講座

ロボット工学ソフトウェア利用の手引 (第4回)

MacSLIM: JIS 規格ロボット言語 SLIM の教育用システム

MacSLIM: JIS-based Robot Programming System for Educational Use

松元 明弘* *東洋大学

Akihiro Matsumoto* *Toyo University

1. はじめに

SLIM とは Standard Language for Industrial Manipulators の略であり, 1992 年に日本工業規格 JIS B 8439 に定められた産業用ロボット用のロボット言語[1]である。筆者はその規格の作成段階から関与したことから, 規格の実証と教育への適用を目的として, 規格制定に先立って SLIM 言語システム MacSLIM[2]を試作した。これを日本ロボット学会誌 Vol. 14, No. 1 付録の CD-ROM 版ソフトウェアライブラリに投稿したのだが, 今回その解説を講座として書くよう指名されて大変光栄に思っている。以下に, まず規格が制定されるまでの経緯と基本的な設計方針を述べ, 次に教育用に試作された MacSLIM について説明する。

2. ロボット言語 SLIM

2.1 規格の制定の経緯と普及状況

「ロボット言語」と聞くと今さらという感もなくはない。今はすでにテキストベースのプログラミングから, GUI 技術を使ったビジュアルなプログラミングへと進化しつつあるからである。ただその場合でも内部では何らかの言語が使われていることが多い。したがってロボット言語の重要性がなくなったわけではない。普及が進んで技術的に成熟しただけである。ただし現実にはまだまだ問題を抱えている[3]。

標準化活動の詳細と SLIM 規格の設計指針は文献[3][4]などに譲るが, 地道な活動の成果として, 1992 年に世界で最初のロボット言語の規格が日本で制定されたことは意義あることである。しかし規格が制定されたとはいえ, その普及は容易なことではない。規格が制定されるまでの時間がかかり過ぎたことにより時代が変わってしまったとの指摘は確かに正しい。一方, 規格制定の精神の誤解

に基づく批判もある (例えば文献[5]の中に述べられている)。規格はあくまでも骨格であって実際のソフトウェア製品では規格が拡張されて実装されることが前提となっているのである。とすると各ロボットメーカーから SLIM 準拠の言語システムが発表されればすむのであるが, 世界的な経済不況の時期に重なり実装への投資が遅れた。現在は数社から SLIM 準拠の言語システムが発表されている。ここで説明する MacSLIM のほか, 規格の解説書[6]が発行され, また新しい試みとして SLIM を採用してのセル制御言語の GUI 環境での実現に関する研究事例[7]もあり, 規格の普及は徐々に進んできている。

2.2 SLIM の文法

SLIM の文法については, ここではスペースの制約からキーワード (図1) をに示すに留める。規格表[1]自身または詳しい解説は文献[6]を参照されたい。プログラム例はスペースの制約から図3の中にあるピック&プレイスの例を参照していただきたい (ただしポーズデータの定義部分が表示されていないことに注意)。SLIM は 1980 年代初頭に各社から発表されたロボット言語と同様に BASIC 風の文法構造を持っている。これは既存言語から標準言語への

Arithmetic operators:																											
+	-	*	/	MOD	AND	OR	XOR	NOT																			
Relational expressions:																											
=	<	>	<=	=<	>=	=>	<>	><																			
Declarations:																											
DEF	DEFIO	DEFFOS	DIM		HAND																						
Assignments:																											
LET	HOME																										
Pose variable handling:																											
LETX	LETY	LETZ	POSX	POSY	POSZ																						
Program flow controls:																											
GOTO	GOSUB	RETURN	IF		THEN	ELSE	FOR		NEXT																		
ON	GOTO	STOP	END																								
Robot controls:																											
(Motion controls)				MOVE				ROTATE				DRIVE															
(End effector controls)				RELEASE				GRASP				CHANGE															
(I/O port controls)				IN				OUT				SET															
(Speed controls)				SPEED				JSPEED				ACCEL															
(Timer controls)				DELAY				WAIT																			
(Termination controls)				HOLD				HALT																			
(Configuration controls)				LEFTY				RIGHTY				ABOVE															
												BELOW															
												FLIP															
												NONFLIP															
Input and Output:																											
PRINT INPUT																											
Basic arithmetic functions:																											
SQR	SIN	COS	TAN	ATN	ATN2	ABS	MIN	MAX	ORD	CHRS																	
VAL	STR\$	BINS	HEXS	MIRRORS	LEFT\$	MIDS	RIGHT\$	STRPOS																			
DIST	DEGRAD	RADDEG																									

