

プロジェクト名： 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

プロジェクト代表者：小島 一恭（理工学研究科・助教）

1. 概要

本研究プロジェクトではFPGA（Field Programmable Gate Array）、CPLD（Complex Programmable Logic Device）などのプログラマブルICの制御器の設計を支援するため、制御器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方法を提案した。進化計算を適用し設計最適化を行う枠組みでは、従来、得られる準最適解が局所解に陥ることに起因してロバスト性を欠く制御器が得られることが問題となっていた。また、VHDLのようなプログラムの最適化に進化計算の枠組みを適用する場合、計算過程で解候補に矛盾が生じ、計算効率が上がらないという問題があった。

そこで本研究では、最適化計算における評価関数の与え方を工夫することにより、環境の不規則性を考慮したロバストな制御回路用VHDLを自動構成した。また、進化計算を適用する際の解候補の表現方法、演算方法を工夫することにより、解候補の矛盾を無くし、計算効率を向上する方法について提案した。

2. 制御器自動構成の枠組み

本研究では、制御器にCPLDを使用し、内部の論理回路を記述するVHDLを進化計算を用いて自動構成する。

PLD（Programmable Logic Device）とは、設計者が半導体内部の論理回路を手許で書き換え可能な大規模集積回路（LSI）で、論理回路の記述にはVHDLを使用する。また、進化計算は生物の進化過程に着想を得た組合せ最適化手法の一つである。解を染色体というデータ構造で表現し、これを複数用意して評価関数で個々に評価した後に、評価の高い複数の染色体を組合わせて新たな染色体を生成し、解集団の中により評価の高い解を生成させ、何回かの繰り返し計算の後、最も評価の高い解を目的の解とする方法である。

本研究では、染色体の表現方法と演算方法を工夫することにより、計算過程で生じる解候補の矛盾を回避し、

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
entity GA_VHDL is
port(
DI000 : in std_logic;
DI001 : in std_logic;
DI002 : in std_logic;
DI003 : in std_logic;
DI004 : in std_logic;
DI005 : in std_logic;
DI006 : in std_logic;
DI007 : in std_logic;
DO000 : out std_logic;
DO001 : out std_logic;
DO002 : out std_logic;
DO003 : out std_logic;
DO004 : out std_logic;
DO005 : out std_logic;
DO006 : out std_logic;
DO007 : out std_logic;
DO008 : out std_logic;
DO009 : out std_logic;
DO010 : out std_logic;
DO011 : out std_logic;
DO012 : out std_logic;
DO013 : out std_logic;
DO014 : out std_logic;
DO015 : out std_logic
);
end GA_VHDL;
architecture Behavioral of GA_VHDL is
signal S000 : std_logic;
signal S001 : std_logic;
begin
S000 <= (((not DI007 nand DI004) nor not DI005) or DI003) and not DI007) nand not DI003);
S001 <= (((not DI007 nor not DI007) nor not DI004) or not DI002) or not DI002);
-- process(S000, DI002) begin
-- if(S000'event and S000='1') then
--     S000 <= (DI002 nand not S000);
-- end if;
-- end process;
DO001 <= not DI000;
process(S001) begin
DO002 <= S001;
end process;
DO003 <= (((not DI006 and not DI002) nand not DI001) nand not DI001) or not DI006) or
DI003);
DO004 <= (((DI001 and not DI003) nand DI006) and DI002) nor not DI000);
process(DI001, S000) begin
DO005 <= (S000 nand DI001);
end process;
DO006 <= (((not DI000 and not DI002) nor DI003) and DI006);
process(S000, DI001) begin
if(S000'event and S000='1') then
DO007 <= DI001;
end if;
end process;
DO008 <= ((((((S001 or DI000) and not DI006) or not DI001) nand not DI000) and not DI003)
nand not DI003) nand not DI002) or DI001);
DO009 <= (((not DI001 nand not DI001) nor DI006) nand DI001) nor not DI001) and S000);
process(DI004) begin
DO010 <= not DI004;
end process;
process(S001) begin
if(S001'event and S001='0') then
DO011 <= S001;
end if;
end process;
process(DI003, DI002) begin
if(DI003='1') then
DO012 <= (DI003 nand not DI002);
end if;
end process;
process(DI005) begin
DO013 <= DI005;
end process;
DO014 <= ((((((not S000 nand not DI003) nand not DI006) or DI005) nand S001) nand S001)
or not S001) or DI005) nand DI007);
DO015 <= ((((((not DI001 nor not DI003) or not DI007) nor not DI003) nand not DI000)
or DI005) and not DI007) nor not DI007) nand not DI001);
end Behavioral;
    
