

# 설계 인자와 설계 목표를 이용한 진화 설계 및 재설계

이강수\* , 이건우\*\*

## Evolutionary Design and Re-design Using Design Parameters and Goals

Kang-Soo Lee\* and Kunwoo Lee\*\*

### ABSTRACT

Design parameters and goals play important roles in design. Design goals are the required functions of the design elements and explicitly expressed by design parameters. Design parameters also indicate the relations among design elements, by which constraint networks can be constructed and some useful information can be induced. In this study, the mechanical design process is assumed to be the assignment of design goals and their realization through the evolutionary refinement of the design parameters. Thus an integrated design system is proposed to support the process of assigning the design goals and refining the values of the design parameters. In the design system, a genetic engine that utilizes a genetic algorithm is installed to simulate an iterative design process, which leads to an evolutionary design. The genetic engine treats design parameters as genes and design goals as evaluation function. Re-design and design modification are facilitated by the design parameters. The re-design can be activated in the design system by using the information stored in the design parameters when design parameters or goals are changed.

**Key Words :** Design parameters (설계 인자), Design goals (설계 목표), Evolutionary design (진화 설계), Re-design (재설계), Genetic algorithm (유전자 알고리즘)

### 1. 서론

기계 제품을 설계 하고자 할 때 설계 목표를 먼저 설정하고 제품을 구성하고 있는 기계 요소와 이러한 요소를 설계하는 과정은 주어진 설계 목표를 실현 하기 위한 것이라 할 수 있다. 공리 설계(axiomatic design)에서 FR<sup>(1)</sup>, 다구찌 방법에서 response<sup>(2)</sup>, QFD 에서 요구 품질<sup>(3)</sup> 등으로 불리는 설계 목표는 이와 같이 설계 과정에서 중요한 역할을 한다.

설계 인자는 설계 요소를 구현하기 위한 변수로 설계 인자를 구체화함으로써 설계 요소가 구체

화 된다. 설계 인자도 설계 목표와 마찬가지로 설계 과정에서 중요하며 공리 설계에서 DP<sup>(1)</sup>, 다구찌 방법에서 factor<sup>(2)</sup>, QFD 에서 품질 요소<sup>(3)</sup> 등이 설계 인자와 유사한 역할을 한다. 위와 같이 설계 목표와 설계 인자는 설계 과정에서 중요한 역할을 하며 본 논문에서는 이를 이용하여 기계 설계를 진행하는 과정, 특히 재설계 및 진화 설계를 지원하는 방법을 제안하고자 한다.

설계 과정에서 관련 정보를 저장하고 이를 이용하는 방식의 설계 시스템<sup>(4,5)</sup>에서 취급하는 설계 정보 중 설계 인자와 설계 목표에 대한 인자를 이용하는 연구로 Gero<sup>(6,7)</sup>의 연구를 들 수 있다.

\* 대우중공업(주) 중앙연구소

\*\* 서울대학교 정밀기계설계공동연구소

Gero<sup>(6,7)</sup>는 유전자 알고리즘을 이용한 진화 설계 방식을 제안하였으며 이 방법은 유전자를 재구성할 때 진화 유전자를 선택해 설계 목표에 빨리 도달하기 위한 것이다.

기계 설계 과정에서는 형상 정보, 설계 인자, 설계 문서, 공학 해석 정보, 설계 요소의 기능(function), 기동(behavior), 상관 관계(relation), 설계 진행 정보, 설계 목표 등과 같은 다양한 설계 정보가 사용된다<sup>(4)</sup>. 본 논문에서는 이러한 설계 정보 중 가장 일반적으로 고려될 수 있는 설계 인자와 설계 목표를 이용하여 진행되는 설계 과정을 정의하고 또한 설계 이력을 이용하여 설계 반복을 쉽게 할 수 있도록 재설계 과정(re-design process)을 제시한다. 기계 설계 과정에서 주어진 제약 조건을 최적으로 만족하는 설계 인자의 값을 구하는 과정이 반복될수록 좋은 설계가 나오며 본 논문에서는 이러한 설계 과정이 생물이 세대를 거듭할수록 진화하는 개념과 유사하다고 생각되어 진화 설계란 개념을 사용하였으며, 이러한 개념에 가장 잘 부합하는 최적화 방법인 유전자 알고리즘을 이용하였다. 초기 설계에서는 적은 개수의 설계 인자가 설정되고 상세 설계로 갈수록 설계 인자가 많이 설정되는 경향이 있으며 또한 설계 인자에 관련된 제약 조건도 경우에 따라 필요한 것보다 적게 또는 많이 설정되기도 한다. 이러한 설계 과정에 유전자 알고리즘을 사용하면 설정된 설계 인자와 제약 조건에 맞게 적절한 결과를 얻을 수 있다. 또한 설계 인자를 유전자로 모델링하면 설계 인자의 값 변화와 설계의 진화 관계를 관찰할 수 있다. 본 연구에서는 최적화 설계를 위해 유전자 알고리즘을 사용하였으며 설계자가 부여한 설계 인자를 유전자로 설정하고 설계 목표를 평가 함수로 설정하여 설계 목표를 만족하는 최적의 인자 값을 구하였다. 이러한 방법을 진화 설계 방법이라고 하였다.

한편, 설계 목표 또는 설계 인자가 변할 경우 재설계를 하게 된다. 본 연구에서는 설계 인자 상호 간에 구축되어 있는 제약 조건을 탐색하여 재설계 계획을 효과적으로 구축할 수 있게 하였다.

본문에서는 본 연구에서 개발한 설계 시스템에서 설계 인자와 설계 목표를 설정하는 방법을 설명하고, 설계 시스템에서 진화 설계 하는 과정과 재설계하는 과정에 대해 설명하겠다. 그리고 대표적인 기계 설계 문제인 감속기 설계를 예제로

개발된 설계 시스템을 이용하여 설계하는 방법을 보이고 제안한 설계 방법의 유용성을 보여준다.