図1 SLIM のキーワード

原稿受付 1996 年 8 月 26 日

キーワード: Robot, Programming, Standard, Education

*〒350 埼玉県川越市鯨井 2100

*Kujirai 2100, Kawagoe-shi, Saitama

乗り換えをスムーズにすることを期待しての設計方針による。言語の利用者の実態調査から、言語としての高級さよりも目的の動作の指定に十分な機能を有しているかどうかが重要であることがわかったからである。

3. MacSLIM のシステム構成

MacSLIM は、アップルコンピュータ製 Macintosh と三菱電機製ムーブマスター RV-M1 を専用のシリアルケーブルを自作して接続することにより稼働する(図2)。前述のように、規格制定に先立っての実装であり、教育現場での利用を意識して設計した。ムーブマスターは独自のコマンドシステムをもっているため、移植性を重視してこれを SLIM から見た中間コードとみなして利用した。世界中どこにでも入手でき GUI 環境の整っていた計算機と教育機関にはよく普及していたロボットを組み合わせただけで、開発当時としては最適な選択肢であった。MacSLIM は MacSE と 漢字 Talk 6.0.7, MacSE/30 と 漢字 Talk 7.1 の組み合わせでそれぞれ動作確認をしている。なお、Macintosh 側には RV-M1 の機構データを組み込んであるため、この処理系は RV-M1 専用であり、MacSLIM-RVM1 が正式名称である。他のムーブマスターシリーズに対応するには、機構データの変更とムーブマスターコマンドの拡張に対する処理が必要である。

SLIM の規格票[1]によれば、各処理系はそれが行なった処理系定義部分と拡張部分を説明書に明記する必要がある。以下にそれを説明するが、これを厳密に解釈すると、MacSLIM-RVM1 は、規格に完全に合致しているとは言えない。この点は注意して使用する必要がある。

図2 システム構成

(1) 処理系定義事項

- 使用できる文字は、規格で規定されているものに加えて、英字として小文字も使用可能である。大文字と小文字は区別する。ただし、予約語に関しては大文字で入力しても小文字で入力しても構わない。内部コードは ASCII を使用している。日本語は使用できない。
- 1 行に書くことのできる文字数は、128 文字である。また行末記号は、Macintosh に合わせて、CR (キャリッジリターン) のみである。
- 識別子の文字数は、31 文字まで許されるが、識別するのは 16 文字までである。
- ポーズ定数における X, Y, Z の座標値は、[mm] 単位で指定し、またピッチ角、ロール角は、「度」により指定する。これはムーブマスターの表現形式をそのまま利用するためである。これ以外の角度表現については、標準的な単位すなわち「ラジアン」を用いる。例えば、PRINT 文でのポーズ定数の出力の場合にはラジアン単位で出力される。
- DRIVE 文における回転角指定は、「ラジアン」により指定する。
- ポーズ定数に含まれる要素の数は 5 個以内である。
- 番号付きポーズ変数におけるポーズ番号の上限は、現在 64 である。
- 文字列変数に結び付けられた文字列の最大長は、128 文字である。
- 実数の精度は IEEE-754 形式に一致させた。すなわち、単精度実数は、符号 1 ビット、指数 8 ビット、仮数 23 ビットの計 32 ビット、倍精度実数は、符号 1 ビット、指数 11 ビット、仮数 52 ビットの計 64 ビットである。
- ハンドパラメタの意味については、ムーブマスターの標準ハンドに合わせた。

(2) システムの制約上実現できない機能

Macintosh からムーブマスターへはムーブマスターのコマンドをそのまま利用しているため、このコマンド体系に含まれていない機能は実現できない。また、ムーブマスターの機構的制約から実現が不可能なものもある。それらは以下に列挙する通りである。

- ロボット動作中の入出力処理 (IN 文, OUT 文, SET 文, RESET 文, IOBLOCK 文)
- MOVE 文における続行オプション
- 腕形態 (LEFTY), 肘形態 (BELOW), 手首形態 (NON-FLIP)

(3) サポートされていない機能

現状では、以下の機能がサポートされていない。

- 角度値 (DEG 付き角度) の取り扱い
- DRIVE 文における負の角度の指定
- 引数のない関数宣言

- ・ポーズ番号式を用いたポーズ変数宣言
- ・ポーズ成分代入 (LETX, LETY, LETZ) およびポーズ成分抽出 (POSX, POSY, POSZ)
- ・フロー制御文 GOSUB-RETURN, ON-GOTO と、飛先を直接指定する IF 文
- ・行番号による、行の昇順へのソート (エディタ上)

