大韓機械學會誌 第18卷 第1號 pp. 28~33, 1978

自 動 設 計

白　　南　　柱*

Antomatic Design

Nam-ju park

1. 緖　　論

設計는 設計條件을 만족하는 범위에서 가장 좋은 設計點을 探索하는 過程이다. 이 過程은 그림 1과 같은 作業을 몇번 되풀이하여 設計點이 얻어진다.

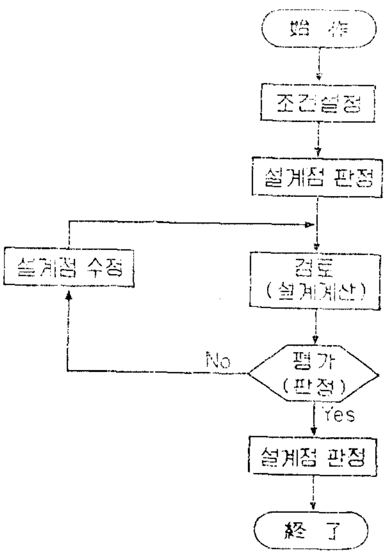

그림 1. 設計의 手順

이 設計點의 評價와 修正作業을 最適化라 한다. 最適化操作이 自動化되는 條件은

(1) 設計點을 얻는 것이 經驗이나 直觀에 의해서, 求하기가 어려울 경우

(2) 設計點의 定量的 決定이 重要한 경우

(3) 定量的 評價가 可能한 경우

最適化의 操作을 完全히 自動化하기 위해서는 設計點에 대한 評價가 定量化되어야 한다. 例를 들면 外觀등과 같은 美的 要素가 덜어있는 것은 現在로서는 定量的 評價가 不可能에 가깝다. 따라서 이 3개 條件 가운데서 特히 重要한 것은 (3)의 定量的 評價의 可否다. 이것이 不可能하거나 困難할때는 最適化操作을 人間이 하여야 하고 自動設計 system은 Man-Machine system이 된다. 이 代表的인 例가 CRT를 使用한 CAD system이다. 이에대해 設計點을 定量的으로 評價할 수 있을 때는 computer를 主體로한 system을 만들수 있고 이것을 AD system이라 한다. AD system에서는 計劃的인 探索法(最適化의 手法)을 쓸 수 있고 設計의 速度는 매우 빠르다. 그러나 이 system에서는 設計途中에 check가 힘들게 된다. CAD system에서는 設計點의 探索이 試行錯誤的이다.

2. 情報檢索形 自動設計 system (AD system)

設計條件과 評價基準이 定해지면 最適化毛法을 써서 自動的으로 最適設計點을 얻을 수 있는 設計 system을 만들수 있으며 그 最適化 方法은 다음과 같다.

- 同時 最適化方法
  - 線形系(LP法)
  - 非線形系(여러가지 探索法)
- 分割 最適化方法 (段層的 最適化)
  - 要素의 入出力을 指定하여 分割하는 方法
    - cascade-DP法
    - 回路網-P行列方法
  - 補助變數를 導入하여 分割하는 方法
    - LP法에 의한 分解原理
    - MP法, 價格法

最適化라 함은 政策變數 $P_1 \cdots P_n$의 函數인 評價函數 $f(p_1, \cdots P_n)$을 最大로 하는 P_i의 값을 求하는 것이다. 이 때 評價函數를 分解하지 않고 最適化하는 方法을 同時 最適法이라 하고 評價函數가 system의 要素에 關한 合 $f(p)=\Sigma f_i(p_i)$로 나타내어질 때 system의 最適化 問題를 各要素에 關한 最適化 問題로 分割하는 方法을 分割最適化法이라 한다. 이때는 各要素에 關한 最適化와 더부러 system 全體에 關한 最適化가 꼭 必要하다.