## 2. 설계 인자와 설계 목표 표현

본 연구에서 설계 데이터가 되는 설계 인자와 설계 목표를 표현하는 방법은 다음과 같다.

### 2.1 설계 요소 자료 구조

본 연구에서 설계 요소를 나타내기 위해서 요소 객체<sup>(4)</sup>라는 표현 방법을 사용하였으며 이 표현 방법은 설계 과정에서 사용되는 다양한 설계 정보를 수용하게 한다. 이 요소 객체는 Fig. 1 과 같이 설계 인자와 설계 목표, 설계 과정을 표현할 수 있도록 간단히 나타내었다<sup>(8,9)</sup>.

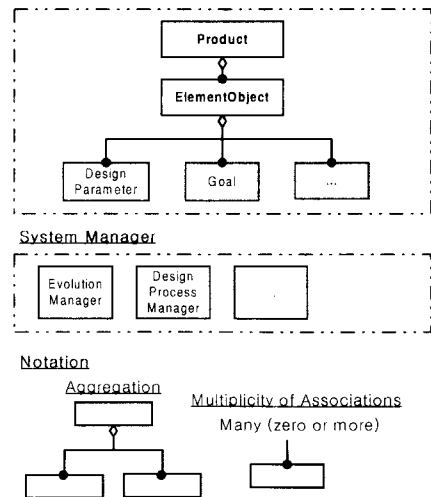


Fig. 1 Simple representation of element object

요소 객체는 설계하려는 제품을 구성하는 설계 요소를 나타내기 위해 객체 지향 방식으로 정의된 것으로 Fig. 1 에서 *ElementObject* 로 나타나 있다. *ElementObject* 는 설계 인자를 나타내는 *DesignParameter* 와 설계 목표를 나타내는 *Goal* 을 구성 객체로 가지고 있다. 원래의 요소 객체는 이외에도 다양한 정보를 다루기 위한 구조<sup>(4)</sup>이나 여기에서는 설계 인자와 설계 목표만 고려하였다. 그리고 설계 시스템에서 전역 객체로 진화 설계를 담당하는 *EvolutionManager* 와 설계 과정을 담당하는 *DesignProcessManager* 가 있다.

## 2.2 설계 인자 설정

본 연구에서 사용되는 설계 인자의 데이터는 Fig. 1의 *DesignParameter* 에 저장되며 다음과 같은 종류의 설계 인자를 사용할 수 있다. 이러한 종류의 데이터로 대부분의 공학 설계 문제를 다룰 수 있다.

- 불리안 (Boolean)
- 정수, 정수 범위, 등차 배열인 정수 범위, 정수 배열 (integer, integer region, integer interval, integer array)
- 실수, 실수 범위, 등차 배열인 실수 범위, 실수 배열 (real, real region, real interval, real array)
- 문자, 문자 배열 (character, character array)
- 문자열, 문자열 배열 (string, string array)

Table 1 에는 *DesignParameter* 클래스에서 설계 인자를 사용하는 예가 나타나 있다. *DesignParameter* 클래스에는 Table 1 의 첫번째 열에 나타난 바와 같이 *Parameter*, *Type*, *Value*, *NoOfArrayComp*, *Array*, *LowerValue*, *UpperValue*, *Increment*, *DataChange*, *Description*, *InducedEO*, *ParameterOfInducedEO* 와 같은 속성이 있다.

*Parameter* 는 설계 인자의 이름이며 *Type* 은 설계 인자의 데이터 종류, *Value* 는 설계 인자에 설정된 값이다. *NoOfArrayComp* 는 설계 인자의 종류가 배열인 경우 구성 원소의 개수이며 *Array* 는 이때 설정된 배열 값을 나타낸다. *LowerValue* 와 *UpperValue* 는 설계 인자가 범위(range, interval)로 주어질 때 주어진 범위의 최소값과 최대값이며, *Increment* 는 설계 인자가 등차 배열인 범위(interval)로 주어질 때 증가분을 나타낸다. *DataChange* 은 설계 인자의 값이 계산 도중 변경될 수 있는지 여부를 나타내는 것으로 그 값이 F 일 때는 설계 과정 중에 변경 될 수 없는 설계 인자, 즉, 입력 인자가 되는 것을 의미하며, T 일 때에는 변경될 수 있는 설계 인자, 즉, 설계 과정에서 정해야 할 인자를 의미한다. *Description* 은 설계 인자에 대한 설명이다. *InducedEO* 는 이 설계 인자가 어떤 요소 객체의 설계 인자에서 유도되는 지를 나타내며, *ParameterOfInducedEO* 는 파생된 요소 객체에서의 *Parameter*, 즉 그 곳에서의 이름을 의미한다.

설계 인자의 데이터 종류에 따라 *DesignParameter* 의 속성 중 필요한 것만 사용된다. Table 1 에서 데이터 종류에 따라 사용되지 않는 속성은

별표(\*)로 표시하였다. 예를 들어 Table 1 의 *a* 는 정수 범위의 설계 인자이며, 이를 나타내기 위해 *DesignParameter* 의 속성 중 *Parameter*, *Type*, *Value*, *LowerValue*, *UpperValue*, *DataChange*, *Description*, *InducedEO*, *ParameterOfInducedEO* 는 사용되며, *NoOfArrayComp*, *Array*, *Increment* 는 사용되지 않는다.

## 2.3 설계 목표 설정과 평가

설정된 설계 인자를 이용하여 설계 목표를 설정한다. 설계 목표에 대한 정보는 *Goal* 클래스에 저장되며 그 속성으로는 Table 2 에서와 같이 *GoalName*, *ObjectiveFn*, *TargetValue*, *Weight*, *Result* 가 있다. *GoalName* 은 설계 목표의 이름이고, *ObjectiveFn* 은 설계 인자로 표현된 목적 함수이다. *TargetValue* 는 목적 함수의 목표값이며, *Weight* 는 가중치, *Result* 는 설계 인자의 값을 결정한 후 결정되는 목적 함수의 계산 결과이다.