4. MacSLIM-RVM 1 の使用方法

4.1 プログラミング環境

MacSLIM-RVM 1 のプログラムを起動すると、図 3 のような画面が現れる。ここでプログラムの作成、修正、ティーチングを含むロボットとの対話処理ができる。そのため単なる処理系ではなく、統合プログラミング環境であると言える。そのユーザインタフェース部分は

- ・テキストエディタウィンドウ (Untitled と表示)、
- ・コマンドライン入力ウィンドウ (Command と表示)、
- ・出力用ウィンドウ (Output と表示)、
- ・ティーチングウィンドウ (Teaching Box と表示)

の 4 種類のウィンドウから構成される。なお、ティーチングウィンドウは初期状態ではオープンされない。

テキストエディタウィンドウは、SLIM プログラムを入力し編集するためのウィンドウである。ウィンドウ名としては、新規プログラムならば Untitled と表示され、ファイルから呼び出した場合にはそのファイル名が表示される。現在のところ、テキストエディタウィンドウは、同時には 1 つしか開けない。マウス操作による縦横方向のスクロールやカーソル移動、ファイル保存、印刷が可能である。

コマンドライン入力ウィンドウは SLIM のステートメントを直接実行するためのウィンドウである。インタプリタの形式の採用により、このウィンドウではリターンキーが入力されるまでの 1 行の SLIM ステートメントが即実行される。ここで対話的に命令を実行でき、実行結果はムーブマスターが動作するか出力用ウィンドウに出力される

ので、確認後入力文が有効であればテキストエディタ・ウィンドウにコピー&ペーストしてプログラムを書くことが可能である。

出力用ウィンドウはディスプレイ上に出力する際の出力ウィンドウである。ロボット関係の命令以外 (例えば PRINT 文) の出力は、このウィンドウに出力される。

ティーチングウィンドウは、ムーブマスターに標準装備されているティーチングボックスの代替の役割を果たす。画面上のスクロールバーをマウスでクリックあるいはドラッグすることで、関節角で指定した位置へムーブマスターの手先を対話的に動かすことができる。マウスボタンを離すとすぐに、指定した位置にムーブマスターが動作する。ムーブマスターが動作中は、各ボタンやスクロールバーは入力抑制のためグレー表示となる。ムーブマスターが停止すると、そのときのムーブマスターの位置・姿勢データを Run メニューの中の Get Pose によりカレントウィンドウに読み込むことができる。各ボタンの機能は次の通りである。

Nesting ムーブマスターをネストの状態にする (機械的原点合わせをする)。

Grasp ハンドを閉じる。ボタンは Release に変わる。

Release ハンドを開く。ボタンは Grasp に変わる。

Reset ドライブユニットをリセットする。

各スクロールバーの Waist, Shoulder, Elbow, Pitch, Roll は、それぞれムーブマスターの第 1 軸, 第 2 軸, 第 3 軸, 第 4 軸, 第 5 軸を操作する。なおスクロールバーの左端が動作範囲の最小値, 右端が最大値を表す。スクロールバーをマウスでスライドさせるとグロスモーション (5 度単位), 矢印をクリックするとファインモーション (1 度単位) で動作する。

4.2 プログラム作成手順

テキストエディタウィンドウに SLIM の文法にしたがって入力し、Run メニューの Run Program により実行するというのが基本である。しかしながら点の位置データを事前に正確に知ることは困難である。想定した場所になくて少しずれていることはしばしば起こる。この場合現物の位置を取り込んで利用すれば良い。このためにティーチング機能が存在する。

Window メニューから Teaching を選択して、まず最初に 1 回だけ Nesting ボタンにより機械的原点合わせを行なう。ロボットがいかなる位置にいても強制的に機械的原点に復帰するので、ロボットの動作範囲内に障害物の多い環境では機械的原点に近いところにロボットをティーチングで動かしてから Nesting を行なうようにすべきである。原点合わせができたら、望みの場所にムーブマスターを動かし、Run メニューの Get Pose によりムーブマスターの現在位置をカレントウィンドウに読み込む。通常、こ

図 3 プログラミング環境

の作業の時にはテキストエディタウィンドウに読み込む。
すなわち、点（ポーズ）の座標を手で入力する代わりに実座標値を読み込んで貼り付けるのである。

上記のようにティーチング機能とプログラム機能を組み合わせることによって、対話的に応用プログラムを開発することができる。これはグラフィックシミュレータのない環境にあっては、最適の方法と言える。

5. システムの評価[8][9]

