知識に弱く試行錯誤に強い「J国現場」にとっての鬼門は、複雑でインテグラルな科学知識集約型製品である、という前述の予想（藤本, 2005）と、このシミュレーション結果は整合的にみえる。

設計プロセス（漸進設計 vs 並行設計）とリードタイム：漸進設計（コーディネーション型）は、結果的に設計開発が成功する（設計解が最適解に十分近づく）場合には、並行設計（コンペティション型）よりも短いリードタイムで最適解に収束する傾向が見られた。

また、並行設計（コンペティション）型の場合、不確実性の高い開発の場合でも、より高い確率で設計解が最適解に収束するが、要求機能に対する実現機能の乖離は、漸進設計

¹⁶ 大隈・藤本（2006）では、この基準を「市場要求品質」と呼んだ。

型の成功製品に比べると大きく、要するに出来は劣る傾向がある。他方、漸進設計（コーディネーション）型の場合、結果的に設計開発に成功している限りは、並行型に比べてリードタイムは短く、製品の出来も良い（実現機能が要求機能に近い）のだが、そもそも設計解が最適解に向かって収束せず、失敗してしまうリスクが小さくない。したがって、不確実性の高い荒れた状況では、並行開発型の方が、効率は悪いが手堅い。これは、現実の製品開発でも見られる、並行開発を不確実性の高い開発初期の段階のみで使う例とも整合的である¹⁷。

市場ニーズの厳しさ：製品に対する市場ニーズの厳しさは、理想の要求機能水準にどこまで近づいたら「収束＝合格」とみなすかを示す「市場が許容する機能乖離」で代理したが、シミュレーションの結果は、予想通り、市場ニーズの条件が厳しい方が平均リードタイムが長くなる傾向を示した。

シミュレーションの意味：以上のように、「2段階設計プロセス」（藤本、2005）を模したシミュレーション・モデルは、基本的には、予想された一連の挙動を示し、ある程度のリアリティを示した。このシミュレーション・モデルは、製品の複雑さ、製品アーキテクチャ、科学的知識の集約度、設計プロセス（並行開発か連携調整か）、問題解決の組織能力（設計調整の試行スピード）、市場要求の厳しさ、などを明示的に取り込んでおり、その意味で、従来に比べ、よりリアルなモデルになっていると考える。シミュレーション手法に関しては、パラメータを調整すれば何でもいえる、との批判も存在するが、同じ構造のモデルで、リアリティのある複数の挙動を同時に示したということは、アドホックなモデルという以上の結果ではないかと考える。

3-3 J国とU国のリードタイム競争—シミュレーション結果の素描

「浅薄なウサギ」対「周到なカメ」の対戦：最後に、このモデルを使って、前述の「J国現場」と「U国現場」の間の「ウサギとカメ」状況の再現を試みた。その結果の概略示す。

このシミュレーションで確認したいのは、前述の「現場は強いがサイエンスに弱いJ国の設計現場」、すなわち「統合型ものづくり」の組織能力が高く、現場での試行錯誤的調整のスピードが速い反面、現場の調整努力に依存しすぎ、事前に体系的に科学知識を得るといふことにあまり熱心でない、という「理念型としての日本の設計現場」の挙動である。逆

¹⁷ 例えば、Iansiti (1998, pp. 130–131)、平井他 (2003, pp. 66–67) など参照。また、藤本・安本 (2000, 12章) では、「複数の異なる要素技術を試作品で比較検討」（競争型開発・並行開発）という項目を含む多変数のアンケート統計分析を多産業のプロジェクトを対象に行なっている。因子分析・回帰分析の結果によれば、前述の「並行開発」変数を含む「要素技術の早期集中探索」因子と「技術的な原因—結果の不確実性」因子の間には負の相関（1%水準）がみられる。つまり、不確実性と並行開発の間には明らかな相関があるという結果を得ている。

に、「U国の現場」が、現場の調整スピードは遅いが、科学知識ネットワークの活用に長けているとするなら、J国現場とU国現場は、設計リードタイム競争で、どんなときにどちらが勝てるのか。とりわけ、J国現場が、足は速いが事前によくものを考えない「浅慮なウサギ」であり、U国現場が、足は遅いが事前によく準備をする「周到なカメ」だとするなら、ウサギはどの条件で勝ち、いつ負けやすいのか。

