

環境の不確実性を考慮したロバストな制御回路用ハードウェア記述言語の自動構成に関する研究\*\*

Study on HDL Design Automation for Robust Electronic Controller Considering of Uncertain Environment

小島一恭\*

Kazuyuki KOJIMA

In this paper, a method of automatic controller design for electronic control systems is described. In order to automate the design of an electronic controller, an evolutionary computation is applied. First, the framework for applying the genetic algorithm to the automation of controller design is described. In particular, the coding of a chromosome is shown in detail. Then, how to make a fitness function is represented, with an air conditioner as an example, and the controller of the air conditioner is developed automatically using our proposed framework. Finally, an evolutionary simulation is performed to confirm our framework.

**Keywords: Control System, Hardware Description Language, Robust Control, Design Automation**

1. 緒言

近年の電子技術の急速な進歩，加えて電子制御で必要となるセンサー，アクチュエータの高性能化，低価格化によって，多くのシステムに電子制御が導入され，以前にも増してシステムの高性能化，エネルギー利用の高効率化が進展している。このような電子制御システムの制御器を設計する場合，設計者はセンサやアクチュエータ，マイクロプロセッサ（MPU）などの資源を有機的に結合する。システムが小規模であり，センサ信号の取得からアクチュエータの駆動へと至る信号処理に複雑な演算を必要としない場合には，制御器設計は比較的容易である。比例・積分・微分制御（PID 制御）に基づく位置決め制御や速度制御などはこのような制御の一例である。比較的容易とはいえ，もちろん，安定性や応答性などの問題を十分に考慮して設計する必要がある。しかし，より多くのセンサ，アクチュエータで構成されるより規模の大きなシステムでは制御器設計は一層複雑で困難なものとなる。障害物を回避しながら歩行

するロボットの設計は，この一例といえる。この場合，制御器の設計者はロボットを構成するすべての資源に多くの注意を払いながら制御プログラムや制御回路を開発しなければならない。近年開発されるシステムの多くは，先に述べたロボットと同様に，多くのセンサ，アクチュエータで構成され，高性能化，高効率化の背後に，設計の自由度を増したことによる設計の困難さの問題を持ち合わせる。家電製品や携帯電話，自動車など日常的に使用する多くのシステムがこのようなシステムである。

本研究では，このような制御器の設計を支援するため，制御器に Complex Programmable Logic Device (CPLD) を利用したシステムにおいて，その制御回路を記述するハードウェア記述言語 (VHDL) を進化計算により自動構成する。進化計算を適用し，制御プログラムや制御回路を自動構成する枠組みでは，従来より，得られる解の局所性に起因して，ロバスト性を欠く制御器が設計されることが問題となっている。本論文では，環境の不規則性を考慮したロバストな制御回路用 VHDL の自動構成方法について述べる。

2. 進化計算による制御器の自動構成

2.1 PLD と VHDL

本論文では，Programable Logic Device (PLD) と呼ば

\* 埼玉大学大学院理工学研究科人間支援・生産科学部門  
Department of Mechano-Robotics, Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama, Saitama 338-8570, Japan.

\*\* (原稿受付日：平成 21 年 5 月 29 日)

れる、設計者が半導体内部の論理回路を手許で書換え可能な大規模集積回路 (LSI) を進化計算を用いて自動構成する。PLD は、Complex Programable Logic Device (CPLD) と Field Programable Gate Array (FPGA) の 2 種類が存在し、双方とも VHDL により内部の論理回路を記述することができる。FPGA の場合は、内部ロジックに直接関与するのは、Configurable Logic Block (CLB) と呼ばれる論理ブロックの機能を選択するためのビット情報とインターコネクとと呼ばれる複数の CLB の接続状態を表すビット情報である。CPLD では、マクロ・セルと呼ばれる複数の機能ブロック同士の接続関係を規定する接続領域のビット情報がこれにあたる。

VHDL は内部ブロックの接続関係を表すビット情報を直接記述するのではなく、よりメタな、抽象的なレベルで言語表現し、PLD のハードウェアと切り離して論理回路を設計することができる。VHDL で表現される論理回路は論理合成ツールを使用して、ハードウェアに適したビット情報に変換され、書き込むことができる。目的のハードウェアが CPLD や FPGA の場合には、ダウンロードケーブルを介して得られた論理回路をハードウェアに書き込み、目的の論理回路をもつ半導体を手許で直ちに得ることができる。簡単な論理回路とこれを表現する VHDL を図 1 に示す。

### 2.2 遺伝的アルゴリズム (GA: Genetic Algorithm)

図 2 に GA の枠組みを示す。最適化問題の決定変数ベクトル  $x$  を  $N$  個の遺伝子座からなる染色体とみなし、この染色体  $K$  個からなる個体集合を考え、世代  $n$  における個体集

