Technical Reports on Information and
Computer Science from Kochi
Vol. 3 (2011), No. 1

FDLによる組込みソフトウェア開発ガイドライン

Development guideline for built-in software written in FDL

多和田　侑，森　雄一郎
Yu Tawata, Yuichiro Mori
高知大学大学院 理学専攻 情報科学分野
Department of Information Science, Graduate School of Science, Kochi Univ.

1. はじめに

1996年にファジィシステムを記述するための言語としてファジィシステム記述言語（FDL：Fuzzy systems Description Language）の言語仕様が策定された[1]．FDL言語仕様はC言語仕様を拡張したものであり，C言語の文法にファジィ推論を実装するための文法を追加したものである．

FDL言語仕様の策定と同時に，FDL処理系であるfdltocとFDLライブラリ関数が開発された[2][3][4]．現在，ファジィシステム記述言語（FDL）は一般的なOSのUNIX環境上でFDL処理系を用いたシステム開発が可能である[5]．しかし，組込み向けのファジィシステムも構築されていることから，FDLを用いた組込みソフトウェア開発について検討する．

2. 組込み向けのファジィ推論の実装

現在，組込み向けにファジィ推論を実装する手法としてはLUTを利用する方法が一般的である．LUTを利用していたのは，組込みCPUの処理能力の低さが要因として挙げられる．だが，近年の組込みCPUの性能の向上により，浮動小数点に関する処理能力も向上している．よって，新たにFDLによる組込み向けファジィ推論の実装を提案する．

2.1. ファジィ推論の構築

はじめに一般的なファジィ推論を構築する際の手順について説明する．ファジィ推論を構築するには，ファジィ推論の入出力の決定，推論ルールの構築，ファジィ集合の定義という手順を行う．そして，ファジィ推論結果が望ましくない結果であった場合にはファジィ推論のチューニングを行う．

ファジィ推論のチューニングでは主に推論ルールの再構築，ファジィ集合の再定義が行われる．ファジィ推論のチューニングはファジィ推論結果に深く関係するため，チューニングには時間を要し，ファジィ推論の構築手順の中で一番重要なプロセスである．

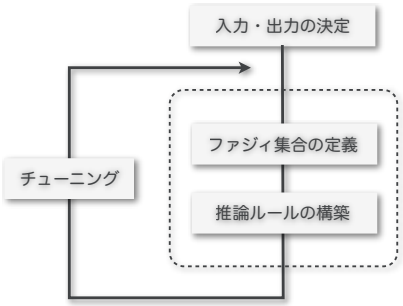

図 1 ファジィ推論の構築作業

2.2. LUTによるファジィ推論の実装

LUTによるファジィ推論の実装について考察する．LUTを利用してファジィ推論を実装することにより，ファジィ推論の計算部をプログラムに含めることなくファジィ推論を実装することが出来る．しかし，LUTでの実装ではファジィ推論の構築作業以外に，LUT生成プロセスが必要になる．ファジィ推論のチューニング作業ではLUT生成プロセスも再検討しなければならない．

LUTによるファジィ推論の実装の利点は，ファジィ推論過程を省略できる点である．つまり，高速にファジィ推論を行うことが出来る．欠点としては，ファジィ推論を実装するためのプロセスの増加，高精度な推論をするためにはLUTサイズを大きくしなければならないことが挙げられる．

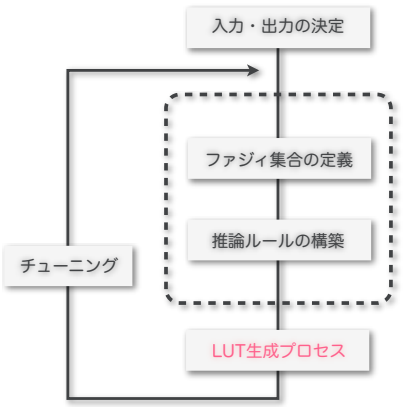

図 2 LUTによるファジィ推論の実装作業

2.3. FDLによるファジィ推論の構築

FDLによるファジィ推論の実装について考察する．FDLを用いてファジィ推論を実装することにより，ファジィ推論の計算部をプログラムに含めてファジィ推論を実装することが出来る．FDLではファジィ集合の定義，推論ルールの構築を記述することが出来る．ファジィ集合はメンバーシップ関数として定義し，推論ルールはルールベースとして構築する．