これらを前節のパラメータを使って表現するなら、まず、科学的因果知識(Aの要素の分散)のレベルは「周到なカメ」の方が「浅慮なウサギ」より高いので、「J国現場=浅慮なウサギ」にとっての分散(σ^2_J)の方が、「U国現場=周到なカメ」にとっての分散(σ^2_U)より大きい($\sigma^2_J > \sigma^2_U$)。

一方、設計の第二段階(試行錯誤による最適解への接近)においては、「J国現場=ウサギ」は「U国現場=カメ」より速い。モデルでは「ウサギ」は「カメ」の3倍のスピードで設計調整ができるものと設定した。すなわち、調整所要時間 t_j で示すなら、 $t_j = (1/3)t_u$ である。

以上を前提に、「J国現場=浅慮なウサギ」と「U国現場=周到なカメ」のリードタイム競争のシミュレーションによる再現を試みたわけである。

以下、前述の「ウサギ」と「カメ」を様々な状況下に置いてリードタイム競争の勝敗を見ていこう(具体的なパラメータの値については、大隈・藤本(2006)を参照されたい)¹⁸。このモデルでは、

(i) 例えば、科学知識集約的でない「通常の製品」における、「統合型のJ国現場」と「分業型のU国現場」のリードタイム比較のシミュレーションは、因果知識を豊富に持つ設計者同士の競争、つまり「周到なウサギ(J)」対「周到なカメ(U)」の競争として設定できよう。

(ii) また、科学知識集約製品の領域で、科学知識の活用に消極的な「J国設計現場」が、U国現場並みの科学知識活用能力を身に着けたならどう変わるか、という疑問に対しては、「浅薄なウサギ(今のJ)」対「周到なウサギ(未来のJ)」の競争として表現できよう¹⁹。

¹⁸ 具体的なパラメータと評価基準の設定値は、たぶんに恣意的だが一貫している。大隈・藤本(2006)で用いた設定値と文章表現の関係は、以下の表の通り(ただしスピードsは、調整所要時間tの逆数)。

構造要素	Aの要素数	試行スピード	プログラム処理スピード	機能要求レベル	市場許容品質
多い	6×6=36	非常に速い	s		
中程度	4×4=16	速い	(1/3)s	厳しい	1.0
少ない	2×2=4	遅い	(1/9)s	緩い	3.0

また、勝敗の判定については、リードタイム比でJ国現場が3倍以上の差をつけて勝った場合、tやsで示したもとの3倍のスピード格差をさらに広げた他という意味で「大勝」と表現し、逆に、勝つには勝ったが1倍近くまで追いつけられたケースは「辛勝」と表現している。

¹⁹ このパターンでのシミュレーションはすでに大隈・藤本(2006)で試行している。その結果、①インテグラル製品で設計プロセスが漸進設計(コーディネーション)の場合に「周到なウサギ」の勝ち方が大きく、

(iii) ここでは、本来の「ウサギとカメ状況」すなわち「浅薄なウサギ (J)」対「周到的なカメ (U)」の競争に絞って、上記の設定によるシミュレーション実行の結果を素描しておく。

レース 1: インテグラル型×漸進設計 (コーディネーション): 「浅慮なウサギ (J国)」は「周到的なカメ (U国)」に対して、試行錯誤的な調整スピードは3倍速い ($t_J = (1/3) t_U$) が、事前の科学知識の入手レベルは「カメ (U)」が上である ($\sigma_J > \sigma_U$)²⁰。

シミュレーション結果によれば、①ウサギ (J国) の調整スピード (t_J) の、カメ (U国) のスピード (t_U) に対する優位がさほど大きくない (一定倍率以下である) 場合には、「ウサギとカメ」の教訓どおり「周到的なカメ (U国)」の「逃げ切り勝ち」となりやすい。②一方、ウサギの調整スピード (t_J) が (一定倍率以上で) カメのスピード (t_U) より圧倒的に速いならば、「浅薄なウサギ (J国)」の「逆転勝ち」となりやすい。③ただし、製品の複雑性 (n) が非常に高まると、再び、「周到的なカメ (U国)」が逃げ切り勝ちしやすくなる

		複雑性の指標		組織能力の指標		不確実性	市場条件		
製品名	設計組織	アーキテクチャ	構造要素数	試行スピード	最適化プロセス	科学技術レベル	機能要求レベル	勝敗	リードタイム比
半導体製造装置	浅慮なウサギ	インテグラル	中程度	速い	コーディネーション	低い	緩い	惜敗	1.4
	周到的なカメ	インテグラル	中程度	遅い	コーディネーション	高い	緩い	辛勝	1

