プロダクションシステムによるロボット作業教示と実行システム

5F−8

高橋友一　大原秀一
NTTヒューマンインタフェース研究所

1。はじめに

ロボットに作業を指示する代表的な方法としては、ティーチングプレイバック方法とロボット言語を用いたプログラミングによる方法がある。井上はロボット言語をロボットの基本動作に対応するコマンドレベルから、ロボットの手の動きを規定する動作レベル、作業対象物の状態変化について記述する対象物レベル、目標状態により指示する作業レベルの言語に分類した[1]。高次レベルのロボット言語で実際にロボットを動かして作業を実行するには、プログラミングされた内容を作業環境にあわせながらコマンドレベルに翻訳する必要がある。対象物レベルのロボット言語としてはAUTPASSやRAPTが発表されているが、実際のロボットを動かすには環境計測など解決すべき点が多い。

ロボットの教示方法が、動作手順（コマンドシーケンス）と作業環境（オペランド）を人間に指示しやすい形で入力できる高次レベルになる程、教示された内容を解釈し不十分な情報を補足する機能が必要になる。本稿では、入力コマンドを解釈する知識をif - then 型のルールで教示し、センサからのデータをもとに実際の作業環境下で教示された内容に従いロボットに作業を行なわせる教示ー実行システムについて述べる。

2。システムの構成

ロボット動作の教示には、ロボットの動きと作業環境の2種類の教示がある。同じ作業でも、部品の位置や姿勢などの作業環境が変化すると動作レベルの言語ではプログラムの書き換えとなる。一方、同じような作業は共通に利用できる事が望ましい。想定された作業環境下における動作手順の教示および実作業環境下での実行について述べる。

2。1　ルールによる動作教示

if - then 型のルールを用いたプロダクションシステムは、モジュール性がよく人間の思考方法にあった知識表現方法である。プロダクションシステムを用いて、作業レベルの指示から動作計画の自動生成をする試みもなされている[2]。ここでは、動作計画は与えられている前提の下に、一連のまとまった複数の動作を一つの動作として定義する。作業環境に応じてたセンサデータの利用方法の規定などのロボット動作の教示に、if - then 型のルールの利用を考え

```
(defrule disk1
  (pattern (disk ?A from pole ?B to pole ?C))
  (into   (move_to_pole ?B)
          (grasp ?A on pole ?B)
          (move_to_pole ?C)
          (release ?A on pole ?C)))

(defrule grasp
  (pattern (grasp ?A on pole ?B))
  (into   (move_along_down ?B)
          (GC)
          (move_along_up ?B))

(defrule move_to_pole
  (pattern (move_to_pole ?A))
  (into   (MP ?X))
  (where  (setq ?X ![?A position-get])))
```

注；GC, MP: ロボットコマンド

図.1 ルールによる動作教示

```
(defclass pole () (color position ) () :gettable :settable)

(!red (make-instance 'pole color 'red position '() ))
(!green (make-instance 'pole color 'green position '() ))
(!blue (make-instance 'pole color 'blue position '() ))

(defmethod (pole position-get) (pole-name)
 (!envdata (svision))  ;画像処理装置のプロセス起動
 (!env-str (make-string-input-stream envdata))
 (!env-list (read env-str))
  (!pole-no ())
  (cond
   ((equal pole-name red) (!pole-no 1))
   ((equal pole-name green) (!pole-no 2))
   ((equal pole-name blue) (!pole-no 3)))
 (!position (first (cadr (nth pole-no env-list)))))
```

図.2 仮想作業環境の定義

Robot teaching System based on Production System
Tomoichi Takahashi, Shuichi Ohara
NTT Human Interface Laboratories

る。図. 1 にdisk をpole からpole 移す作業をmove_to_pole, grasp, release の３つの動作で、grasp とmove_to_pole を作業に応じた手順で定義した例を示す。