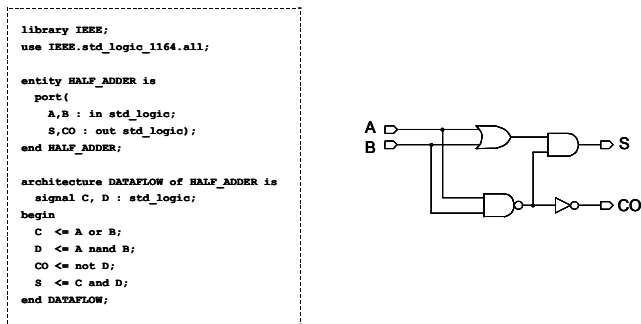


Fig.1 VHDL for a simple logical circuit

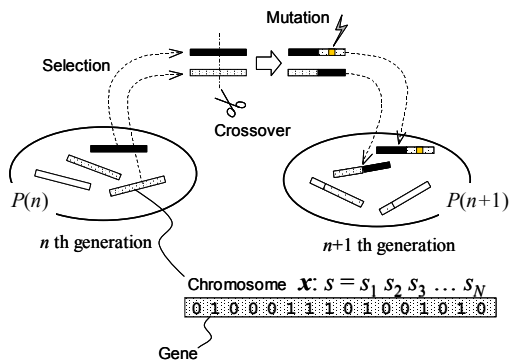


Fig.2 Outline of the genetic algorithm

合  $p(n)$  が遺伝子の複製・変異を経て次の世代  $n+1$  における個体集合  $p(n+1)$  になるものとする。このような世代の更新が繰り返され、更新のたびに、より最適値に近い解  $x$  を表す個体が選択されて、増殖するようにすれば、やがて最適解が得られるであろうとするのが GA の基本的な考え方である。適用する場合には、得ようとする解の染色体表現と解の適応度を評価する目的関数が重要である。

### 2.3 進化計算による制御器自動構成の枠組み

進化計算を用いて PLD を最適化する研究が多く行われている。従来研究<sup>1)~5)</sup>では、進化計算で使用する遺伝子に論理ブロック機能選択のビット情報とインターコネクの接続状態を表すビット情報を対応させて PLD の内部ロジックを自動構成してきた。このようにすると、進化計算で使用する遺伝子の表現が簡便であり、また、ハードウェアに依存して染色体長が決定されるため、自動構成の計算における解探索範囲を狭める効果も期待できる。その一方で、設計者は使用するハードウェアの規模とピン配置をあらかじめ決定しておかなければならないというデメリットがある。設計の前段階でそれらを決定することは難しい。そこで、本論文ではこれらのビット情報を自動構成の対象とするのではなく、ハードウェアとは切り離して論理回路を表現することのできる VHDL を自動構成の対象とし、VHDL を進化計算の染色体として表現する。この方法では、染色体表現は複雑になるが、ロジックが自動構成された後に論理回路の規模に応じて適切なハードウェアを選択することができ、また、ピン配置も自由に選択することが可能となる。さらに、CPLD や FPGA だけではなく、ASIC など HDL で設計可能な他の半導体への転用も可能である。その際、染色体の再コーディングや進化計算の再計算は全く必要ない。

Xilinx 製 XC9572<sup>6)</sup> を一例に適用の枠組みを述べる。図 3 に 44 ピンの XC9572 の内部ブロック図を示す。この CPLD は 72 個のマクロセルと 1600 個のゲートをもち、44 ピンのうち 34 ピンを入出力 (I/O) として利用できる。これらの I/O をどのようなピン数でセンサやアクチュエータの信号として割付けるかは、設計者に委ねられる。図 4 の例では、1 個のセンサに 8 ビット、2 個のアクチュエータにそれぞれ 8 ビットを割当てている。

制御器として自動構成された VHDL の一例を図 5 に示す。この VHDL は図 4 で例示した I/O に対応している。VHDL は、

- (a) エンティティ
- (b) 信号宣言
- (c) アーキテクチャ

の 3 つの部分で構成される。(a) エンティティに PLD の I/

0を、(b) 信号宣言に内部信号を、それぞれ定義する。また、信号の型として「std\_logic」という単純な1ビットの論理を表す型のほか、多ビットの論理を表す「std\_logic\_vector」という型を使用できる。ここでは「std\_logic」型のみ取り扱うが、後述の染色体表現は「std\_logic\_vector」を取り扱うことも可能である。また、「std\_logic\_vector」型は複数の「std\_logic」型で表現できるため、「std\_logic」型のみでの取り扱いによって、自動構成される論理回路が制限されるわけではない。