FDLによるファジィ推論の実装の主な利点は，状況に応じて柔軟にファジィ推論を行え，ソースコードの可読性が向上することにより，ファジィ推論のチューニング作業が円滑に行える点である．欠点はファジィ推論を行うための計算量の増加と，リソース消費量が増加することである．

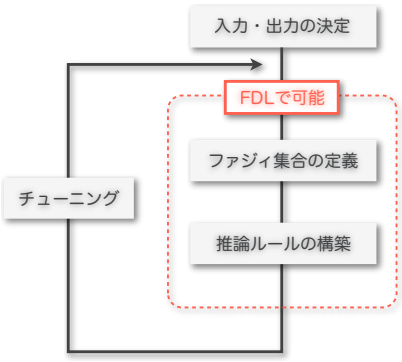

図 3 FDLによるファジィ推論の実装作業

2.4. 効率的なファジィ推論の実装

組込みソフトウェア開発ではシステム要件やハードウェアの制約から，主に処理時間とリソース消費量に配慮しながら開発が進められる．LUTによるファジィ推論の実装とFDLによるファジィ推論の実装は，互いに利点と欠点があり，どちらを採用するかの判断は難しい．LUTによる実装は計算量を削減でき，FDLによる実装ではチューニングを容易に行える．つまり，両手法の利点を活かせるような開発が望ましい．よって，組込みソフトウェア開発をファジィ推論の構築とファジィ推論を利用する処理の開発に分け，それぞれの手法の利点を活かせるようにする．

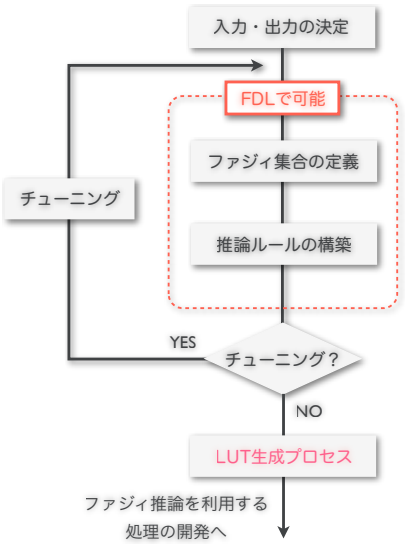

図 4 両手法を採用したファジィ推論の実装作業

図4にあるように，推論のチューニングを必要かどうかで実装方法を変更する．チューニングが必要な場合はFDLでファジィ推論を実装し，チューニングが必要ない場合はLUTでファジィ推論を実装する．ファジィ推論の実装を図3のようなフローにすることで，両手法の利点を最大限に活かすことが出来る．

3. 開発ガイドライン

FDLで組込み向けファジィ推論を実装する場合も処理時間とリソース消費量に配慮しなければならない．つまり，FDLによるファジィ推論構築の特徴を熟知しておく必要がある．従って，本研究では円滑にFDLによる組込み向けファジィ推論の実装を支援するために，FDLによる組込みソフトウェア開発ガイドラインを提案する．

開発ガイドラインでは，プログラムサイズとスタックメモリ消費量，ヒープメモリ消費量，ファジィ推論の計算時間に関する指針を示す．

3.1. プログラムサイズ

FDLプログラムサイズについて考察する．FDLでファジィ推論をするためにはFDLライブラリ関数を必須である．そのFDLライブラリ関数の中にはC言語の標準ライブラリ関数を用いて実装されている関数がある．つまり，特定の関数を利用すると，C言語の標準ライブラリをリンクしてしまう．FDLを用いて一般的なファジィ推論をする上では問題ないが，メンバーシップ関数を他のファジィ型にキャス

トしたり，ファジィ型の操作関数を利用すると，C言語の標準ライブラリ関数をリンクしてしまう．

```
// メンバーシップ関数のキャスト
fuzzy TYPE_A {double,-20,20,10};
fuzzy TYPE_B {double,0,20,20};
fuzzy TYPE_A a = triangle{-10,-5,5};
fuzzy TYPE_B b;
b = (TYPE_B)a;
```