일반적으로 설계 목표는 다목적 함수로 표현되며 Table 2 에 나타난 *Goal* 은 n 개의 목적 함수를 가지고 있다.

Table 1 Examples of *DesignParameter*

Attributes	<i>DesignParameter</i>				
<i>Parameter</i>	<i>a</i>	<i>b</i>	<i>m</i>	<i>i</i>	<i>l</i>
<i>Type</i>	integer region	Integer Region	real	real	real
<i>Value</i>	(21)	(74)	5	3.5	237.5
<i>NoOfArrayComp</i>	*	*	*	*	*
<i>Array</i>	*	*	*	*	*
<i>LowerValue</i>	15	21	*	*	*
<i>UpperValue</i>	30	74	*	*	*
<i>Increment</i>	*	*	*	*	*
<i>DataChange</i>	T	T	F	F	F
<i>Description</i>	no. of teeth	no. of teeth	module	ratio	offset
<i>InducedEO</i>	<i>Input-Gear</i>	<i>Output-Gear</i>	<i>Input-Gear</i>	<i>Requirement</i>	<i>Requirement</i>
<i>ParameterOfInducedEO</i>	<i>Teeth</i>	<i>Teeth</i>	<i>Module</i>	<i>ratio</i>	<i>offset</i>

Note:

1. The asterisk(\*) means that data for the position is meaningless.
2. The value in parenthesis may be changed during design process.
3. F at the *DataChange* column means that the value of the parameter cannot be changed during design process, or, the parameter is input data of the design. T means that the value can be changed during design process.

Table 2 Attributes of Goal

GoalName	ObjectiveFn	Target Value	Weight	Result
GOL <sub>0</sub>	f <sub>0</sub>	T <sub>0</sub>	w <sub>0</sub>	R <sub>0</sub>
GOL <sub>1</sub>	f <sub>1</sub>	T <sub>1</sub>	w <sub>1</sub>	R <sub>1</sub>
...	...	...	...	...
GOL <sub>n-1</sub>	f <sub>n-1</sub>	T <sub>n-1</sub>	w <sub>n-1</sub>	R <sub>n-1</sub>

Michalewics<sup>(10)</sup>는 유전자 알고리즘으로 다목적 함수를 풀기 위해 여러 가지 방법을 제시하였는데 이 중 각 목적 함수마다 목표값을 지정하고 푸는 방법이 기계설계 문제에 잘 적용되며 본 연구에서는 목적 함수의 목표값과 가중치를 고려하여 평가하는 방식으로 식 (1)<sup>(9,10)</sup>을 사용 하였다.

$$F(x) = \left( \sum_{i=0}^{n-1} w_i \times |f_i(x) - T_i|^r \right)^{\frac{1}{r}} \quad (1)$$

여기에서,

- r = 일반적으로 2 (유클리디안 거리)
- x = 주어진 변수 값
- F(x) = x 가 주어졌을 때 평가값
- w<sub>i</sub> = i 번째 목적 함수의 가중치
- f<sub>i</sub>(x) = x 가 주어졌을 때 i 번째 목적 함수 값
- T<sub>i</sub> = i 번째 목적 함수의 목표값

### 2.4 요소 객체 사이의 제약 조건 구축

설계 인자를 설정하는 *DesignParameter* 클래스를 통해 요소 객체 간의 제약 조건이 구축된다. *DesignParameter* 클래스에는 Table 1의 *InducedEO*와 *ParamOfInducedEO*는 설계 인자가 어느 요소 객체의 설계 인자에서 파생 되었는지를 나타내며 이 속성을 추적 하면 요소 객체 간의 제약 조건 네트워크가 구축된다. Fig. 2의 감속기의 예를 살펴 보자. Fig. 2에는 *Reducer*, *Input-Gear*, *Output-Gear*, *Requirement* 네 개의 요소 객체가 있으며 *Reducer*에는 a, b, m, i, l 다섯 개의 설계 인자가 있다. a와 m은 각각 *Input-Gear*의 *Teeth*와 *Module*에서, b는 *Output-Gear*의 *Teeth*에서, i와 l은 각각 *Requirement*의 *ratio*와 *offset*에서 파생된 설계 인자이다. 이렇게 설정이 되면 Fig. 2의 실선 화살표처럼 제약 조건 네트워크가 성립되고 앞에서 설명한 방법으로 요소 객체 *Reducer*의 설계 목표를 풀면 설계 인자 a, b의 값이 결정된다. 이렇게 결정된 값들은 요소 객체 간에 구축된 제약 조건 네트워크로 전파되며 점선 화살표가 이를 나타낸다.

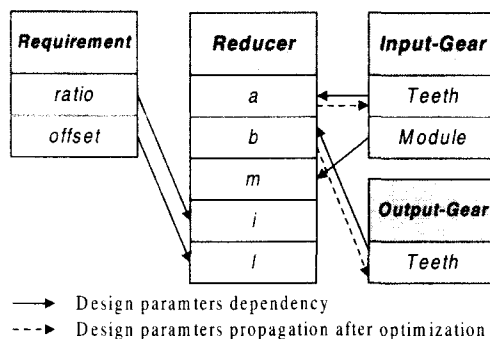


Fig. 2 Constraint networks among element objects and their propagation

즉, a의 값은 *Input-Gear*의 *Teeth*로, b의 값은 *Output-Gear*의 *Teeth*로 전파된다.

## 3. 진화 설계 방법(Evolutionary design)

### 3.1 진화 설계 알고리즘

본 연구에서 사용한 요소 객체를 이용한 진화 설계 방법은 다음과 같다. 설계 인자를 2.2 절에서 설명한 바와 같이 요소 객체의 구성 클래스 중 *DesignParameter*를 통해 설정하고, 목적 함수는 *Goal*을 통해 설정한다. 이 때 설계 인자를 다양한 형태로 사용하고, 목적 함수에서는 설계 인자의 값이 설정되지 않은 상태로 문자열을 이용하여 표시하기 때문에 이를 처리하기 위해 Fig. 3과 같이 유전자 엔진(genetic engine)을 이용하는 알고리즘을 사용하였다.