* 正會員, 釜山大學校 工科大學 機械工學科

큰 system에서는 變數의 數가 많으므로 同時 最適은 無理이고 分割最適法을 써야 한다. 이 最適法에서 重要한 點은 어떻게 하면 全體의 最適化와 모순이 되지 않고 部分最適化를 할수 있느냐 하는 點이다. 이 最適法에는 두가지 基本解決法이 있다. 하나는 system으로서 許容되는 入出力을 各 要素에 分割하여 그 條件下에서 要素를 部分最適化하는 方法이고 또 하나는 價格과 같은 補助變數를 導入하여 要素를 部分最適化하는 方法이다. DP法은 前者에 屬하고 MP法은 後者에 屬한다.

(1) LP法(線形系)

線形 問題는 具體的으로 다음 形式을 하고 있다.

constraint $g_i(x)$ $(i=1,\ 2,\cdots m)$ 및 目的 함수 $f(x)$가 모두 線型인 경우를 線型計劃法이라 하고 이 問題는 convex 問題이다. Linear Programming 問題는 다음 式으로 나타낸다.

$$\text{constraints: } \sum_{j=1}^{n} a_{ij}\ x_{ij} \leqq b_i \quad (i=1,2,\cdots n) \tag{1}$$

$$x_j \geqq 0 \quad (j=1,\ 2,\cdots n) \tag{2}$$

$$\text{objective}: Z=\sum_{j=1}^{n} c_j x_j \rightarrow \min \tag{3}$$

일반적으로 서로 一次獨立인 式이 m 變數가 n $(m \leqq n)$의 경우에는 $(n\text{-}m)$의 變數를 parameter로 하여 다른 m의 變數를 구할 수 있다. 이 解가운데 $(n\text{-}m)$개의 parameter를 0으로 하였을 때의 解를 基底解라 한다. 基底解가운데 各 變數의 값이 0이 아닌것을 實行可能解라하고 0이 아닌 변수를 基底變數라 한다. 이 線形問題의 最適解를 구하는데 simplex method를 쓴다.

a) Simplex method

simplex method는 한 개의 實行可能解에서 出發하여 目的함 수의 값을 減小하도록 基底變數를 차례로 바꿈으로써 最適解를 구하는 方法이다.

式 (1)에서 b_i를 正이라 하고 Slack variable (余剩變數)를 加하여 변형하면 다음과 같다.

$$\left.\begin{aligned} &a_{11}x_1+a_{12}x_2+\cdots\cdots+a_{1n}x_n+x_{n+1} &&=b_1 \\ &a_{21}x_1+a_{22}x_2+\cdots\cdots+a_{2n}x_n+x_{n+2} &&=b_2 \\ &\quad\vdots \\ &a_{m1}x_1+a_{m3}x_2+\cdots\cdots+a_{mn}x_n+x_{n+m} &&=b_m \end{aligned}\right\} \tag{4}$$

이 때의 實行可能解의 한 예는 $x_1=0, \cdots,\ x_n=0,\ x_{n+1}=b, \cdots x_{n+m}=b_m$ (3)을 변형하면

$$Z\text{-}c_1x_1\text{-}c_2x_2\cdots\text{-}c_nx_n=0 \tag{5}$$

또 (5)를 表記하는데 便利하도록 變形한다.

$$z+c_1'x_1+c_2'x_2+\cdots+c_n'x_n=0 \tag{5'}$$

表 1.

基底變數	b_i	x_1	$x_2\cdots\cdots x_n$	x_{n+1}	$x_{n+2}\cdots\cdots x_{n+m}$
z	b_0	c_1'	$c_2'\cdots\cdots c_n'$	0	$0\cdots\cdots 0$
x_{n+1}	b_1	a_{11}	$a_{12}\cdots\cdots a_{1n}$	1	$0\cdots\cdots 0$
x_{n+2}	b_2	a_{21}	$a_{22}\cdots\cdots a_{2m}$	0	$1\quad 0\cdots\cdots 0$
$\vdots$		$\vdots$	$\vdots\qquad\vdots$	$\vdots$	0
x_{n+m}	b_m	a_{m1}	$a_{m2}\cdots\cdots a_{mn}$	0	$0\quad 0\cdots\cdots 1$
z	b_0^*	c_1^*	$c_2^*\cdots\cdots c_n^*$	c_{n+1}	$0\cdots\cdots 0$
x_2	b_1^*	a_{11}^*	$1\quad a_{13}^*\cdots a_{1n}^*$	a_{1n+1}	$0\cdots\cdots 0$
x_{n+2}	b_2^*	a_{21}^*	$0\quad a_{23}^*\cdots a_{2n}^*$	a_{2n+1}^*	$1\cdots\cdots 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$0\cdots\cdots 0$
x_{n+m}	b_m^*	a_{m1}^*	$0\quad a_{m3}^*\cdots a_{mn}^*$	a_{mn+1}^*	$0\cdots\cdots 1$