図 5 プログラムサイズが肥大化する記述例

3.2. メモリ消費量

リソース量のスタックメモリサイズについて考察する．本研究ではファジィ集合をローカル変数として定義した場合を扱う．

FDLでのファジィ集合はファジィ型の定義とメンバーシップ関数の定義をすることで表現する．それらを定義する際に設定するパラメータの中で特に重要なのはファジィ型の分割数である．分割数はファジィ集合の精度に深く関わっている．メンバーシップ関数を定義するときに分割数の値にほぼ比例してスタックメモリを大きく消費する．

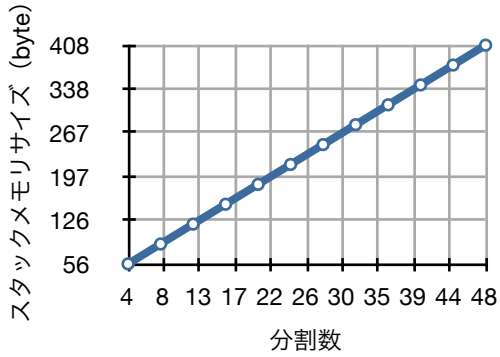

図 6 分割数と消費するスタックメモリサイズ

次にヒープメモリサイズについて考察する．ファジィ推論をするだけではヒープメモリを消費することはない．ヒープメモリはメンバーシップ関数のキャストを行うときに消費する．

3.3. 計算時間

計算時間について考察する．推論ルールの構築では，ファジィ集合を利用して推論を行う．その推論ルールではメンバーシップ関数同士の演算も行うことが出来る．しかし，メンバーシップ関数同士の演算は浮動小数点同士の演算になり，組込みCPUの処理能力では大きな負荷となってしまう．つまり，推論ルールは計算時間に多大な影響を与える．従って，組込みソフトウェア開発では計算時間が増大するようなメンバーシップ関数同士の演算を行うようなルールを利用しないことが望ましい．

しかし，推論ルールの江健部ではメンバーシップ関数同士の演算が必須となる．つまり，推論ルールの後件部の工数に比例して計算時間がかかる．よって，推論ルールを削減すれば，その分計算時間が軽減される．また，ファジィ型の分割数がメンバーシップ関数同士の計算の演算回数に関係している．分割数が大きいほど演算回数が多くなり計算時間も増大する．計算時間を抑えるためにも，分割数は最低限の値にしたほうがよい．ルール数を固定し，分割数を10～100まで10刻みで変化させた場合の分割数と実行時間の関係を図7に示す．

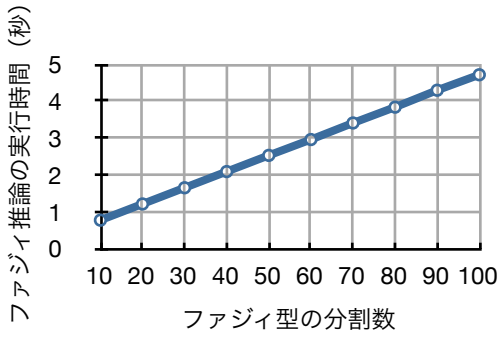

図 7 分割数と実行時間

3.4. 組込みソフトウェア向けファジィ推論構築

FDLによる組込みソフトウェア開発ガイドラインにFDLを用いてファジィ推論をするときに配慮すべきことをまとめた．開発ガイドラインに沿って開発をすることで，スタックサイズと計算時間を改善することが出来る．

FDLを用いてファジィ推論を構築しても組込みソフトウェアの要件から，リソース量が大きすぎたり，処理時間が長すぎる場合がある．問題に合わせた対処法を図8に示す．

問題	対応策	効果	影響
スタックメモリサイズオーバー	ファジィ型の分割数の変更	スタックメモリサイズの抑制，処理時間の高速化	推論精度
処理時間			
	推論ルールの変更	処理時間の高速化	推論結果（内容）

図 8 問題に対する対処法と，その効果と影響

図8にある対処を行えば，問題を容易に解決することが出来る．しかし，推論ルールの変更や分割数を変更することで推論精度や推論結果そのものに影響があることも注意する必要がある．