```

図1 VHDLの記述例

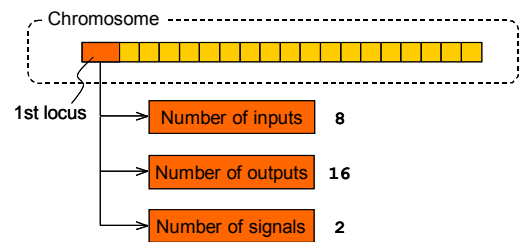


図2 染色体上の信号の定義

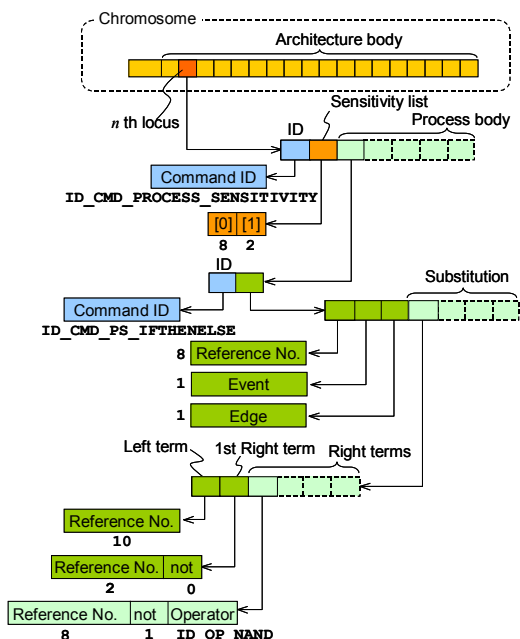


図3 プロセス文に対応する染色体の部分

計算効率を向上する。図1にVHDLの記述の一例を、図2、図3に記述の一部に対応する染色体を示す。VHDLの複雑な記述を表現するため、染色体を多重の木構造としている。図2に示すごとく上層の先頭にCPLDへの入力数、出力数、そして内部の信号数を格納する。また、図3は図1(d)のVHDLのプロセス文に対応する染色体構造である。条件文の引数の個数や条件文内の代入文の個数、代入文の右辺の項の個数などは未定であり、これに対応させるために、染色体は各層において可変長とする。

さらに、進化計算の演算過程において次の制限を課す。

- (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上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を入れ替える。

以上のように染色体を表現し演算することで、計算過程における解の矛盾を回避し、計算効率を高めることができる。

3. 環境の不確実性を考慮した制御器の生成法

次に評価関数について述べる。本手法の最も大きな特徴は、解を評価する度に評価値が異なる点である。すなわち、実際の系での不確実性を考慮するために、評価関数に不確実性を表現する機構を取り入れた点に特徴がある。従来の進化計算の枠組みでは一つの染色体に対して一つの評価値が決定された。しかし、本手法では評価関数に不確実性が導入されているために、一つの染色体に対する評価値は一つに固定されず、ある一定の不確実性を持つ。

図4に空調システムの概略図を利用し適用例を示す。本システムは左から流入した空気を冷却再加熱して右側の室内を空調するモデルである。室内モデルには外乱として熱負荷 Q が与えてある。 Q の変動を考慮するために、従来の進化計算の枠組みでは、一つの染色体に対して一意に評価値が決定させるために評価関数の計算の際に常に同じ変動を与えていた。本手法では、環境の不確実性を考慮するため Q をランダムに与える。評価値は計算する度に異なるが、環境の不確実性に対しロバストな制御器を得ることが可能となる。図5に生成された制御器の性能を示す。異なる熱負荷変動に対して快適度が一定になるように制御されていることが確認できる。

4. 研究業績

- (1) K.Kojima and K.Watanuki: Supporting Air-Conditioning Controller Design Using Evolutionary Computation, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol.2, No.1(2008).
- (2) K.Kojima, K.Watanuki: Supporting VHDL Design for Air-Conditioning Controller Using Evolutionary Computation, Proceedings of the 17th World Congress The International Federation of Automatic Control(IFAC'08), CD-ROM, pp.12318-12323(2008).
- (3) K.Kojima: Prediction of Individual Thermal Sensation Using Unspecified Sensors in Sensor Networks, International Conference on Control, Automation and Systems 2008(ICCAS2008), pp.123-126(2008).
- (4) 堀竹直, 綿貫啓一, 小島一恭: 遺伝的プログラミングによる多様な環境に適応する多足歩行ロボットの歩行獲得, デザインシンポジウム2008講演論文集, pp.99-100(2008).
- (5) 小島一恭, センサネットワークを用いた複数居住者の個別快適度予測, 第43回空調和・冷凍連合講演会講演論文集, pp.61-64(2009).

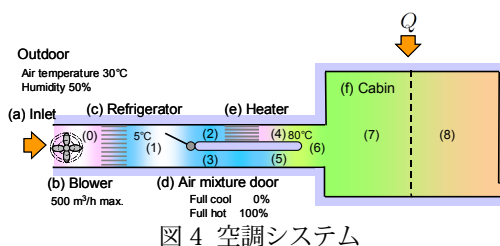


図4 空調システム

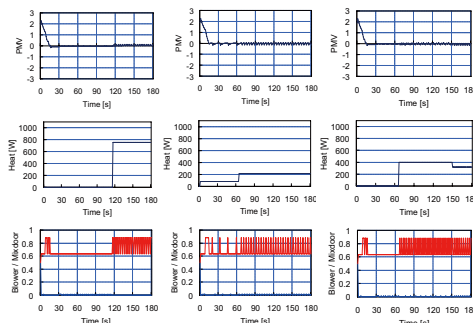


図5 生成された制御器の性能