２。２　オブジェクトによる仮想作業環境の定義

動作レベルのプログラム言語では、プログラミングの大半を作業環境に関連した動作点などの記述が占めている[3]。仮想的な作業環境は構成している部品の種類、個数やその属性で記述される。図.2に構成要素をオブジェクトとして、属性をメソッドにより定義し作業環境を記述した例を示す。

２。３　センサ情報に基づく実行

指示された内容を教示されたルールを前向きに照合することで、ロボットコマンドシーケンスに解釈する。ルールの解釈時に、動作の対象となるオブジェクトにメッセージ伝達することで部品位置など仮想作業環境では記号で表現されている情報に対する実際の作業環境における座標値を得る。図.1のmove_to_pole 動作においては必要となるpole の位置座標を対象pole オブジェクトにメッセージ：position_get を送り得られる座標位置をロボットコマンドMPの実行時のオペランドとする。

２。４　実験システム概要

ロボットとして力覚センサつきのMOVEMASTER™、画像処理装置としてTospix™、仮想環境の定義やルールにより指示をロボットの言語に解釈するプログラムをELIS™を用いてシステムを構築した。システムの概要を図.3に示す。力覚センサからのデータを用いてdiskをpoleに挿入する運動モジュールはサブルーチンとしてロボットコントローラに用意されている[4]。

ロボット作業としては、３本のpoleを用いてdiskを移し換える作業を取り上げた。図.4にロボットに指示した対象レベルの作業指示の内容を示す。pole の識別が容易できるように先端が色付けした実験では、作業途中においてpoleの位置が変わる状況においても、教示内容を変更することなくdiskの移し換え作業を行なうことができた。

３。おわりに

作業者が想定した作業環境でif - then 型ルールより対象レベルで作業のモジュール化や教示が容易にでき、教示内容に従いセンサのデータにより実作業環境が想定作業環境のなかで変化しても実行するシステムについて述べた。

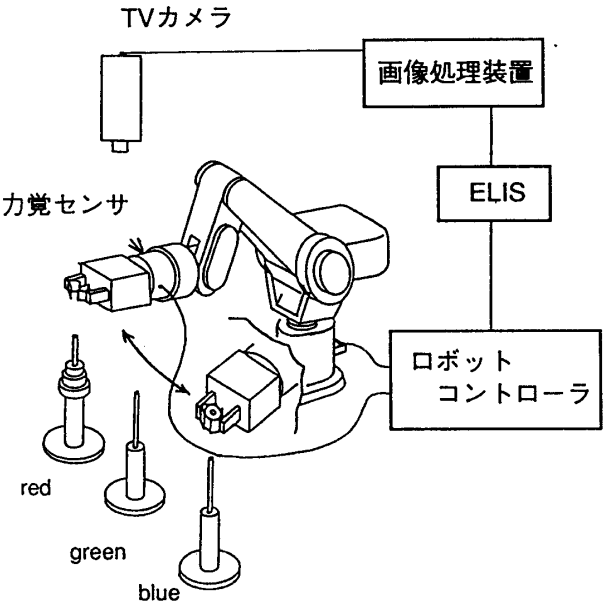

図.3　システム構成図

```
(defwork hanoi (disk d1 from pole red to pole green)
               (disk d2 from pole red to pole blue)
               (disk d1 from pole green to pole blue)
               (disk d3 from pole red to pole green)
               (disk d1 from pole blue to pole red)
               (disk d2 from pole blue to pole green)
               (disk d1 from pole red to pole green))
```

図.4 作業指示

参考文献

1) 井上；ロボット工学とその応用、電子通信学会編、辻、江尻監修　(1984)

2) 松原他；プロダクションシステムによるロボットの行動計画の自動作成、知識工学と人工知能 41-8 (1985)

3) P. D. Summers, D. D. Grossmann; XPROBE: An Experimental System for Programming Robots by Example, The Int. J. Robotics Research, Vol. 3, No.1 pp.25-39 (1984)

4) 大原他；センサベースの環境モデルを用いたロボットオペレーション、日本ロボット学会第８回学術講演会 2305 (1990)