図6～図8に染色体構造の一部を示す。VHDLの複雑な記述を表現するために、染色体を可変長の木構造とする。染色体の先頭には図6に示す通り、入力数、出力数、信号数の3つの値を格納する。すべての入出力信号、内部信号が同じ「std\_logic」型である場合には、信号数から入出力信号と内部信号のVHDLの記述を復元することができる。図6では入力信号数が8、出力信号数が16、内部信号数が2であることを意味しており、染色体上のこの数からVHDLの入出力信号の記述(図5(a))と内部信号の記述

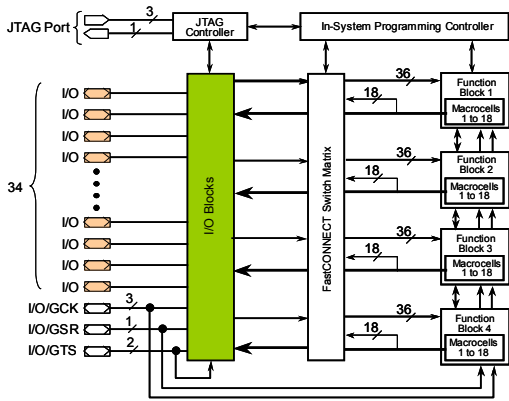


Fig.3 XC9572 architecture

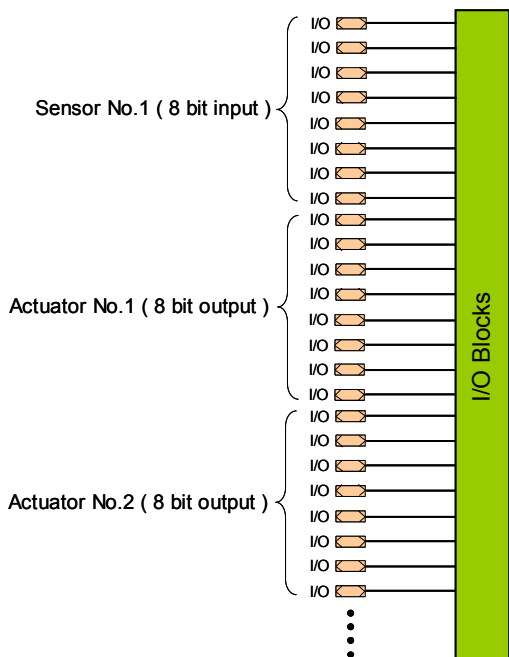


Fig.4 Assignments of I/Os

(図5(b))が復元される。

図7はVHDLの代入文を表す染色体構造で、図5(d)に対応する。また、図8はVHDLのプロセス文における条件文を表す染色体構造で、図5(e)のセンシティビティリストに「S000」、「DI002」をもつプロセス文に対応する。この例では、プロセス文の中にさらに条件文があり、ネストレベルが高く、その分、染色体の階層も深くなっている。プロセス文のセンシティビティリストやプロセス文内の複数行の記述に相当する染色体は可変長になっており、センシティビティリストに列挙される変数が変化したり、プロセス文内の記述が複数行にわたっても対応することができる。このように、染色体を木構造とし、各階層の遺伝子座によって可変長の構造を採用することにより、複雑なVHDLの記述を染色体として表現することが可能となる。

### 2.4 遺伝的操作

GAでは、複数の染色体からなる集団を形成し、その中から適応度の高い2つの染色体を選択し、交叉、突然変異の遺伝的操作を加え、新たな染色体を複製する。この遺伝的操作を繰り返すことで元の集団の持つ良い形質を維持しつつ新たな形質をもつ集団を生成し、集団全体の適応度を上げ、目的の解を探索する。