(4)와 (5)를 변형하여 기저변수로 바꿀 수 있다. 예를 들컨,

x_2를 기저변수라 하고 x_{n+1}을 기저 변수에서 제거하려면

$$\left.\begin{aligned} &a^*_{11}x_1+x_2+a_{13}x_3+\cdots+a_{1n}x_n+a_{1,\ n+1}\ x_{n+1}=b_1^* \\ &a_{21}^*x_1+\quad+a_{23}^*x_3+\cdots+a_{2n}^*x_n+a_{2,\ n+1}^*\ x_{n+1} \\ &\quad\vdots \qquad\qquad\qquad\qquad\qquad\qquad +x_{n+2}=b_2^* \\ &a_{m1}^*x_1+\ +a_{m3}^*x_3\cdots+a_{mn}^*x_n+a_{m,\ n+1}^*x_{n+1} \end{aligned}\right\} \tag{6}$$

$+x_{n+m}=b_m^*$ }

로 변형하면 $x_1=0$ $x_2=b_1^*$ $x_3=0, \cdots, x_n=0,$ $x_{n+1}=0,$ $x_{n+2}=b_2^*\cdots,$ $x_{n+m}=b_m^*$로 하여 基底解가 구해진다. 表에서 말하면 x_2의 列과 x_{n+1}의 行의 交點의 係數 a_{12}를 1로 하고 다른 x_2의 列의 要素 $(a_{22}, \cdots, a_{m2})$를 0으로 하는 것을 뜻한다. 이와 같이 하는 것을 sweep out라 하고 a_{12}와 같이 1로 하는 要素를 pivot라 한다.

a_{ij}를 pivot로 하여 sweep out하려면 다음과 같이 한다.

(1) 제 i行을 a_{ij}로 나눈다.

$$a_{ip}^*=a_{ip}/a_{ij} \quad b_i^*=b_i/a_{ij} \tag{7}$$

(2) 제 q行 $(q\neq i)$의 各要素를 다음과 같이 한다.

$$\left.\begin{aligned} &a_{qp}^*=a_{qp}-a_{ij}^*\times a_{qj} \\ &b_q^*=b_q-b_i^*\times a_{qj} \quad (q\neq 0) \\ &c_p^*=c_p'-a_{ij}^*\times c_j^1 \\ &b_0^*=b_0-b_i^*\times c_i' \end{aligned}\right\} \tag{8}$$

여기서 x_j가 제 i行의 基底變數로 변한 기저 변수로 되었으므로 제 i行의 기저변수의 欄은 x_j로 한다. 이 때의 b_j^* 값이 새로운 z의 값이다.

最適解를 구하려면 各各

$$b_q^*\geqq 0 \quad (q\neq 0) \quad b_0^*\leqq b_0$$

가 되도록 sweep out를 반복한다. 결국 pivot의 要素 a_{ij}를

$$c_j'>0$$
$$b_i/a_{ij}=\min_q b_q/a_{qj} \quad (a_{qj}>0) \tag{9}$$

이 되도록 선택하면 된다. 만일 $\min_q b_q/cl_{qj}$ $(a_{qj}>0)$

가 두개 以上 存在하면 b_q^*가 0이 된다. 이것을 退化라 한다. 退化가 이르다면 같은 것을 되풀이하게 되고 最適解가 구해지지 않는다. 만일 退化가 일어나지 않으면 $m+n\ C_m$회 정도의 반복 뒤에 다음 두가지 중에 어느 한쪽 狀態로 된다.

