

2008.4.3 4247

2/23

平成 20 年 1 月 12 日

付録

付録 1. 「牧師補佐の傑作 — 素晴らしい一頭立て二輪馬車」 オリバー・ウェンデル・ホームズ (1895、p.158)

1858 年に発表されたこの詩は、ペトロスキー (1992、p.29) が指摘しているように、構造物の最適化と工学一般に当てはまる概念を見事に描写している。奇しくもハーバード大学の解剖学教授であり、将来の最高裁判所判事の父でもあった詩人らしい諧謔的な調子で、O. W. ホームズ (図 A1.1) は「いかなる困難があろうと完全に知的な概念」として次の説明をしている。「どのような機械でも、観察すればどこが壊れやすいかがわかる。たとえば馬車ならば、弱点は車軸がハブまたはこしきに入る部分である。馬車が壊れるときは、私の考えでは 4 回に 3 回はそこが壊れる。職人はそこが絶対に壊れないように直すべきである。そして次に脆弱な部分を見つけ、といった具合に続けてゆくと、ついに論理的には牧師補佐が達成したような完璧な結果にいたる」。

牧師補佐は彼の「一頭立て二輪馬車」について、次のような均一の信頼度で性能規定の仕様を考え出した。

さて、一頭立て二輪馬車を造るときは、いいかね、必ずどこかにもっとも弱い部分があるものだ。

ハブ、輪縁、ばね、あるいは轆かもしれない。

羽目板、あるいは横木、あるいは床、あるいは下枠かもしれない。

ねじ、締め釘、貫帯、まだ隠れている。

どこが弱いのか見つけなければならない、そして見つかるだろう。

上か下か、あるいは内か外か。

それが原因だ、間違いない。

一頭立て二輪馬車が古びていないのに壊れるのは。

.....

この素晴らしい計画のためには、

もっとも弱い部分がひずみに耐えられなければならない。

そしてそのためには、私が断言したように、

その部分をそれ以外の部分と同じくらい強くすればよい。

100 年後のある日：

一頭立て二輪馬車には老化の形跡がある。

全体が穏やかに朽ちている。

しかし局所的に目立って痛んでいるわけではない。

達成するのはきわめて難しい  
鉄'け'れ'でも

ルビ、ふりがな

そして突然、二輪馬車は塵と化した。

おわかりかな、もちろん馬鹿じゃなければね。  
どんなふうにも馬車が一挙にばらばらになったかが。  
一挙にだ、どこかが最初に壊れたのではない。  
まるで泡が破裂するように。

泡は一様に最適化された形状なので、このアナロジーは正しい。最適化と破壊解析に対する警告として、この詩は今でも有効である。「脆弱性」という用語は、第4章での用法と同じように用いられている。車軸受けに言及した箇所は、似たような橋梁のディテールを思い出させる。いくらかふまじめな「論理は論理、それが私の言いたいことだ」という結語は、全体の論証方法に疑問を投げかけている。

ひょっとすると、オリバー・ウェンデル・ホームズ・ジュニア (図 A1.2) が次のように結論づけたのは、この父親の警告に対する答えだったのかもしれない。「法の生命は論理ではなく、経験であった」(図 A1.3)。

法とは対照的に、工学は経験によって確認された論理に立脚している。詩に描かれたユニークな経験は供用寿命が 100 年であることを示唆している。では、設計寿命はどのくらいとすべきだろうか。二輪馬車はいつ、どのような基準で安全ではなくなったのだろうか。

その過程が建造、使用、維持管理といった独立した作業の連なりとみなされ、それぞれに 10%の破壊尤度が伴うとすれば、累積信頼度は受け入れられないレベルになるはずである(以下の付録を参照)。この過程に点検を加えなければならず、破壊モードに関して手がかりのない点検の信頼度は 50%以下になる。

この観点からすれば強度が一様な二輪馬車は管理しにくく、だからこそ、そのようなものを造るインセンティブがあまり働かないのであろう。不完全であっても予測可能な構造物は競争力があり、ライフサイクルマネジメントの対象にしやすい。

(図タイトル)

図 A1.1 オリバー・ウェンデル・ホームズ (1809-1894)、栄誉の殿堂、ブロンクス

図 A1.2 オリバー・ウェンデル・ホームズ・ジュニア (1841-1935)

図 A1.3 図 A1.2 の記念像の銘刻文

どこれでもかきとるん

この説明唐突?

(文)

前後の中でいふこと

前後の  
あか?

何のことか?