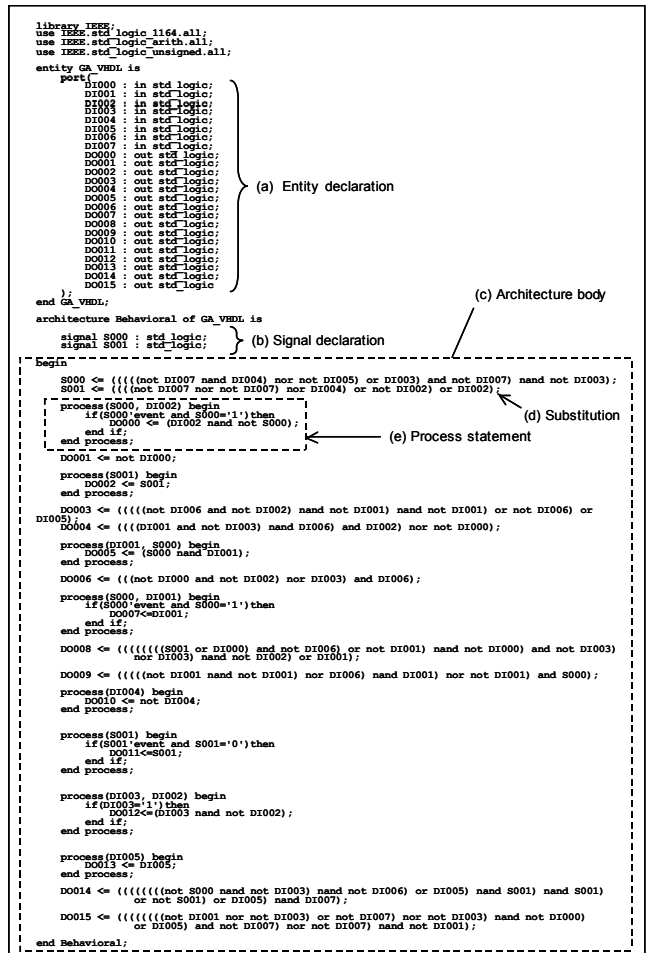


Fig.5 Auto-generated VHDL

交叉は選択された2つの染色体をそれぞれある位置で分割し一部を入れ替える操作、突然変異は1つの染色体の一部を変化させる操作である。どちらの操作を加える場合も、染色体上の遺伝子座に矛盾が生じないようにする必要がある。本手法のように複雑な階層構造をもつ染色体では、単純にランダムに染色体上の位置を決定し、交叉・突然変異の遺伝的操作を作用させると、直ちに染色体に矛盾を生じる。そこで、染色体の生成と遺伝的操作に関して次の制限を課す。

- (1) 最上位層における染色体長は、内部信号数と出力信号数の和に1だけ加えた長さとする。
- (2) すべての信号は直接染色体上に符号化せず、参照

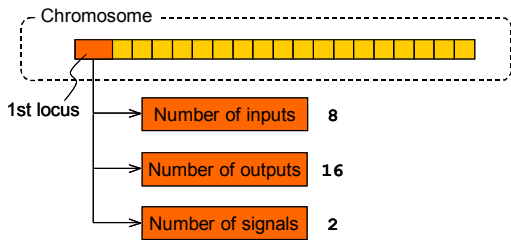


Fig.6 Signal definition on the first locus

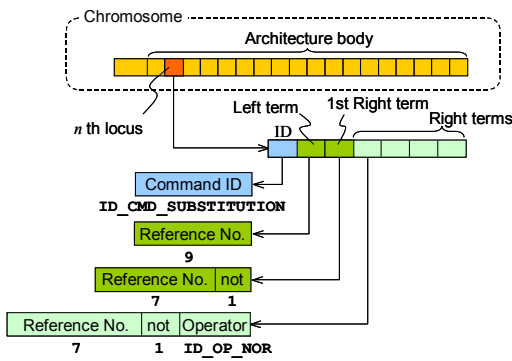


Fig.7 Substitution

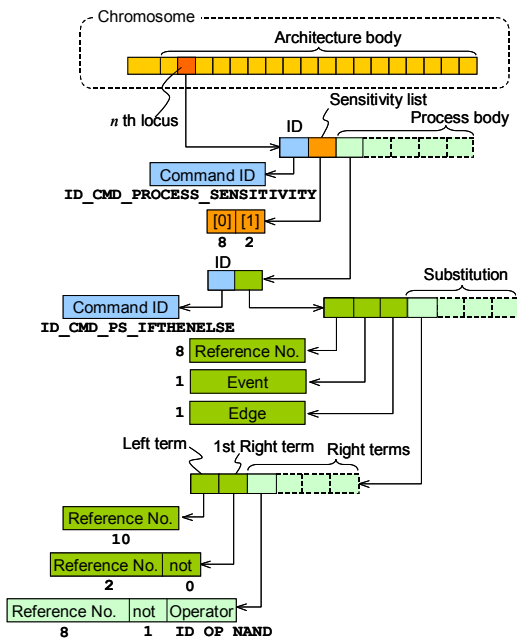


Fig.8 If-then-else statement in process body

番号を用いて符号化する。

- (3) 最上位層の染色体のエンティティ宣言部には内部信号、出力信号の順で、かつ、優先度の低いものから、それぞれ一つの遺伝子座に一回だけ符号化する。
- (4) すべての内部信号に対して優先度を決め、優先度の高い内部信号は優先度の低い内部信号と入力信号のみで記述する。
- (5) 交叉は最上位層の染色体のみで行い、交叉点は選択された2つの染色体のうち、短い側の染色体長を超えない位置で行い、交叉点が奇数の場合は、先頭の遺伝子座を入れ替える。

以上の制限を設けることで、矛盾を生じさせることなく、初期化、交叉、突然変異が可能となる。

図9に長さの異なる2つの染色体を例示する。それぞれの染色体の右側の表は染色体への符号化で使用される参照番号を示す。

染色体(A)は、内部信号が5、出力信号が16であるため、染色体長はこの和に1を加えた22となる。同様に染色体(B)では19となる。出力信号数は設計者が決定するため、長さの違いは初期化時にランダムに決定される内部信号数に依存する。遺伝的操作によりこの長さは変化する。