(1) $c_j'\leqq 0$ $c_j=1, 2, \cdots, m+n)$

(2) $c_i'>0$이나 $a_{qj}>0$이 되는 q가 存在 안한다. (10)

(1)의 경우는 이때의 基底解가 最適解가 되고 b_0의 값이 Z의 最小値이다. (2)의 경우는 Z의 값은 얼마든지 작게할 수 있다. 現在는 $b_i>0$로 했기 때문에 實行可能解가 쉽게 구해졌지만 다른 경우에는 여러가지 방법이 있으나 그 段階法을 說明한다. 식 (7)의 右邊의 $b_i\leqq 0$ 대신에 不等解를 변화시키면 문제는 다음과 같이 된다.

Constraints

$$\left.\begin{aligned} &a_{11}x_1+a_{12}x_2+\cdots\cdots+a_{1n}x_n\leqq b_1 \\ &\vdots \\ &a_{i1}x_1+a_{i2}x_2+\cdots\cdots+a_{in}x_n\leqq b_i \\ &a_{i+1,1}x_1+a_{i+1,2}x_2+\cdots a_{i+1,n}x_n\geqq b_{i+1} \\ &\vdots \qquad\qquad (b_1>0, b_2>0, \cdots b_m>0) \\ &a_{j1}x_1+a_{i2}x_2\cdots\cdots a_{n}x_n\geqq b_j \\ &a_{j+1,1}x_1+a_{j+1,2}x_2+\cdots+a_{j+1,n}x_n=b_{j+1} \\ &\vdots \\ &a_{m,1}x_1+a_{m,2}x_2+\cdots\cdots+a_{m,n}x_n=b_m \\ &x_1\geqq 0, \ x_2\geqq 0, \cdots\cdots, \ x_m\geqq 0 \end{aligned}\right\} \tag{11}$$

objective

$$z=c_1x_1+cx_2+\cdots+c_nx_n \tag{12}$$

을 最小라 한다.

앞과 같이 slock variable을 도입하고 Artificial variable하는 변수를 $=0$ 및 $\geqq 0$의 式에 加하면

(11)은 다음과 같이 된다.

Constraints

$$\left.\begin{aligned} &a_{1,1}x_1+a_{1,2}x_2+\cdots\cdots+a_{1,n}x_n+x_{n+1} &&=b_1 \\ &\vdots \\ &a_{i,1}x_1+a_{i,2}x_2+\cdots\cdots+a_{i,n}x_n+x_{n+i} &&=b_i \\ &a_{i+1,1}x_1+a_{i+1,2}x_2+\cdots+a_{i+1,n}x_n-x_{n+i+1} &&=b_{i+1} \\ &\vdots \\ &a_{j,1}x_1+a_{j2}x_2+\cdots\cdots+a_{j,n}x_n-x_{n+j}+x_{n+2j-i} &&=b_j \\ &a_{j+1,1}x_1+a_{j+1,2}x_2+\cdots+a_{j+1,n}x_n+x_{n+2j-i+1} &&=b_{j+1} \\ &\vdots \\ &a_{m,1}x_1+a_{m,2}x_2+\cdots+a_{m,n}x_n \quad +x_{n+m+j-i} &&=b_m \\ &x_1\geqq 0, \cdots\cdots\cdots\cdots, x_{n+m+j-i}\geqq 0 \end{aligned}\right\} \tag{13}$$

(13)下에서

$$z'=x_{n+j1}+\cdots+x_{n+m+j-i}\text{를}$$

최소로하는 문제를 풀어 $z'=0$인 解가 存在하면 이解를 最初의 實行可能基底解로 한다. 다음에 (13)에서 Artificial variable를 뺀式에서 元來의 목적함수를 최소로 하면 된다.

(2) 非光形計劃法

실제의 設計에서는 非線形系가 많다. 비선형계의 最適點의 條件을 준것이 Kuhn-Jucker의 定理이고 이 定理는 다음과 같다.