## 付録 2. ベイズ統計的決定理論と信頼性設計

コルネイユ (Freudenthal の文献、1972、pp.47-66) は、構造物の信頼性へのベイズ統計理論の適用を以下のように提示している。

### 不確定性の分析

パラメータの統計的不確定性は、確定論ではなく、物理的メカニズムの確率モデルに固有の確率的不確定性と並行して処理される。判断を伴う確率と構造性能のさまざまなレベルの相対便益または効用を抽出あるいは評価するための技法が提案されている..... 統計的決定理論には二つの新しい概念が取り入れられており、この二つの概念はいくらか分けて考えることができる。一方の概念は統計的であり、パラメータの不確定性の分析と関係づけられる。他方の概念は決定分析であり、決定によるさまざまな成果の便益と期待便益の定量化にかかわる...

決定理論では、どの不確定因子も確率変数として扱うことができる。よく知られた確率変数、すなわち物理的過程の確率モデルのアウトプットに加えて、確定的モデルまたは確率モデルのパラメータも、さらにはモデルそのものさえ確率変数として扱うことができる。固有のランダム性、物理的ランダム性、あるいはモデル化されたランダム性に加えて、技術者は常にモデルのパラメータ値についても不確定である。この不確定性は、より信頼できるパラメータ推定値を求めるために利用できる過程の観測結果が多いほど小さくなるという意味において、(確率的不確定性と対立するものとして) 統計的不確定性と呼ぶことができる。さらに、モデル自体に常に根本的な不確定性が残る。材料の実際の挙動は線形ではなく、疲労寿命はワイブル分布またはガンマ分布 (あるいは、もっと可能性あるものとして、まだ名前がつけられていないなんらかの分布) に従う可能性がある。ここで重要なのは、これらすべての不確定性が工学的設計の選定に適切に盛り込まれるようにすることを目的とした、不確定性の原因の一貫性のある並行処理である....

ベイズ統計的決定理論は、職業上の情報と (...) 統計データに含まれる情報を結びつけるための手順に大きく寄与する。その手段はベイズの定理である。この定理を簡単に言うと、観測 A が与えられたときの仮説 i の事後確率  $P [H_i | A]$  は、 $H_i$  が与えられたときの A の尤度、すなわち  $P [A | H_i]$  と事前確率  $P [H_i]$  との積に比例するというものである。

(式 A2.1)

決定解析では、行動-成果の対  $u (a_i, \theta_j)$  ごとに選好の測度、相対値、あるいは効用を割り当てなければならない。効用に割り当てる数値は単純に貨幣換算してもよ

いが、一般的には定められた決定基準が有効になるような方法で割り当てなければならない。この決定基準は、期待効用を最大化する行動または設計が選定されることである。

正式には、ベイズの決定理論によれば、意思決定者は代替行動  $a_1, a_2, \dots, a_m$  の中から  $a_0$  を選択するとき、行動が与えられたときの期待効用  $E[u | a_i]$  が最大になるように選択する。期待値は、 $a_i$  が与えられたときの可能事象の集合  $\theta_1, \theta_2, \dots, \theta_n$  にわたる期待値である。

(式 A2.2)

期待値の最大化は、決定が一連の同じような多くの小さい決定のなかの一つである場合は明白な基準となる。これを構造設計で通常遭遇するような他に類のない重要な決定に用いるには、さらに検討が必要である。このような場合、期待貨幣価値は妥当な決定基準にはならないはずである。職業上の評判の低下や人命の喪失などのように数量化しにくい事象を貨幣ターム（たとえば E. Rosenblueth 博士が私信で述べているように、同業者または社会が損失を回避するために支払う意志のある金額）で表わすことができたとしても、期待貨幣価値は妥当な決定基準にはならない... 大半の企業がこのような潜在的もうけを考えて一か八かで危険を冒すと思うのは妥当ではない...

構造設計への決定理論の適用については、多くの研究者が報告している.... これらの研究者が確率の定義にあたって明確にベイズ的立場をとっているとは限らないし、期待値基準が常に直観以外のなんらかの根拠にもとづいて正当化されているわけでもない。あるいは貨幣以外の価値が適用されているわけでもない。初期コスト増大と不良性能のリスク増大とのバランスを明確にとるという概念は、長い間、構造設計への確率論的アプローチの魅力的な利点であった。しかし、この概念は、現実的、実用的アプローチとしては、すべての不確定性の原因が考慮されるわけではなく、期待貨幣価値基準に限界があることから批判されている。正式なベイズ決定理論はこれらの難点を取り除く。

↑  
ベイズ... と サホトに... ので?!

### 付録 3. 「戦いに勝ったマシン」 1. アシモフ (1990)

スーパーコンピュータは敵に対して最終的に勝利したことであまねく称賛されている。若いチーフプログラマー、中年のチーフデータベース管理者（たとえばオラクル）、年配のチーフエグゼクティブが密かに思っていたことをそれぞれ個人的に、声に出して言いはじめた。チーフプログラマーは、今では「ガイゴー（入力为正しくなければ出力情報も正しくない）」として知られるようになった原理を思い出させる。彼は、宇宙のあちこちから寄せられた膨大な量の無意味な情報をかみ砕いて一貫性のあるデータインプットにまとめた