- (1) 목적 함수를 구축한다. *Goal* 클래스에 설정된 목적 함수를 찾는다. 목적 함수는 2.2 절에 설명된 바와 같이 *Goal* 클래스의 속성으로 설정되어 있다.
- (2) 목적 함수에 사용된 설계 인자를 찾는다. *DesignParameter*에서 정의된 설계 인자 중에서 목적 함수를 정의하는 데 사용된 설계 인자를 찾는다. 이를 위해 먼저 *DesignParameter*에 정의된 모든 설계 인자를 나타내는 문자열(2.2 절에서 *Parameter* 속성) 리스트를 구축한다. 목적 함수도 문자열로 되어 있으므로 목적 함수와 설계 인자를 비교하여, 목적 함수를 정의하는 데 사용된 설계 인자를 찾아 낸다.
- (3) 설계 인자를 유전자 알고리즘에 적용하기

위해 엄격체로 표현한다. Fig. 3의 Genetic-Algorithm-Mechanism에서는 GALib<sup>(11)</sup>를 이용하였고 여기에서는 정수(integer, integer region, integer interval, integer array)와 실수(real, real region, real interval, real array) 데이터를 사용할 수 있다. 따라서 (2) 과정에서 찾은 설계 인자를 GALib에서 정의한 엄격체 형태로 표현 한다.

(4) 유전자 알고리즘을 적용 한다. 설정된 목표 함수와 설계 인자를 이용하여 유전자 알고리즘을 적용한다. 목적 함수가 설계 인자를 값이 아닌 문자 형태로 갖고 있으므로, 목적 함수를 평가할 때 이를 처리 해 주어야 한다. 한 편 개체 집단 개수, 계산 세대 수, 교배 확률, 돌연 변이 확률 등 환경 변수는 설계 시스템에서 기본값을 설정하고 목적 함수의 특성에 맞게 설계자가 변경할 수 있게 하였다.

(5) 유전자 알고리즘을 적용할 때 목적 함수는 다음과 같이 계산한다. 문자열을 입력으로 받아 계산을 수행할 때 파서(Fig. 3에서 Parser)를 이용하고 이 파서는 숫자와 연산, 함수를 나타내는 문자열 만을 취급할 수 있다. 그러나 요소 객체에서

사용하는 목적 함수는 설계 인자를 문자 형태로 저장하고 있으므로, 문자로 표시된 설계 인자를 Genetic-Algorithm-Mechanism에서 제공하는 실제의 값으로 대체하여 사용해야 한다. 이 과정이 Fig. 3에 나타나 있다. 예를 들면 다음과 같이 된다. 먼저, 설계 인자가 문자로 표현된 목적 함수(b/a)를 구한다. 여기에 사용된 설계 인자(a,b)에 현재 설정된 값을 Genetic-Algorithm-Mechanism에서 받아 (a=21, b=74), 목적 함수에 대체(74/21) 시킨다. 이를 파서에 넘기면 파서는 이를 계산해 값(3.5238)을 계산하여 이 값을 유전자 알고리즘에 사용할 수 있게 한다.

### 3.2 설계 과정에서 제약 조건의 충돌

설계 과정에서 발생하는 제약 조건의 충돌 (constraint conflict)을 다음과 같이 처리<sup>(12)</sup>한다.

#### (1) 제약 조건의 충돌 검사

다음과 같은 경우에 충돌이 생긴 것이다.

- 3.1 절에서와 같이 요소 객체에 설정되어 있는 설계 목표를 풀어 설계 인자의 값을 계산한다. 여기에서 결정된 설계 인자의 값이 설정된 값의 범위와 다르면 충돌이 발생한 것으로 취급한다. 이 경우는 설계 인자의 데이터 종류가 region, interval, array 일 경우에 해당된다.

- 새로 결정된 설계 인자의 값이 주어진 범위를 벗어날 때와 진화 설계에서 계산한 목적 함수의 값과 목표값(Target Value)과의 차이가 시스템에 설정되어 있는 허용 오차보다 큰 경우에 제약 조건의 충돌이 생긴 것으로 취급한다.

#### (2) 충돌 발생 시 조치

제약 조건과 충돌이 발생하면 그 내용을 기록하고 설계 시스템 화면 상에 에러 메시지를 보내 설계자에게 알린다.

#### (3) 충돌 해결

설계자는 화면상의 에러 메시지 또는 충돌 기록을 통해 제약 조건의 충돌이 발생했음을 알게 되고 설계 인자나 환경 변수를 조정함으로써 충돌을 해결 한다.

#### (4) 기록

설계자는 요소 객체에 충돌 발생과 그 해결 기록을 남겨 참고 자료가 되게 한다.

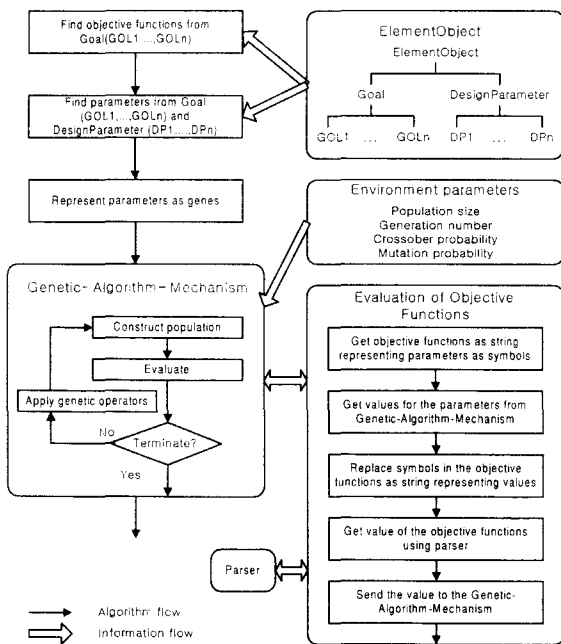


Fig. 3 Flow chart of genetic engine for evolutionary design

#### 4. 재설계

제안된 설계 시스템에서 재설계 과정<sup>(13)</sup>은 다음과 같이 진행된다.