染色体の交叉の例を図10に示す。ここでは、左側の2つの染色体(A)、(B)を交叉し、右側の(A')、(B')が得られるものとする。例えば第6番目の遺伝子座の直後を交叉点とする場合、図示のごとく最上位層の6番目の遺伝子座で染色体を分断し、下層の構造をまとめて交換する。染色体(A)、(B)は長さと同層の構造が異なるが、このように最上位層のみで交叉を行うことでVHDLに復元しても矛盾のな

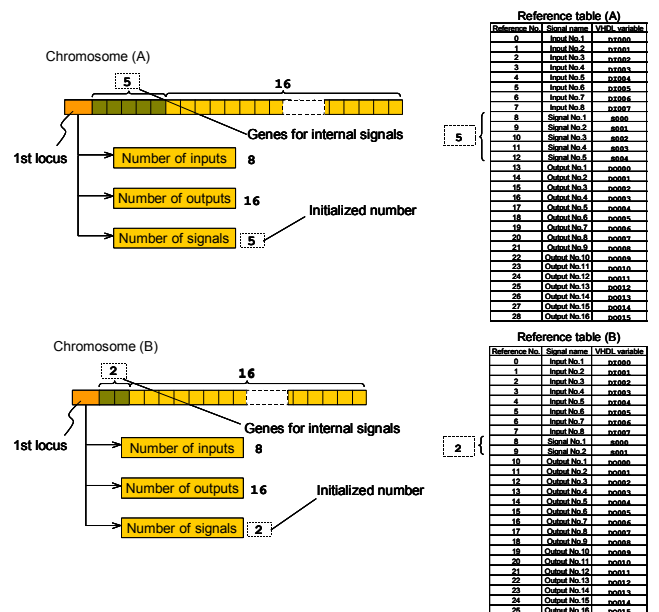


Fig.9 Two different length chromosomes

い新たな染色体 (A'), (B) が得られる。

### 3. 不確実環境の考慮と空調システムへの応用

#### 3.1 空調システム

本論文では、制御器の一例として空調制御器を取上げ、進化計算を用いた制御ロジックの自動構成手法の適用例を示す。図 11 に対象となる実験装置を示す。ファンの回転数に応じた量の外気が空調装置内に取り込まれ、一度、すべての空気が冷却され、同時に除湿される。その後、一部がヒータにより再加熱される。再加熱された暖気と冷気との混合比はエアミックスドアの開度により調整される。室内の温度は、ファンの回転数とエアミックスドアの開度を制御することにより調整される。VHDL による制御器の構成を図 12 に示す。ここでは、温冷感を 8 ビットの入力、ファン回転数、ミックスドア開度をそれぞれ 8 ビットの出力とする制御器を考える。

#### 3.2 空調システムのモデル化

VHDL による制御器の制御性能をシミュレーションにより評価するため、空調システムを次のようにモデル化する。空調システムの流路を図 11 の (0) ~ (8) で示す検査体積に分割し、各検査体積における熱収支を考える。図 13 に 3 つの検査体積で構成される流路を示す。この流路における質量流量  $G_i$  [kg/s] は、ファンモータの開度を  $n$ 、全開時の質量流量を  $G_{max}$  [kg/s] として、

$$G_0 = G_i = nG_{max} \quad (0 \leq n \leq 1) \quad (1)$$

と表される。全質量流量  $G_0$  [kg/s] は、空気の質量流量  $G_{a0}$  [kg/s] と水蒸気の質量流量  $G_{w0}$  [kg/s] の和であるので、

$$G_0 = G_{a0} + G_{w0} \quad (2)$$

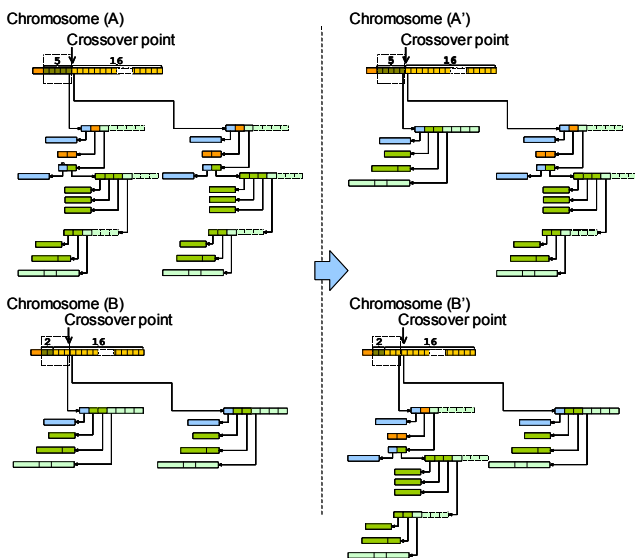


Fig.10 Two different length chromosomes

検査体積  $i=0$  における水蒸気圧  $P_{w0}$  は、湿度を  $\phi_0$  として、

$$P_{w0} = \phi_0 \cdot P_{s0} \quad (3)$$

ここで、 $P_{s0}$  は飽和水蒸気圧で、Tetens の式<sup>7)</sup>により求める。

$$P_{s0} = 610.78 \times e^{\frac{17.2694(T_0-273.15)}{(T_0-273.15)+238.3}} \quad (4)$$