制約條件

$$g_i(x)\leqq 0, \quad i=1, 2, \cdots, m$$
$$x=(x_1, x_2, \cdots, \cdots x_n)^T \tag{14}$$

下에서 目的函數

$$z=f(x)\rightarrow\min \tag{15}$$

의 問題에서 $g_i(x)$, $f(x)$가 convex 函數이고 $\phi(x, \lambda)$를

$$\phi(x, \lambda)=f(x)+\sum_{i=1}^{m}\lambda_i g_i(x) \tag{16}$$

로 정의할 때 x_0가 最適點이기 위한 必要充分條件은 任意의 x, $(\lambda)0$에 대하여 λ_0가 存在하고

$$\phi(x_0, \lambda)\leqq\phi\ (x_0\lambda_0)\leqq\phi_0\ (x_1\ \lambda) \tag{17}$$

를 滿足하는 것이다. λ_i: Lagrange 乘數 (simplex乘數)

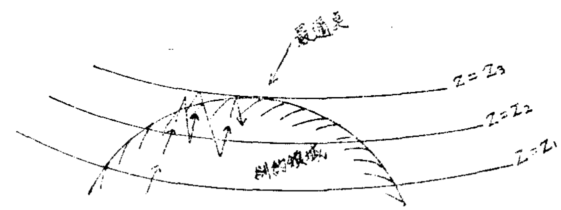

그림 2. Direct Differential Gradient Method

이 kuhn-Tucker의 定理를 利用하여 x는 $\phi(x,\lambda)$가 減小하도록 λ는 ϕ가 增加하도록 x와 λ를 逐次移行시켜 kuhn-Tucker의 必要充分條件을 滿足하는 x_0와 λ_0에 接近하는 傾斜法을 開發하였다.

즉, (14)의 制約領域內의 1點에서 出發하여 最大傾斜方向式 (18)의 方向으로 進行하여 最適點에 도달하는 方法이다.

$$\frac{dx}{dt}=-\nabla f(x)\equiv\left(-\frac{\partial f}{\partial x},\ -\frac{\partial f}{\partial x_2},\cdots,-\frac{\partial f}{\partial \lambda_n}\right) \quad (18)$$

이 方法은 Algorism 간단하나 최적판단이 힘들기 때문에 특수문제 외에는 사용되지 않는다.

그 후에 항상 制約條件의 接平面에서 안쪽으로 進行方向을 취하면서 目的 함수를 改善하는 Method of fensible direction을 개발하였다.

이 方法은 制約領域의 境界上의 한 點 A에서 式(19)의 方向으로 進行하면서 最適點에 到達하는 方法이다.

$$\left.\begin{array}{ll}(\Delta g_i)(x)\cdot\gamma)+\sigma\leqq 0 & \gamma=\text{進行方向}\\ (\Delta f\ (x)\cdot\gamma)\ \sigma\leqq\sigma & \\ |\gamma|\leqq\sigma & \sigma=\text{正의 임의 定數}\end{array}\right\} \quad (19)$$

$z=\sigma\rightarrow$最大

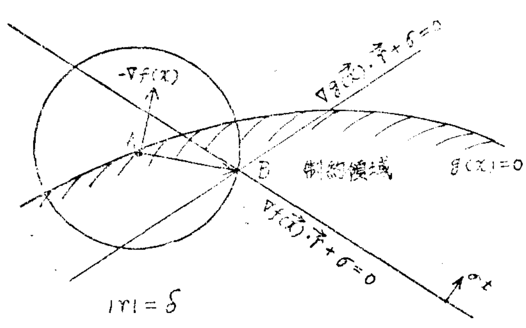

그림 3. Method of Feasible Direction

이 方法은 最適點까지의 step數는 減小되나 한 step의 計算量이 많다. 혹은 Rosen에 의해서 목적 함수의 最大傾斜를 제약 조건의 接平面에 投影한 方向으로 進行시키는 Gradient projection method등이 開發되었다. 또 制約條件이 없는 問題는 쉽게 풀수있는 것에 착안하여 제약 조건식을 목적 함수 속에 넣어 制約條件이 없는 問題로 바꾸어 푸는 方法등을 提示하였다. 이와같이 非線形計劃問題에는 많은 毛法이 있고 問題에 따라서 適當한 毛法을 써서 써야 한다.