		複雑性の指標		組織能力の指標		不確実性	市場条件		
製品名	設計組織	アーキテクチャ	構造要素数	試行スピード	最適化プロセス	科学技術レベル	機能要求レベル	勝敗	リードタイム比
半導体製造装置	浅慮なウサギ	インテグラル	中程度	非常に速い	コーディネーション	低い	緩い	勝	1
	周到的なカメ	インテグラル	中程度	遅い	コーディネーション	高い	緩い	負	2.2

		複雑性の指標		組織能力の指標		不確実性	市場条件		
製品名	設計組織	アーキテクチャ	構造要素数	試行スピード	最適化プロセス	科学技術レベル	機能要求レベル	勝敗	リードタイム比
半導体製造装置	浅慮なウサギ	インテグラル	多い	非常に速い	コーディネーション	低い	緩い	大敗	∞
	周到的なカメ	インテグラル	多い	遅い	コーディネーション	高い	緩い	大勝	1

つまり、組織的な問題解決能力の高さと、それがもたらす試行錯誤スピードの速さに頼る日本企業が、非常に複雑で先端科学知識集約的なインテグラル・アーキテクチャ製品では開発リードタイム競争に敗れる恐れがある、という一見逆説的な状況が、このシミュレーション結果で再現されている。非常に複雑なインテグラル製品であって、しかも科学技術集約的な製品 (例えば半導体露光装置) で、なおかつJ国現場 (例えば日本企業) がリードタイム競争で劣勢になる可能性がある、というパラドックスと整合的な結果が、シミュレーションで得られたわけである。

レース 2: モジュラー型×漸進設計 (コーディネーション): 次の例では、製品アーキテ

②モジュラー製品で漸進設計の場合も「周到的なウサギ」が勝つが、インテグラル製品ほどの勝ち幅ではなく、③並行開発 (コンペティション) の場合には、「周到的なウサギ」が「浅薄なウサギ」に勝つ確率はたかだか半分である。つまり、本来得意なはずのインテグラル型製品をコーディネーションで設計する「J国現場」こそ、追加で「科学知識の吸収能力」を構築することのメリットが大きい、というパラドックスが、このシミュレーションで再現されている。

²⁰ この設定は、開発の試行スピードは速いが科学知識獲得が苦手な日本の半導体製造装置メーカーと、開発試行スピードでは日本企業に負けるが科学知識へのアクセスでは勝る欧州企業がいるという、近年の半導体露光装置の状況に近い (中馬、2004)。

クチャはモジュラー型（例えばスーパーコンピュータ）で、設計プロセスは漸進設計（コーディネート型）型のままである。J国現場とU国現場の調整スピード（ $t_J = (1/3)t_U$ ）と科学知識に関する設定（ $\sigma^2_J > \sigma^2_U$ ）は、レース1と同様である。

シミュレーション結果によれば、インテグラル製品の場合（レース1）に比べ、「浅慮なウサギ（J国）」がカメに逆転勝ちする可能性が高い。設計要素数（n）が大きくなり、製品が複雑になっても、この傾向は変わらない。つまり、モジュラー型のケースでは、そもそも問題が簡単になるので知識による初期値の差が大きくなり、カメとウサギの距離差は大きくはない。したがって、浅慮なウサギ（J国）が比較的容易に逆転できる。

製品名	設計組織	複雑性の指標		組織能力の指標		不確実性 科学技術レベル	市場条件 機能要求レベル	勝敗	リードタイム比
		アーキテクチャ	構造要素数	試行スピード	最適化プロセス				
スーパーコンピュータ	浅慮なウサギ	モジュラー	多い	速い	コーディネート	低い	厳しい	勝	1
	周到なカメ	モジュラー	多い	遅い	コーディネート	高い	厳しい	負	2.7