##### 4.1 설계 이력 저장

설계 과정에서 요소 객체에는 설계 이력을 나타내는 설계 정보와 설계 진행 정보가 저장된다. 요소 객체에는 다양한 설계 정보가 저장되며 설계 진행 정보로는 요소 객체 생성/소멸 리스트, 진화 설계 리스트 정보 등이 저장된다. 여기에서 요소 객체의 생성, 소멸, 진화 설계 등이 설계 활동(design activity)이 되며 요소 객체에 저장된 정보의 상태가 설계 상태(design state)가 된다. 이렇게 설계 활동을 구분해 저장하면 설계가 구체화 되는 과정을 쉽게 파악할 수 있다.

##### 4.2 재설계 계획

본 연구에서는 재설계를 할 때 저장된 설계 이력대로 진행하지 않고 재설계 할 상황에 맞게 설계자가 저장된 정보를 이용하여 재설계 계획을 시스템과 대화식으로 세운다. 이 방법은 설계 이력을 그대로 따라가며 재설계하는 것보다 재설계될 요소 객체의 수가 적어 효율적이며 재설계 계획에 루프가 생기는 것을 방지할 수 있다. 재설계 과정은 다음과 같이 진행된다.

(1) 설계자가 재설계를 시작할 요소 객체와 설계 인자를 선정한다. 대부분의 경우는 설계자가 어떤 요소 객체에서 재설계를 해야 할 지 알고 있으며 필요에 따라 설계 이력 데이터를 참조하여 재설계를 시작할 요소 객체를 선정할 수 있다.

(2) 한 요소 객체가 재설계 되었을 때 이로 인해 다른 요소 객체가 재설계 되어야 한다. 이 때 시스템은 다음과 같은 요소 객체를 재설계 리스트로 저장 한다.

i) 변경된 설계 인자가 유도된 요소 객체, 즉, *InducedEO* 가 있는 설계 인자의 값이 변경되었을 경우, 이때의 *InducedEO* 는 재설계 대상이 된다.

ii) 변경된 설계 인자를 사용하는 요소 객체, 즉, 다른 요소 객체의 *InducedEO* 와 *ParameterOf-InducedEO* 가 변경된 설계 인자를 나타낼 경우 이 요소 객체는 재설계 대상이 된다.

이 외에도 재설계한 요소 객체와 설계 인자로

연관되지는 않았지만 관련 요소 객체<sup>(14)</sup>로 설정되었을 때에도 재설계 대상이 된다. 위의 기준으로 구한 재설계 대상 요소 객체 중 현재 재설계 리스트에 저장되어 있는 요소 객체는 중복 저장하지 않는다.

(3) 재설계 리스트에 저장된 요소 객체에서 아래의 기준으로 평가해 재설계 순서를 정하며 아래의 기준은 우선 순위대로 기술 되었다. 재설계할 요소 객체는 요소 객체의 이름과 계층 레벨, 변경 설계 인자 리스트를 저장하고 있다.

i) 두 요소 객체를 비교해 다른 요소 객체의 입력 인자가 되는 설계 인자를 가진 요소 객체를 먼저 재설계해야 한다. 이는 한 요소 객체가 재설계 되었을 때 변경된 설계 인자를 사용하는 다른 요소 객체는 재설계 되어야 하기 때문이다.

ii) 먼저 재설계 리스트에 저장된 요소 객체를 먼저 재설계한다. 이는 일반적으로 먼저 재설계 리스트에 저장된 요소 객체일수록 재설계를 할 때 다른 요소 객체에 영향을 많이 주기 때문이다.

iii) 설계 과정에서 계층 레벨이 높은 요소 객체가 앞선다. 설계 변수가 같은 상위 계층에서 추상화 되어 있고 하위로 갈수록 구체화 되어 있기 때문에 상위 계층의 설계 변수가 다른 설계 인자에 영향을 많이 준다. 그래서 상위 계층의 요소 객체를 먼저 재설계한다.

##### 4.3 재설계 실행

설계자는 시스템이 제시하는 순서를 참고하여 재설계를 한다. 대부분의 경우는 시스템이 제공하는 재설계 리스트와 순서를 따라 진행하면 되나 설계자가 임의로 조정할 수도 있다. 설계자가 하나의 요소 객체를 재설계할 때마다 4.2 절에서 설명한 바와 같이 새로 재설계 계획이 수립된다. 재설계 리스트에 저장된 모든 요소 객체를 재설계하면 재설계 과정이 끝난다.

재설계 과정에서 각 요소 객체는 다음과 같은 과정을 거친다.

(1) 다른 요소 객체에서 가져온 설계 인자를 확인 설정한다.

(2) 요소 객체에 저장된 설계 인자와 설계 목표 데이터를 이용해 진화 설계 한다.

재설계 과정에서 제약 조건의 충돌이 발생한

경우 설계자는 충돌을 해결하고 현재의 요소 객체 또는 처음부터 재설계를 진행한다.

### 5. 적용 예제

본 연구의 내용을 설계 시스템으로 구현 하였으며 설계 시스템에서는 설계 요소의 형상을 나타내는 형상 프로세서로 SDRC의 I-DEAS를 사용하였으며 요소 객체의 정보를 취급하는 정보 프로세서로 DHII (Design History, Information and Intent)를 개발하였다. 두 프로세서는 CORBA를 이용하여 데이터를 공유한다<sup>(4)</sup>. 개발된 설계 시스템을 전형적인 기계 설계 문제인 감속기 설계에 적용하여 설명한다.