相対湿度  $x_0$  は、水蒸気  $P_{w0}$  から次式により求められる。

$$x_0 = 0.622 \frac{P_{w0}}{P_0 - P_{w0}} \quad (5)$$

空気の質量流量  $G_{a0}$  は気体の状態方程式より、

$$G_{a0} = P_{a0} \cdot \frac{V}{R_a T_0} \quad (6)$$

と求められる。ここで、 $R_a$  は気体定数である。また、水蒸気の質量流量  $G_{w0}$  は次式で与えられる。

$$G_{w0} = x_0 \cdot G_{a0} \quad (7)$$

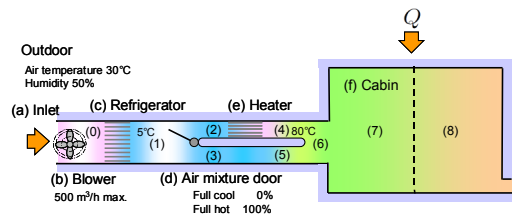


Fig.11 Air-conditioning system

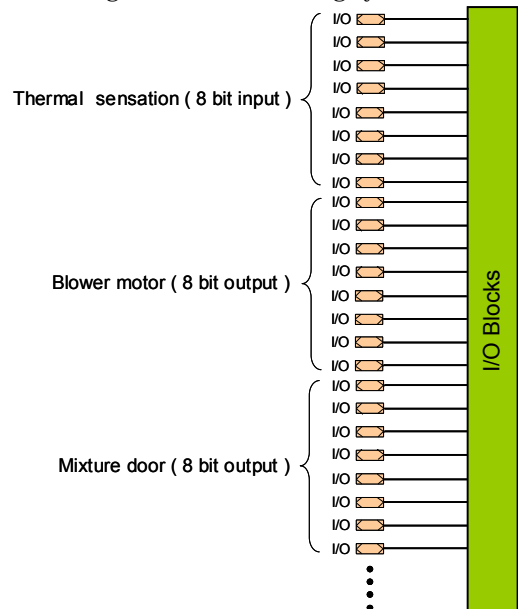


Fig.12 Two different length chromosomes

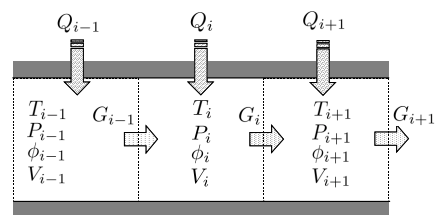


Fig.13 Control volume

空気の質量流量は流路内では一定であり、水蒸気の質量流量は除湿される場合には減少し、それ以外は一定とする。単位時間  $dt$  [秒] 後の検査体積  $i$  の温度  $T_i$ , 湿度  $x_i$ , 質量  $M_i$  は、

$$T_i' = \frac{G_{i-1}C_{i-1}T_{i-1}dt + (M_i - G_i dt)C_i T_i}{G_{i-1}C_{i-1}dt + (M_i - G_i dt)C_i} + \frac{Q_i dt}{M_i C_i} \quad (8)$$

$$x_i' = \frac{M_i x_i + (1 + x_i)(G_{wi-1} - G_{wi})dt}{M_i + (1 + x_i)(G_{ai-1} - G_{ai})dt} \quad (9)$$

$$M_i' = M_i + (G_{i-1} - G_i)dt \quad (10)$$

となる。ここで、比熱  $C_i$  [kJ/kg・K] は、次式により与えられる。

$$C_i = \frac{1.005 + x_i \{ (2501.6/T_i) + 1.859 \}}{1 + x_i} \quad (11)$$

ミックスドアでの分岐流路では、ミックスドアの開度  $m$  に応じてそれぞれの流路の質量流量を決定する。図 5 (1) の分岐流路では分岐後の図 12 (2), (3) の位置の質量流量  $G_2, G_3$  と分岐前の質量流量  $G_1$  との間に次式が成り立つ。

$$G_2 = mG_1 \quad (12)$$

$$G_3 = (1 - m)G_1 \quad (13)$$

同様に、合流流路 (図 12 (4), (5), (6)) において次式が成り立つ。

$$G_6 = G_4 + G_5 \quad (14)$$

以上より得られた温度、湿度、流量に加え、代謝量、着衣量を設定し、温熱的快適指標 PMV を求める。PMV (Predicted Mean Vote) は、人体の定常熱収支式と温冷感や快適感に関する 1000 人の被験者の実験結果を基に導出された温熱指標で、空調制御でよく用いられる指標の一つである<sup>8)</sup>。PMV=0 のとき快適と感じる人の割合が最大となり、この場合の温冷感が中立とされる。また、+3 は暑い、+2 は暖かい、+1 はやや暖かい、-1 はやや涼しい、-2 は涼しい、-3 は寒いと温冷感が対応づけられている。