レース3：インテグラル型×並行設計（コンペティション）：調整スピードと科学知識レベルに関する設定はレース1、2と同様である。

シミュレーション結果によれば、「浅慮なウサギ（J国）」が勝ちやすい。しかし、製品が複雑化（設計要素数nが増加）すると、ウサギ（J国）、カメ（U国）ともに最適解との乖離が大きくなり、要するに機能面の設計費が低下する。ただし、当該製品が市場要求を一応満たして成功する確率は、概してコーディネート型の場合よりも高い。このように、効率や品質は相対的に悪いが、手堅くまとめる「並行開発」の特徴が出ている。要するに、並行開発（コンペティション）型では、科学的知識の利用能力の差（ σ^2 の差）がリードタイムにあまり影響しないので、スピードのある「浅慮なウサギ」が常に有利となりやすい。

製品名	設計組織	複雑性の指標		組織能力の指標		不確実性 科学技術レベル	市場条件 機能要求レベル	勝敗	リードタイム比
		アーキテクチャ	構造要素数	試行スピード	最適化プロセス				
半導体製造装置	浅慮なウサギ	インテグラル	中程度	速い	コンペティション	低い	厳しい	勝	1
	周到なカメ	インテグラル	中程度	遅い	コンペティション	高い	厳しい	負	3

シミュレーション結果4：モジュラー型製品の並行競争型開発：調整スピードと科学知識レベルに関する設定はレース1、2と同様である。

シミュレーション結果によれば、「浅慮なウサギ（J国）」が勝ちやすい。製品の複雑性（n）や科学知識集約度（ σ^2 ）の高低にかかわらず、常にJ国の現場が勝つ。

製品名	設計組織	複雑性の指標		組織能力の指標		不確実性 科学技術レベル	市場条件 機能要求レベル	勝敗	リードタイム比
		アーキテクチャ	構造要素数	試行スピード	最適化プロセス				
スーパーコンピュータ	浅慮なウサギ	モジュラー	多い	速い	コンペティション	低い	厳しい	勝	1
	周到なカメ	モジュラー	多い	遅い	コンペティション	高い	厳しい	負	3

以上が、シミュレーションが再現した「ウサギとカメ」状況である。むろん、暫定的な結果であるし、アドホックな結果解釈も戒めねばならないが、このシミュレーション分析

では一定の収穫も得られたと考える。第 1 に、現実の設計プロセスが持つ本質的な特徴を幾つか再現したシミュレーション・モデルを作ることができたと考える。第 2 に、このモデルを用いたシミュレーション実行結果は、概ね、現実のケースで常識として言われている幾つかの傾向と整合的であった。第 3 に、「浅慮なウサギ (=J 国現場、例えば日本企業)」に関しては、概ね実証研究から抽出された「定型的事実」と矛盾の無い結果がえられた。すなわち、以下のような点で、今回のシミュレーション結果は「定型化された事実」と整合的である。

- ① 「浅慮なウサギ (J 国)」は密なコミュニケーションを必要とする漸進設計 (コーディネーション) 型の設計プロセスにおいて「周到なカメ (U 国)」に対する競争優位を發揮している。緊密なコミュニケーションに基づくコーディネーション (連携調整) は、藤本 (2005) の言う「統合型ものづくり」の必須条件である。
- ② 「浅慮なウサギ (J 国)」が得意なコーディネーション型の設計プロセスは、厳しい市場の条件を満たすことが可能である。これは、消費者が品質にうるさい日本市場で日本企業が強いという事実と整合的である。
- ③ 製品のための科学知識が形成途上にあるような科学知識集約商品の場合、仮に製品が複雑でインテグラルなケースでも、「浅慮なウサギ」が優位性を失いやすい。「製品がインテグラル型でも J 国現場が負ける、というこのパラドックスは、今回のシミュレーションの結果でも再現されている。
- ④ 複雑度が中程度、例えばアーキテクチャはインテグラル型だが設計要素数 (n) があまり大きくないケースでは、今回のシミュレーションでも概して「浅慮なウサギ (J 国)」が有利であった。

このように、今回試行したシミュレーション・モデルは、およそシミュレーション・モデルが持っている弱点を全て背負っているとはいえ、それなりにかなりのリアリティを持つように仕上がったと筆者は考える。第 1 に、モデルの基礎となった「公理系設計」と「2 段階設計モデル」は、極めてシンプルなモデルとはいえ、現実の設計活動がもつ本質的な部分のある程度描写できていると考える。第 2 に、シミュレーションの結果示された、設計リードタイムとモデルに含まれる諸変数の関係の中に、少なくとも常識に著しく反するものは見つかっていない。

そして第 3 に、本稿の冒頭で述べた「定型的事実」を「2 段階設計プロセス」を基礎とする今回のシミュレーションで再現することが、ここでの第一義的な目的であったが、一つのめどが立ったといえそうだ。