4. ファジィ推論の構築実験

提案したFDLによる組込みソフトウェア開発ガイドラインに沿って，組込み向けファジィ推論を構築し，組込みソフトウェアの要件を満たしつつ，ファジィ推論を実装することができるかを検証する．また，LUTによるファジィ推論構築のと比べて，開発効率が向上するかを検証する．

4.1. 自走式倒立振子のファジィ制御の構築

本研究室ではロバスト性に優れたバランス制御に関する研究として，ファジィ制御を用いた自走式倒立振子の開発を行なっている．自走式倒立振子では装置本体の倒立を保つように姿勢制御を行う．自走式倒立振子の姿勢制御はLUTを用いたファジィ推論で制御実験を行っていたが，今回初めてFDLによるファジィ推論を実装して制御実験を行った．

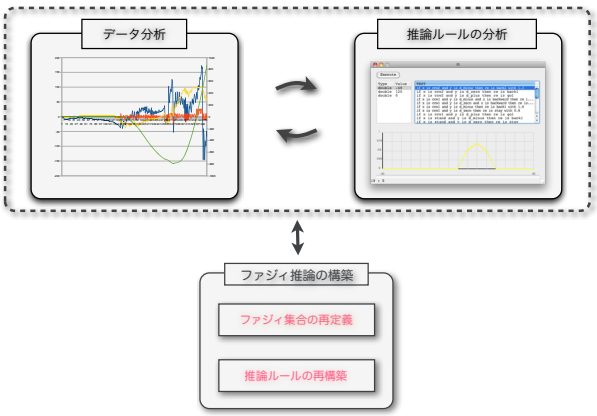

図 9 自走式倒立振子のファジィ推論のチューニング

2.3で述べたように，FDLによるファジィ推論の実装はチューニング作業時間を短縮することが出来る．よって，FDLによるファジィ推論の実装のほうが少ない時間でより多くのチューニング回数をこなすことができるので，LUTによるファジィ推論の実装よりも短時間で高精度のファジィ推論の構築が可能である．チューニングの際は提案した開発ガイドラインにより，効率的な開発が行えていた．

また，本実験により，組込みソフトウェアの要件を満たしつつ，FDLによるファジィ推論の実装が可能であることも実証できた．

表 10 チューニング作業時間の比較

チューニング	FDLによるファジィ推論 チューニングに要した時間（分）	LUTによるファジィ推論 チューニングに要した時間（分）
1回目	10	21
2回目	4	16
3回目	20	16
4回目	8	10
5回目	12	21
平均	**10.8**	**16.8**

5. まとめ

本研究では，組込みソフトウェアでファジィ推論を実装する手段としてFDLによるファジィ制御の実装を提案した．そして，FDLで開発する際に組込みソフトウェアの要件を満たすために，FDL処理系とFDLライブラリ関数の改良を行い，FDLによる組込みソフトウェア開発ガイドラインを提案した．ガイドラインに沿ってファジィ推論の構築行うと，効率的にファジィ推論を実装できることを示した．

今後の研究方針は，FDLによる組込み向けファジィ推論構築事例の充実と，FDLとLUTの連携について検討する必要がある．また，近年では全てのCPUがマルチコア化されている．従って，マルチコアに対応したFDL言語仕様の検討，FDL処理系の開発が挙げられる．

参考文献

[1] "ファジィシステム記述言語(FDL)標準仕様解説書"，社団法人 日本電子機械 工業会(EIAJ)編，1996
[2] 鑓田威，石畑清: "ファジィシステム記述言語の試作処理系"，11th Fuzzy System Symposium，pp. 23-26，Jul 1995
[3] 石畑清，森雄一郎，大塚和彦，新沢洋太，向殿政男: "ファジィシステム記述言語の 標準化に向けて"，11th Fuzzy System Symposium，pp. 27-30，Jul 1995
[4] 大塚和彦，森雄一郎，向殿政男: "ファジィシステム記述言語の標準化に向けて2"，14th Fuzzy System Symposium，pp. 291-292，Jun 1998
[5] 多和田侑，辻由彦，森雄一郎: "FDLの現状と今後の方向性",平成20年度日本知 能情報ファジィ学会中国・四国支部講演論文集，pp. 49-50
[6] 多和田侑，森雄一郎: "組込み系に対応したFDL言語仕様と開発ガイドライン"，26th Fuzzy System Symposium，pp. 413-416，Sep 2010