### 3.3 不規則環境を考慮した適応度の算出

評価関数は目標 PMV を 0 とし、一定時間の空調制御シミュレーションを行い、目標 PMV との偏差を積算し、積算値の最小のものほど評価値が高くなるようにする。

$$\text{Fitness} = - \int_0^{t_{\text{end}}} |PMV_{\text{ctrl}} - PMV_{\text{target}}| dt \quad (15)$$

進化計算による最適化の枠組みを制御システムに応用しようとする場合、従来研究では進化過程で使用した環境と

実際に使用する環境の不整合や環境の不規則性のため、制御器がうまく機能しないことが問題となることがあった。簡単なフィードバック系からロボットの歩行まで様々なシステムに進化計算の枠組みが応用されてきているが、同種の問題を孕んでいる。多くの例ではこの問題を明示していないが、進化計算の実システムへの応用では潜在的にこの問題をもつものと考えられる。そこで、本研究ではシミュレーションレベルではあるものの、不規則環境を考慮した適応度の計算を試みる。染色体の適応度計算の際に不規則性を取り入れる。本論文の空調システムへの応用の場合には、外乱として与えている熱負荷を、適応度計算する度に不規則に与える。このように適応度計算を行うと、同じ染色体であっても計算するごとに適応度が異なり、また、同じ集団内の染色体に対して異なる尺度で適応度を求めることになるが、このことによって染色体の集団の多様性を維持するとともに、集団として不規則環境に対するロバスト性能を蓄積し、結果として、不規則環境に対してロバストな制御器を得ることが期待できる。

### 3.4 自動構成された制御器の制御性能

個体数 50, 交叉率 1.0, 突然変異率 0.5, トーナメント戦略, トーナメントサイズ 10 で進化計算を行い制御器を自動構成した。自動構成された制御器による制御シミュ