3-4 まとめ：設計プロセス論による設計比較優位説の再現

本稿では、「組織能力とアーキテクチャの適合性が高いと設計面の競争優位が生まれる」という「設計の比較優位説」に対して、設計プロセス論的リアリティを付け加える試みを行った。具体的には、工学系の公理系設計論 (Suh, 1990, 2001; 中尾, 2003, 中尾・畑村・服部, 1999) をヒントに、企業の設計活動を、「機能設計パラメータ群=f (構造設計パラメータ群)」という連立方程式を解く問題にたとえて定式化し、これにより、統合型の開発組織が調整重視の設計プロセスを通じて、インテグラル型アーキテクチャにおいて競争優位を得る経路をシンプルな例で素描した。この例では、まず、何らかの形で (たとえば不正確な連立方程式を解くことによって) 構造設計パラメータ群の初期値を設定し、次に試行錯誤によって目標とする機能パラメータへ漸近させる、という「2段階設計プロセス」に近い事例が現実にも多いと考え、これによって日本企業の設計開発活動を近似しようと考えた。

すなわち、日本企業の研究開発活動は、下流 (開発による新製品・工程の創出) においては、統合型組織能力を生かした迅速な試行錯誤によるリードタイム短縮を特徴としているが、上流 (研究による科学知識の創出) においてはオープンな科学者ネットワークによる科学知識創造を軽視する傾向がある (中馬, 2004)、という定型的な事実 (stylized fact) を出発点として、日本企業の持つアーキテクチャの優位性を「2段階設計プロセスモデル」で説明しようとした。その過程で、上記のような組織能力を持つ日本企業は、中程度の複雑性を持ち科学知識の蓄積の十分なインテグラル・アーキテクチャ製品では国際競争力を持つが、モジュラー型アーキテクチャの製品では設計コストの比較優位がないこと、またその反面、科学知識が蓄積途上であるような科学知識集約的で複雑なインテグラル製品では日本が競争力を持たない可能性があることを例示した。以上は、実際に観察される事実をベースに組み立てた「アーキテクチャと組織能力の比較優位説」である。

しかしながら、実際に企業の設計現場で参与観察をすることで、上記のような「2段階設計プロセス」の実証データを収集することは容易ではない。そこで本稿では、代替的なアプローチとして、上記の「公理系設計論」をベースにするシミュレーション分析を援用してみることにする。

本論で前提とする、日本の製品開発に関する「定型化された事実」は以下の通りである。

- ① 戦後日本企業は生産資源が不足する中での急成長を通じて、長期雇用・長期取引をベースとする「統合型ものづくり」の組織能力を構築してきた。そうした企業では、チームによる製品開発が発達し、技術者間・チーム間の試行錯誤による設計最適化のス

ピードも速い。

- ② 日本の消費者は品質にうるさいので、企業は、より高い精度で設計パラメータの最適化を行う必要がある。
- ③ 日本企業は、科学知識が形成途上にあるような先端商品よりも、ある程度科学知識は確立している「非ハイテク製品」で国際競争力を持つ傾向がある。
- ④ 日本企業は、中程度の複雑度を持つ「インテグラル」型製品で国際競争力を持つ傾向がある。

こうした「定型的事実」を、2段階設計プロセスを基礎とするシミュレーションで再現をこころみた。仮にそれができるのであれば、国際比較優位というマクロ的なテーマに、設計プロセス論というミクロ的・工学的な基礎を与えることにつながるかもしれない。