DP法은 强力하고 有効한 方法이다 時間的으로 展開하는 System 또는 cascade system에 만 適用된다.

AD system에서는 다음과 같은 手順으로 한다.

(1) 從來의 旗製品圖面을 整理하고 나아가서 未來의 發展도 考慮하여 몇개의 標準 pattern으로 정리한다.

(2) spec에 따라서 最適 標準 pattern을 선택하는 檢索 system을 만든다.

(3) 標準 pattern과 더불어 그 製圖 加工 layout 및 기타의 何屬情報를 computer llbrary에 기억시키고 必要에 따라서 出力할 수 있도록 한다.

(4) 受法때 標準 pattern으로서 등록된 것 가운데서 선택하여 spec를 받도록 한다.

以上에 대해서 實際例로서 說明한다. 그림 1은 大型 motor의 自動設計, 自動製圖 system이다. 設計者는 spec를 받어 入力 Format에 맞추어 code化하고 computer에 入力한다. computer側에는 30種의 標準構造 pattern이 準備되고 이가운데 몇개의 部品은 option으로 선택가능하다. option은 computer側에서 最適인것을 선택하는 것이 原則이나 設計者가 指定하는 경우도

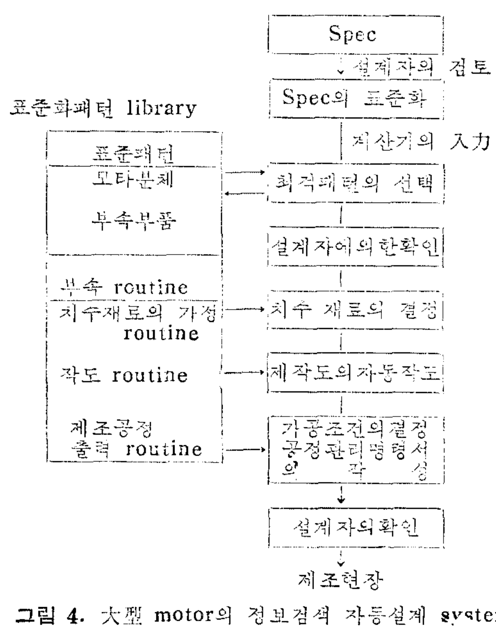

그림 4. 大型 motor의 정보검색 자동설계 system

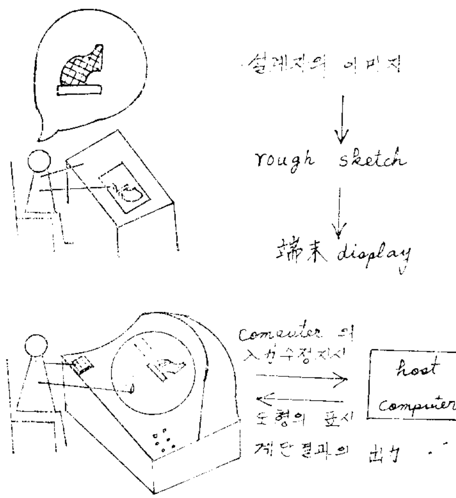

그림 5. 시행 process 기계화

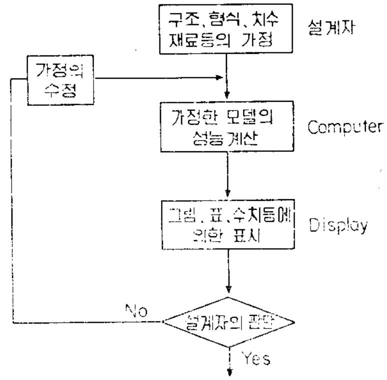

그림 6. Cut and Try

많고 얻어진 pattern은 LP로 하여 設計者가 確認한다. 여기서 말하는 被선택 pattern은 形式의 뜻이고 크기를 포함하지 않는다. 즉, Library中의 標準 pattern은 치수上의 伸縮이 可能한 상태로 memo되어 있다.