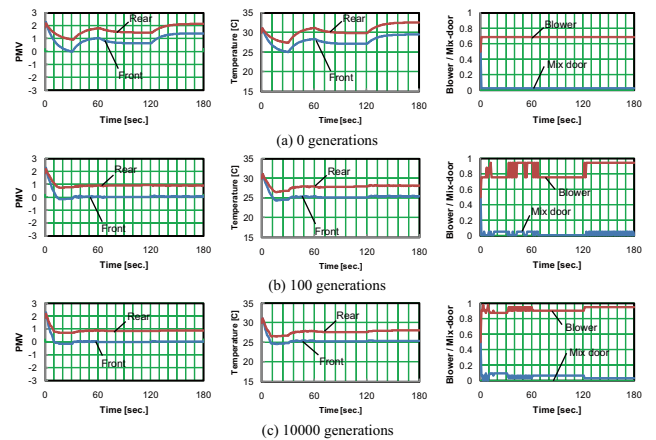


Fig 14 Simulation results

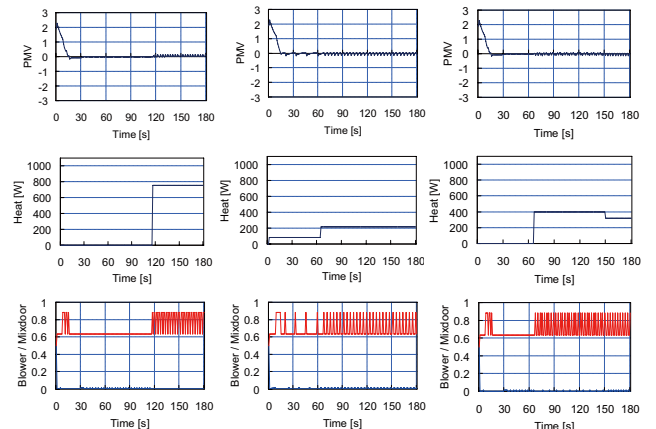


Fig 15 Simulation results

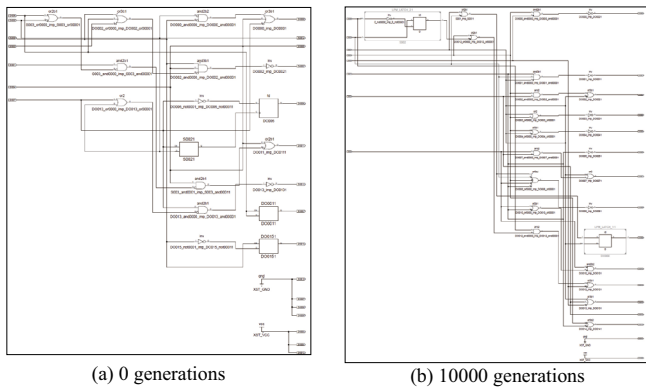


Fig.16 Auto-generated logical circuit

レーションの結果を図 14, 15 に示す. 図 14 は進化計算の計算世代が 0 世代, 100 世代, 10000 世代の場合の制御シミュレーション結果を示す. すべてのグラフにおいて確認できる, 30 秒, 60 秒, 120 秒のグラフの傾向の変化は熱負荷が変化したことによるものである. 左側の PMV のグラフに注目すると, 0 世代では, Front と示される PMV のグラフが目標 PMV の 0 から外れ, 熱負荷の増減に応じて上下に変化している. このとき, プロワ回転数, ミックスドア開度は適切な制御がなされていない. 100 世代, 10000 世代の進化計算で得られた制御器では, 熱負荷の変動に応じてプロワ回転数, ミックスドア開度の運転モードが変化し, 目標 PMV の 0 付近で安定するよう機能していることが確認できる.

図 15 は同一の制御器を用い 3 つの異なる熱負荷環境下で制御シミュレーションを行った結果を示す. このシミュレーションで使用した制御器は 10000 世代の進化計算により自動構成されたものである. いずれの熱負荷に対しても, プロワ回転数, ミックスドア開度を変化させ, 目標 PMV の 0 付近で安定する制御が行われていることが確認できる.

図 16 に 0 世代と 10000 世代の回路図を示す. 左側の 8 つの入力から右側の 16 の出力までの論理回路は 0 世代より 10000 世代のほうがやや複雑になっている.

#### 4. 結 言

本研究では, 制御システムの制御器設計支援のため, 制御器に PLD を利用したシステムにおいて, その制御回路を記述する VHDL を進化計算により自動構成する方法について述べた. VHDL を多重の可変長の木構造をもつ染色体で表現することを提案し, さらに, 染色体構造と遺伝的操作に 5 つの制限を設けることで, 進化計算を進めても常に染色体が表現する VHDL に矛盾を生じさせない方法を示した.

また, 進化計算を適用して制御プログラムや制御回路を

自動構成する枠組みで問題となっている不規則環境に対するロバスト性を解決するため, 適応度計算に不規則性を導入する方法を提案した. 空調システムを例にシミュレーションを行い, 本枠組みの有用性を確認した.

#### 謝 辞

本研究の一部は, 平成 20 年度科学研究費補助金基盤研究 (C) 課題番号 20500664, 埼玉大学平成 20 年度総合研究機構研究プロジェクト (研究経費) A08-24 の支援により行われた. ここに謝意を表す.

#### 参考文献

- 1) T. Higuchi et al., Evolvable Hardware with Genetic Learning: A First Step Towards Building a Darwin Machine, Proceedings of the 2nd International Conference on the Simulation of Adaptive Behavior, MIT Press, pp. 417, 1992.
- 2) H. Hemmi, J. Mizoguchi and K. Shimohara, AdAM: A Hardware Evolutionary System, Proc. 1997 IEEE Conf. Evolutionary Computat.(CEC'97), pp. 193-196, 1997.
- 3) I. Kajitani and T. Higuchi, Developments of Myoelectric Controllers for Hand Prostheses, Proc. of the Myoelectric Controls Symposium 2005, pp. 107-111, 2005.
- 4) H. Sakanashi, M. Iwata and T. Higuchi, Evolvable Hardware for Lossless Compression of Very High Resolution Bi-level Images, IEEE Proceedings-Computers and Digital Techniques, Vol.151, No.4, pp.277-286, 2004.
- 5) D. A. Gwaltney and M. I. Ferguson, Hardware Evolution of Analog Speed Controllers for a DC Motor, Proc. of the Genetic and Evolutionary Computation Conference 2003(GECCO-2003), pp. 442-453, 2003.
- 6) Xilinx, XC9572 In-System Programmable CPLD Product Specification, Xilinx, pp. 2, 1998.
- 7) O.Tetens, Uber einige meteorogogische Begriffe, Zeitschrift fur Geophysik, Vol.6, (1930), 297.
- 8) P. O. Fanger, Thermal Comfort, McGraw-Hill, 1970.
- 9) D. E. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
- 10) 三宮信夫・喜多一・玉置久・岩本貴司, 遺伝的アルゴリズムと最適化, 朝倉書店 (1998), 13.