参考文献

- Alexander, C. (1964) *Notes on the Synthesis of form*. Cambridge, MA: Harvard University Press.
- 青木昌彦 (1995) 『経済システムの進化と多元性：比較制度分析序説』 東洋経済新報社.
- 青木昌彦・安藤晴彦編著 (2002) 『モジュール化—新しい産業アーキテクチャの本質』 経済産業研究所・経済政策レビュー. 東洋経済新報社.
- 青島矢一・武石彰 (2001) 「アーキテクチャという考え方」 藤本隆宏・武石彰・青島矢一編 『ビジネス・アーキテクチャー：製品・組織・プロセスの戦略的設計』 有斐閣. 27-70.
- 浅沼万里 (1997) 『日本の企業組織—革新的適応のメカニズム』 有斐閣.
- Baldwin, C. Y. and Clark, K. B. (2000) *Design Rules, Vol. 1: The Power of Modularity*. Cambridge, MA: MIT Press. (安藤晴彦訳 (2004) 『デザイン・ルール』 東洋経済新報社)
- 中馬宏之 (2004) 「日本のサイエンス型産業が直面する複雑性と組織限界：半導体露光装置産業の事例から」 『一橋ビジネスレビュー』 52 (3) 64–85.
- 中馬宏之 (2007) 「サイエンス・イノベーションの時代：特集にあたって」 『一橋ビジネスレビュー』 54 (4) 4–5.
- 中馬宏之・橋本哲一 (2007) 「ムーアの法則がもたらす複雑性と組織限界：」 『一橋ビジネスレビュー』 54 (4) 22–43.
- Clark, K. B., Chew, W. B. and Fujimoto, T. (1987) *Product Development in the World Auto Industry. Brookings Papers on Economic Activity, Special Issue, Issue 3, 729–781.*
- Clark, K. B. and Fujimoto, T. [1991], *Product Development Performance: Strategy, Organization, and Management in the World Auto Industry*. Boston: Harvard Business School Press. (田村明比古訳 (1993) 『製品開発力』 ダイヤモンド社)
- Cusumano, M. A. and Selby, R. W. (1995) *Microsoft Secrets*. New York: Free Press.
- Fujimoto, T. (1989). *Organizations for effective product development: The case of the global automobile industry*. Unpublished DBA dissertation, Harvard University Graduate School of Business Administration, Boston.
- Fujimoto, T. (1999) *The Evolution of a Manufacturing System at Toyota*. New York: Oxford University Press.

- 藤本隆宏 (1997) 『生産システムの進化論』 有斐閣.
- 藤本隆宏 (2001a) 「アーキテクチャの産業論」 藤本隆宏・武石彰・青島矢一編 『ビジネス・アーキテクチャー：製品・組織・プロセスの戦略的設計』 (1章) 有斐閣.
- 藤本隆宏 (2001b) 『生産マネジメント入門 II』 日本経済新聞社.
- 藤本隆宏 (2003a) 「組織能力と製品アーキテクチャー下から見上げる戦略論」 『組織科学』 36 (4) 11-22.
- 藤本隆宏 (2003b) 『能力構築競争 日本の自動車産業はなぜ強いのか』 中央公論新社.
- 藤本隆宏 (2004) 『日本のものづくり哲学』 日本経済新聞社.
- 藤本隆宏 (2005) 「アーキテクチャの比較優位に関する一考察」 『赤門マネジメント・レビュー』 4 (11) 523-548.
- 藤本隆宏 (2007) 「設計立地の比較優位：開かれたものづくりの観点から」 『一橋ビジネスレビュー』 55 (1) 22-37.
- Fujimoto T., Ge D. and Oh, J. (2006) Competition and Co-operation in Automotive Steel Sheet Production in East Asia, MMRC Discussion Paper No. 73.
- 藤本隆宏・西口敏宏・伊藤秀史編 (1997) 『サプライヤー・システム』 有斐閣.
- 藤本隆宏・大隈慎吾 (2007) 「設計立地の比較優位に関する試論—枠組・実証・シミュレーション」 RIETI ディスカッションペーパー 07-J-025.
- 藤本隆宏・新宅純二郎編 (2005) 『中国製造業のアーキテクチャ分析』 東洋経済新報社.
- 藤本隆宏・東京大学 21 世紀 COE ものづくり経営研究センター (2007) 『ものづくり経営学』 光文社.
- 藤本隆宏・武石彰・青島矢一編 (2001) 『ビジネス・アーキテクチャー：製品・組織・プロセスの戦略的設計』 有斐閣.
- 藤本隆宏・安本雅典 (2000) 『成功する製品開発—産業間比較の視点』 有斐閣.
- Fujita, M., P. R. Krugman and A. J. Venables (1999) , The Spatial Economy: Cities, Regions, and International Trade, MIT Press. (藤田昌久、P. R. クルーグマン、A. J. ベナブルズ [2000] 『空間経済学—都市、地域、国際貿易の新しい分析』 小出博文訳、東洋経済新報社)