따라서 形式을 確認한 後에 spee에 適應하도록 치수나 材料가 自動으로 定해진다. 同時에 그 結果는 自動作動되고 必要에 따라서 加工條件이나 工程表등이 出力된다. 이 system은 具體的으로 設計가 어느 程度 完了되고 規格化되어 있는 것에 効果的이다. 이 system은 moter, 發電機 Turbine Trans등에서 各種 電子回路나 板金部品에 computer化되어 있다.

그러나 이 system에는 다음과 같은 問題點이 있다.

(1) 새로운 設計對象이나 標準 pattern의 修正을 要할때는 適用안된다.

(2) 이 system을 大規模로 構成하려고 하면 全製品에 設計의 標準化를 實施해야하고 設計者가 받아드리기 힘들게 된다.

(3) 이 system은 製造設備와 關聯되어 있고 變更이나 追加가 쉽지 않다.

(4) 人間에 쉬운 舊圖面의 修正이 computer에는 힘들때가 많다.

3. 試行形 自動設計 System (CAD)

前節의 system에서는 對處할 수 없는 새로운 設計問題 혹은 舊圖面으로 할 수 있는 設計라도 修正을 要때는 computer에만 맡기지 않고 設計者와의 共同作業에 의해서 機械化하이야 한다.

이 方法에 cut and Try 設計法이 있고 그림 6에 그 模型圖를 나타내었다. cut and Try의 過程은 display 장치에 의해 設計者와 computer 사이의 過程은 機械化한 것이다. Block diagram으로 쓰면 그림 6의 process가 반복된다. 이 system은 많은 努力이 있음에도 不拘하고 아직 自動化는 完成이 안되어 있다. 이 system의 困難한 點은 다음과 같다.

(1) computer는 원래 새로운 設計에 對應할 수 있는 Routine을 準備할 수 없다.

(2) 이 system에는 display (CRT) 혹은 自動作圖機가 必要한데 그 Hard ware soft ware에 많은 問題點이 있고 特히 圖形處理 soft ware가 빈약하다.

4. 自動設計 system의 評價基準

system의 良否는 使用目的이나 使用환경에 의해서 영향을 받기 때문에 一般論은 힘들지만 생각하는데 必要한 基準은 다음과 같다.

1. 能力 : system의 規模에 대한 汎用性이 比較規準이 된다.

2. 效率 : 같은 出力을 얻는데 必要한 computer 規模와 計算時間으로 比較한다.

3. 使用의 용이성 : 연습기관과 使用期間에 專門 programmer가 必要한지 안한지가 評價에 도움이 된다.

4. 追加 削除 變更 修正의 용이성 : 自動設計 system에서 매우 重要하고 system에 이와같은 機能을 가지게 하는것은 不可能하다.

5. 擴張性 : system을 계획할때 이것이 고려되어 있나 그리고 擴張때 Document을 다시 쓰지 않고 追加에 의해서 實施하는 것이 바람직하다. 特히 周邊 Routine은 Document의 規制를 받지않고 마음대로 擴大할수 있는 것여 좋다.

6. 他 system과의結合 : 實際上 困難하나 예컨데 APT와 FORTRAN의 共用과 같이 그 必要性이 있다.

5. 結 論

前述한 두가지 system은 모두 會話形으로 構成해야 하고 試行形의 경우는 勿論이며 AD system을 會話形으로 構成하면 設計者의 감시가 잘 이루어지고 必要에 따라서 設計者가 設計 諸元을 줄 수 있다. 會話形式에는 program을 좀더 soft하게 하이 Man-Machine system을 開發하여야 한다. 이것에 의해 CAD system은 開發될 것이다. 앞으로는 時系列的 處理에서 空間的 處式方式으로 變換시키고 設計者가 program할때 自由롭게 算式의 一部를 變更하거나 評價判定의 規準을 修正할 必要가 있고 Data의 構造化와 program의 管理化와 더불어 人間의 意志決定을 제 1順位로 하여 進行할 것이 要望된다.

參 考 文 獻

(1) 沖野教郎, 機械設計, 17-8 (1973), 8
(2) 中島尙正, 自動設計 (1971), 丸善.
(3) 寺野壽郎, 機械 system 設計 (1971), 丸善.