#### 5.1 감속기 모델링

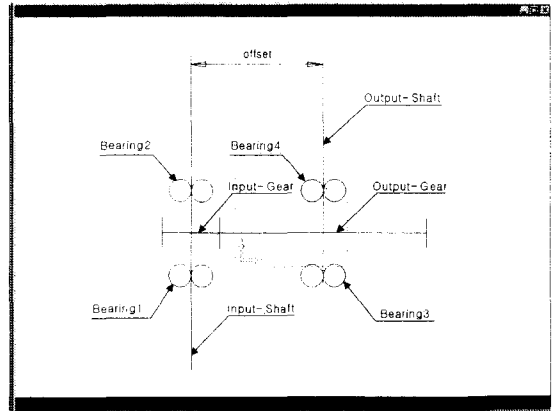
Fig. 4(a)는 형상 프로세서로 사용되는 I-DEAS에서 1 단 감속기를 와이어프레임 모델로 모델링한 것이고 Fig. 4(b)는 정보 프로세서인 DHII에서 설계할 요소 객체를 모델링한 것이다. Fig. 4(a)와 Fig. 4(b)는 하나의 화면에 두 개의 윈도우로 나타나며 사용자는 두 개의 윈도우를 자유자재로 옮겨가며 설계를 진행할 수 있다.

Fig. 4(b)에서는 최상위 요소로 Reducer가 있으며, 이의 하위 객체로 Input-Assy, Output-Assy, Housing, Requirement, Team이 있다. Input-Assy는 입력축계를 나타내는 것으로 하위 요소 객체로 Input-Shaft와 Input-Gear, Bearing1, Bearing2가 있다. Output-Assy는 출력축계를 나타내는 것으로 하위 요소 객체로 Output-Shaft와 Output-Gear, Bearing3, Bearing4가 있다. Requirement와 Team은 각각 설계 요구 조건과 설계 팀에 대한 정보를 저장하기 위한 요소 객체이다. 정보 프로세서인 DHII에서 요소 객체를 모델링하는 방법은 설계자가 정한다.

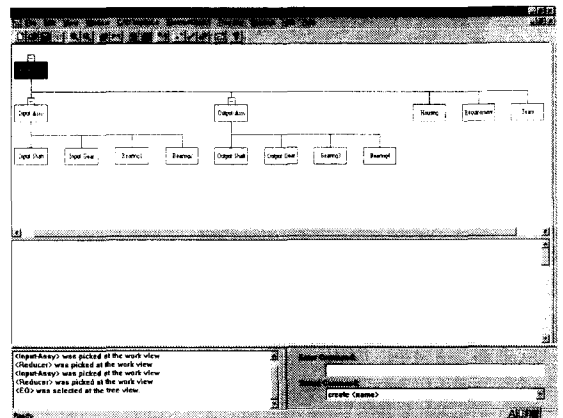
#### 5.2 설계 인자 설정

Fig. 5는 요소 객체 Requirement에 감속기의 설계 요구 조건으로 감속비(Fig. 5에서 ratio) 3.5, 입력축과 출력축 간의 거리(offset) 237.5 mm, 전달 동력(power) 10 kW, 입력 회전수(rpm) 500 rpm을 입력시킨다.

Fig. 6은 주어진 설계 요구 조건을 만족시키는 최상위 요소인 Reducer에서 기어비 설계를 하는 과정을 보여 준다. 여기에서 기어비 설계를 하기



(a)



(b)

Fig. 4 Modeling reducer both I-DEAS and DHII  
(a) Layout at I-DEAS (b). Modeling at DHII

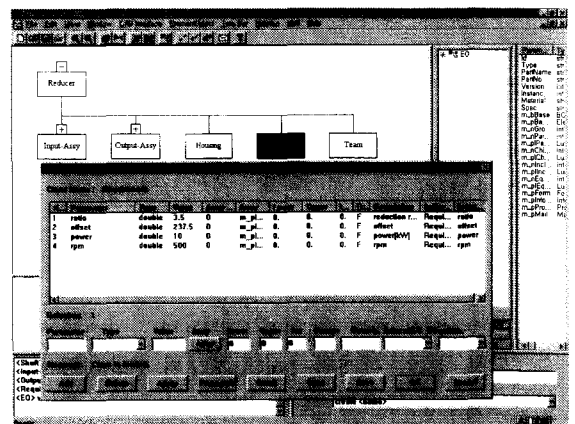


Fig. 5 Setting design requirement at Requirement

위한 요소 객체로 *Reducer* 를 선택하는 것은 설계자가 임의로 선택한 것이다. *Reducer* 에는 설계 인자로  $a, b, m, i, l$  이 설정되어 있다. 입력 인자로  $m, i, l$  이 있으며(DataChange 가 F), 출력 인자로  $a, b$  가 있음(DataChange 가 T) 알 수 있다.  $a$  는 *Input-Gear* 의 잇수로 [15,30] 사이의 정수이며  $m$  은 *Input-Gear* 의 모듈로로 5이며  $b$  는 *Output-Gear* 의 잇수로 [50,100] 사이의 정수이다. *Input-Gear* 와 *Output-Gear* 는 모듈로가 같으므로 어느 쪽에서  $m$  을 유도해도 되나 여기에서는 임의로 *Input-Gear* 에서 유도 하였다.  $i$  와  $l$  은 설계 요구 조건을 나타내는 *Requirement* 로부터 유도된 설계 인자이며  $i$  는 감속비로 3.5,  $l$  은 윗셋으로 237.5 mm 이다.