- Grant, R.M. [2005] *Contemporary Strategy Analysis* (5th Edition) Blackwell, Oxford.
- 平井敏彦他著, 小早川隆治編 (2003) 『マツダ／ユーノスロードスター：日本製ライトウェイトスポーツカーの開発物語』三樹書房.
- Iansiti, M. (1998) *Technology Integration: Making Critical Choices in a Dynamic World*. Boston, MA: Harvard Business School Press.
- 伊藤元重・奥野正寛・清野一治・鈴木興太郎 (1988) 『産業政策の経済分析』東京大学出版会.
- 小池和男 (1999) 『仕事の経済学』(第2版) 東洋経済新報社.
- 國領二郎 (2005) 『オープン・ネットワーク経営：企業戦略の新潮流』日本経済新聞社.
- クルグマン, P. R., オブズフェルド, M. (1996) 石井・浦田・竹中・千田・松井訳 『国際経済 (第3版) I 国際貿易』新世社.
- 桑嶋健一 (2006) 『不確実性のマネジメント：新薬創出の R&D の「解」』日経 BP 社.
- Leachman, R. C. and Hodges, D. A. (1996) Benchmarking Semiconductor Manufacturing. *IEEE Transaction on Semiconductor Manufacturing*, 9 (2) 158–169.
- 長岡貞男 (1999) 『内外価格差の経済分析—生産性からのアプローチ』NTT 出版.
- 中尾政之 (2003) 『創造設計学』丸善.
- 中尾政之・畑村洋太郎・服部和隆 (1999) 『設計のナレッジマネジメント』日刊工業新聞社.
- Nelson, R. R and Winter, S. G. (1982) *An Evolutionary Theory of Economic Change*. Cambridge, MA: Harvard University Press.
- Nishiguchi, T. (1994) *Strategic industrial sourcing: The Japanese advantage*. New York; Tokyo: Oxford University Press. (西口敏宏 (2000) 『戦略的アウトソーシングの進化』東京大学出版会)
- 延岡健太郎 (1996) 『マルチプロジェクト戦略』有斐閣.
- 大隈慎吾・藤本隆宏 (2006) 「設計プロセスとアーキテクチャの競争優位」東京大学ものづくり経営研究センターディスカッションペーパー MMRC-J-70.
- 大鹿隆・藤本隆宏 (2006) 「製品アーキテクチャと国際貿易論の実証分析」東京大学ものづ

くり経営研究センターディスカッションペーパー MMRC-J-72.

Pisano, G. P. (1997) *The Development Factory: Unlocking the Potential of Process Innovation*. Boston: Harvard Business School Press.

Samuelson, P. A. (2004) Where Ricardo and Mill Rebut and Confirm Arguments of Mainstream Economists Supporting Globalization. *Journal of Economic Perspectives*, 18 (3) Summer 135–146.

Simon (1969) *The Sciences of the Artificial*. Cambridge, MA: MIT Press.

Simon, H. A. (1947, 1997) *Administrative Behavior: A Study of Decisionmaking Processes in Administrative Organization* (4th ed.). New York: Free Press. (二村敏子他訳 (2009) 『経営行動：経営組織における意思決定過程の研究』ダイヤモンド社)

Suh, N. P. (1990) *The Principles of Design*. New York: Oxford University Press. (畑村洋太郎監訳 (1992) 『設計の原理—創造的機械設計論—』朝倉書店)

Suh, N. P. (2001) *Axiomatic Design: Advances and Applications*. Oxford University Press, New York. (中尾政之・飯野謙次・畑村洋太郎共訳 (2004) 『公理的設計』森北出版)

澤田康幸 (2003) 『国際経済学』新世社.

関満博 (1993) 「フルセット型産業構造を超えて」中公新書.

椙山泰生 (2001) グローバル化する製品開発の分析視角：知識の粘着性とその克服』『組織科学 35 (2) 81–94.

高増明・野口旭 (1997) 『国際経済学』ナカニシヤ出版..

武石彰 (2003) 『分業と競争：競争優位のアウトソーシング・マネジメント』有斐閣.

竹森俊平 (1995) 『国際経済学』東洋経済新報社.

Ulrich, K. T. (1995) Product Architecture in the Manufacturing Firm. *Research Policy*, 24, 419–440.

Vernon, R. (1966) International Investment and International Trade in the Product Cycle. *Quarterly Journal of Economics*, 80 (2) 190–207.

Von Hippel, E. (1994) Sticky Information and the Locus of Problem Solving: Implication for Innovation. *Management Science*, 40 (4) 429–439.