システムの評価のために、ロボット工学研究室に配属されたばかりの4年生を対象として実習をしてもらった。内容はロボットの挙動の理解とロボット言語の概要の理解であり、具体的には、動作命令文の MOVE 文（直角座標指定）と DRIVE 文（関節座標指定）を実行させて順変換・逆変換を理解すること、MOVE 文の3つの補間方式（Pose-To-Pose, 直線補間, 円弧補間）の違いを理解することといった基礎的な事項である。教育の観点からは、授業で習ったとはいえ理解度が十分とはいえない学生からの意見は貴重である。それらの代表的なものを列挙すると、

- ・プログラム入力やティーチングなどのモードの切り替えの容易さの点でウィンドウ環境は使いやすい。
- ・複数のファイルを同時に開けないのは不便である。
- ・ロボットの関節角では直感的にはわかりにくいので、ポーズデータを直角座標で読み込む機能は便利である。
- ・ロボットのポーズデータは常に表示されているべき。
- ・正確に構文を覚えていなくてもプログラム作成ができるようなエディタ機能の充実が望まれる。
- ・グラフィックシミュレータがあるとよい。
- ・SLIM 言語仕様としてなぜ BASIC 風なのか。もしそうだとしたら行番号はなくてもよいのではないか。
- ・ロボットと計算機間の通信は同じことをやってもうまくいくときといかないときがある。

これによれば、とりあえずロボットおよびロボット言語の初期学習に耐えることがわかり、教育用途という当初の目的は達成されているようである。システム実装における問題点もいくつかわかっているため、それらの情報は CD-ROM 付属のドキュメントを参照してほしい。

6. おわりに

ロボット言語 SLIM の規格制定の概要と普及状況を簡単に説明し、教育用を狙った試作 MacSLIM の概要と教育への適用例を説明した。本稿が産業用ロボットのプログラミング方式の普及や進歩に役立つことを期待している。

謝 辞

規格の内容は日本ロボット工業会内に開かれた委員会で審議し、規格原案はその中のワーキンググループで作成しました。委員長は東京大学新井民夫教授のほか、委員の方々には長期にわたって大変ご協力いただきました。委員の中でも特に、石井博氏（小野技術研究所）、毛利峻治氏（日立製作所）、森俊二氏（三菱電機）、斉藤和春氏（元、セイコーエプソン）に感謝いたします。

また、MacSLIM のシステム作成にあたって、Macintosh 上でのプログラミングをしてくれた安藤真敏氏（本学卒業生）およびドキュメントの作成と規格整合性のチェックをしてくれた横山雅一氏（本学卒業生）に感謝します。

参 考 文 献

- [1] 日本工業標準調査会：“産業用ロボットプログラム言語 SLIM (JIS B 8439)”，日本規格協会，1992。
- [2] 松元，安藤：“ロボット言語 SLIM のためのプログラミング環境”，日本機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH '92) 講演論文集，pp. 501-504，Jun. 1992。
- [3] 松元，新井：“ロボット言語の現状と動向”，日本ロボット学会誌，vol. 11，no. 1，pp. 36-39，Jan. 1993。
- [4] 松元：“JIS 規格ロボット言語 SLIM の設計”，設計工学（日本設計工学会誌），vol. 30，no. 2，pp. 53-59，1995。
- [5] 原口：“普及するロボット標準言語”，日経メカニカル，1994 年 3 月 7 日号，pp. 40-46，1994。
- [6] 新井（編）：産業用ロボット言語“SLIM”。日本規格協会，東京，1994。
- [7] 小山，三宅：“FA セル制御言語の開発（第1報：セル制御言語によるユーザプログラミングの概要）”，第13回日本ロボット学会学術講演会論文集，pp. 1053-1054，1995。
- [8] A. Matsumoto and M. Ando：“Interactive Robot Programming System for Educational Use,” Proc. IEEE International Workshop on Robot and Human Communication (ROMAN '92)，pp. 419-424，Sep. 1992。
- [9] 安藤，松元：“ロボット言語 SLIM によるプログラミング教育”，第10回日本ロボット学会学術講演会講演論文集，pp. 1117-1118，Oct. - Nov. 1992。

松元明弘 (Akihiro Matsumoto)

1958 年 6 月 8 日生。1983 年東京大学大学院工学系研究科精密機械工学専攻修士課程修了。1983 年東京大学工学部助手。1988 年東洋大学工学部機械工学科講師，1990 年同大学助教授，現在に至る。工学博士。1994-1995 年ルイバスツール・ストラスブール第1大学(フランス)訪問研究員。ロボット言語，自律分散型ロボットシステムの研究に従事。精密工学会，日本機械学会，IEEE，日本ロボット工業会の会員。

(日本ロボット学会正会員)