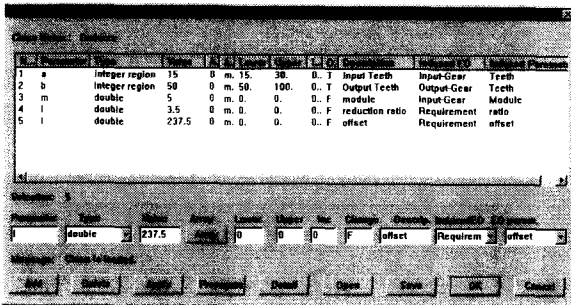


Fig. 6 Setting design parameters of *Reducer*

### 5.3 설계 목표 설정

Fig. 6에서 설정한 설계 인자를 사용하여 설계 요구 조건으로 주어진 감속비와 윗셋을 만족시키는 기어 잇수를 설계한다. 이 과정은 다음 식 (2), (3)을 만족 시켜야 한다.

$$i = \frac{b}{a} \tag{2}$$

$$l = \frac{m \times (a + b)}{2} \tag{3}$$

Fig. 7은 설계 목표를 설정하는 화면이다. 상단 부 컨트롤 리스트에는 식 (2)와 (3)을 목적 함수로 표현하였다. 중간 부분에 있는 컨트롤 리스트에는 목적 함수를 구축하는 데 사용된 설계 인자들이 나타나 있다. 하단부의 컨트롤 리스트에는 유전자 알고리즘에 사용되는 환경 변수인 세대수, 개체

집단 크기, 교배 확률, 돌연 변이 확률이 나타나 있다. Fig. 7에는 세대수가 500, 개체 집단 크기가 20, 교배 확률이 50%, 돌연 변이 확률이 10%로 나타나 있다. 하단의 탭 중 *Evolve* 를 누르면 최적화 계산이 수행되어 중간 부분의 설계 인자의 값이  $a$  는 21로  $b$  는 74로 결정된다. 이는 감속비가 3.5238로 목표값과 0.7% 정도의 차이가 있으며, 윗셋이 237.5가 되게 하는 값으로 좋은 결과를 보여 준다. 하단부의 탭 중 *Propagate* 를 누르면  $a$  와  $b$  에 계산된 값 21과 74이 설정되고  $a$  의 값은 *Input-Gear* 에,  $b$  의 값은 *Output-Gear* 로 전파된다. 이때 *Reducer* 는 *Input-Gear* 와 *Output-Gear* 에게 각각  $a, b$  가 변경되었다면 메일을 보낸다.

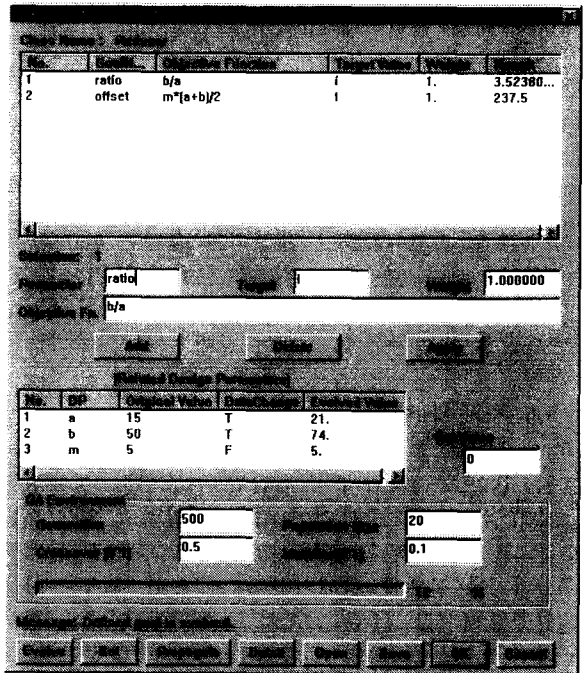


Fig. 7 Design goal constructed in *Reducer*

### 5.4 설계 구체화

Fig. 7에서 *Input-Gear* 와 *Output-Gear* 의 잇수가 결정되므로 기어의 강도 계산, 축의 강도 계산으로 설계를 구체화 시킨다. 기어의 강도 계산, 축의 강도 계산 기능은 각각의 요소 객체에 이미 저장되어 있으며(4.9) 기어와 축의 구체화 설계를 한 후 형상은 Fig. 8과 같이 된다.



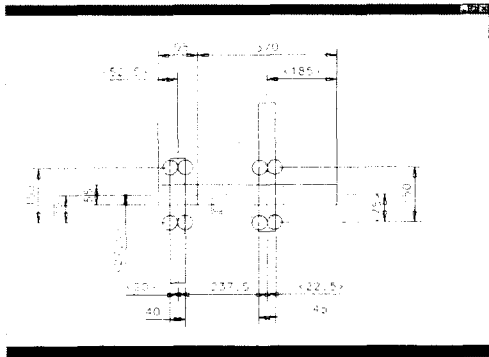


Fig. 8 Reducer reflecting embodiment design

5.5 재설계

설계 요구 조건이 변경된 경우 재설계하는 과정을 살펴 보자. 감속비(Requirement의 ratio)와 윤셋(Requirement의 offset)이 각각 3과 220으로 변경되었다. 설계자는 설계 인자의 변경이 있는 Requirement를 재설계 시작 요소 객체로 설정하며 이로 인해 설계 시스템은 Fig. 9와 같이 Reducer를 재설계하게 된다. Fig. 9에서 재설계한 결과 a는 22로, b는 66으로 계산되고 윤셋은 220을 만족하게 된다. 이 계산 결과를 사용하여 기어와 축의 구체화 설계를 거치게 되며 이로 인해 설계하고자 하는 형상은 Fig. 10과 같이 변하게 된다.

5.6 지게차 변속기 설계

제안된 시스템은 일반적인 설계에 모두 적용될 수 있는 개념이며 제안된 시스템의 실용성을 보기 위해 대우중공업(주)에서 개발하고 있는 2톤급 지게차의 변속기를 제안한 시스템으로 설계하였다. 이 설계를 위해 103개의 요소 객체를 사용하였으며 Fig. 11은 지게차 변속기가 설계된 형상이다. 이 설계 과정은 예제로 설명한 감속기의 설계 과정과 유사하게 기어 설계, 축 설계, 베어링 설계 과정을 거치나 훨씬 복잡하게 진행되며 제안한 시스템으로 효율적인 설계가 가능하였다.

6. 결론

본 연구에서는 기계 설계 과정에서 설계 인자와 설계 목표를 저장하고 활용함으로써 설계를 효율적으로 하는 방법을 제안하였다. 이 연구에서는 다음과 같은 일들이 수행되었다.

1	ratio			1.	3.
2	offset	$m^*(a+b)/2$		1.	220.

1	a	21	T	22.
2	b	74	T	66.
3	m	5	F	5.

Input fields: 500, 20, 0.5, 0.1

Fig. 9 Re-design of Reducer

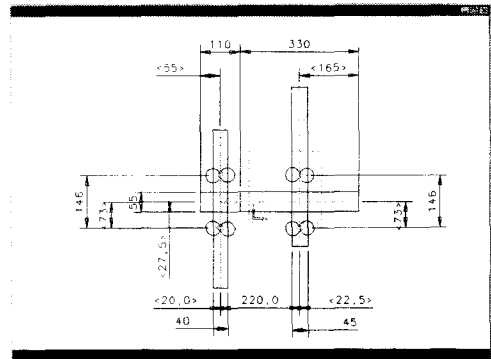


Fig. 10 Re-designed shape of reducer

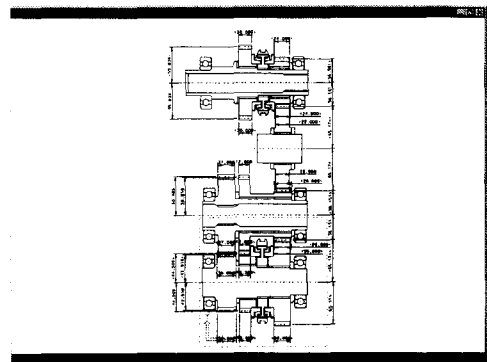


Fig. 11 Design of a forklift transmission

- 설계 인자와 설계 목표를 나타내기 위한 설계 요소의 표현 방법
- 진화 설계와 재설계에 효율적인 설계 인자의 정보 설정 방법
- 설계 목표 설정 방법
- 유전자 알고리즘을 이용하는 진화 설계 방법
- 설계 인자 또는 설계 목표가 변경되었을 때 재설계를 쉽게 하는 방법

이 연구에서 제안된 방법을 수행할 수 있는 설계 시스템이 개발되었으며 개발된 설계 시스템을 실제 개발 중인 지게차 변속기 설계에 적용하여 실용성을 증명 하였다.

본 연구를 다음과 같이 확장하면 더욱 유용한 설계 시스템이 될 것이다. 첫째로, 다양한 설계 분야에 본 설계 시스템을 적용해 여러 가지 설계 시나리오를 구축하는 것이다. 이를 위하여 지게차 변속기 설계, 굴삭기 습식 다관 클러치 브레이크 시스템 설계에 적용해 보았으며, 아크 용접 로봇 설계에 적용을 고려하고 있다. 그리고 현재는 진화 설계 개념이 요소 객체 단위로 적용 되는데, 적용 범위를 조립체 단위 또는 제품 단위로 확장 하는 것이다. 마지막으로 본 연구에 설계 인자와 설계 목표를 사용하는 다른 설계 방법론을 적용하는 것이다.

### 참고문헌

1. N.P. Suh, "The Principles of Design," Oxford University Press, 1990.
2. G. Taguchi and M.S. Phadke, "Quality Engineering through Design Optimization," Proceedings of I.E.E.E. GLOBECOM-84 Conference, pp. 1106-1113, Atlanta, GA, November, 1984.
3. J.R. Houser and D. Clausing, "The House of Quality," Harvard Business Review, pp. 63-73, May-June, 1988.
4. 이강수, 이건우, "설계 과정과 설계 정보, 설계 의도를 모델링하기 위한 프레임워크," 한국 정밀 공학회 1998 년도 추계학술대회, pp. 841-846, November, 1998.
5. K. Lee, "Required Features of CAD Systems as a Real Design Tool," Sixth IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications (GEO-6), pp. 199-212, Tokyo, Japan, December 7-9, 1998.
6. J. S. Gero and V. Kazakov, "Evolving Design Genes in Space Layout Problems," Artificial Intelligence in Engineering ,Vol. 12, No. 3, pp.163-176, 1998.
7. J. S. Gero, V. Kazakov and T. Schnier, "Genetic Engineering and Design Problems," In Evolutionary Algorithms in Engineering Applications, eds. D. Dasgupta and Z. Michalewicz, Springer Verlag, Berlin, pp.47-68, 1997. Also described in [http://www.arch.usyd.edu.au/dept\\_ads/index.html](http://www.arch.usyd.edu.au/dept_ads/index.html).
8. James Rumbaugh, Michael Blaha and William Premerlani, Frederick Eddy and William Lorensen, "Object-Oriented Modeling And Design," Prentice-Hall, 1991.
9. 이강수, "설계 이력과 설계 정보 및 의도를 수용하는 통합 설계 시스템의 개발," 박사학위 논문, 서울대학교 기계설계학과, 1999.
10. Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs," Spring-Verlag, 1996.
11. M. Wall and Matthew's Galib: A C++ Library for Genetic Algorithm Components. <http://www.lancet.mit.edu/ga/>, 1996.
12. J.P. Charles, H. Jeon and M. R. Cutkosky, "Combining Constraint Propagation and Backtracking for Distributed Engineering," In ECAI96 Workshop on Non-Standard Constraint Processing, Budapest, August 1996. Also described in <http://cdr.stanford.edu/CDR/publications.html>.
13. G.J. Kim and G.A. Bekey, "Constructing Design Plans for DFA Redesign," 1993 IEEE Conference on Robotics and Automation, pp. 312-318, 1993.