KL1処理系によるロボット協調作業*

1N－7

溝口文雄†　大和田勇人†　岡本祐太郎†

東京理科大学 理工学部†

1　はじめに

ロボット制御に関して，複数台のロボットによる作業を対象にした研究がされている．本研究は，複数台のロボット（マニピュレータ）の協調作業を，計算機により中央集権的に管理することを想定し，その制御を，並列言語を用いてインプリメントすることを試みる．

従来の逐次型言語で制御した場合，ロボットに対し，1台ずつ順次命令を送ることになる．即ち，ロボットは命令を受けた順に動きだすことになり，同時に並行した動作はせず，ロボットの数が増えれば，動作のずれも大きくなると予想できる．各ロボットが独立に作業をする際は，その作業は同時進行するのが望ましく，また，密接な協調作業をするとすれば，なおかつ動作に時間的な差があってはならない．

従って，各ロボットに並行して作業を行なわせたいが，それには，各ロボットに並行して動作命令を送ればよい．その実現のために，ロボットの制御に並列言語を用いることが，1つの手段と考えられる．

2　KL1処理系の利用

KL1は並列論理型言語 Guarded Horn Clauses に基づいて設計された，並列処理の記述に適したプログラム言語である．

KL1処理系を用いた研究において，ロボットを制御し，実際に動かすというものは見られない．そこで，ロボットの作業を制御できれば，コンピュータ上での処理だけでなく，KL1処理系の新たな適応分野を示すことができる．複数台のロボットを扱うのに必要となる同期制御なども，KL1の特徴と一致しており，処理に適すると思われる．

なお，本研究では，並列言語 FGHC を Arity-Prolog 上に移植したものを使用した．

3　協調作業の設定

ロボットの協調作業とは，1台では実行不可能，あるいは複雑で大規模な作業を，複数のロボットが分担・協力して行なうことを意味している．この際生じる問題として，ロボットどうしの相互干渉，同期の取り方，作業の分担方法などがある．

本研究では，簡単な問題として，

- 位置的な相互干渉を防ぐため同期をとる．
- 次にすべき作業を，ロボットを選択して行なわせる．

といったことを含む作業を設定する．これに基づき，ブロック（積み木）の世界を対象とし，以下のような作業を設定する．

2台のロボットをA，B，ブロックを置く位置をp，q，r，sとし，それらを図1のように配置する．

p，rに積まれているブロックをqに運ぶ作業を行なわせ，外部の要求により，sにブロックを運び出すことを設定する．

q，sには双方のロボットとも手が届くが，p，rはそれぞれのロボットにしか届かない．即ち，p，q間とq，r間は，それぞれのロボットのみ作業が可能な部分，r，s間は双方とも作業が可能な部分である．

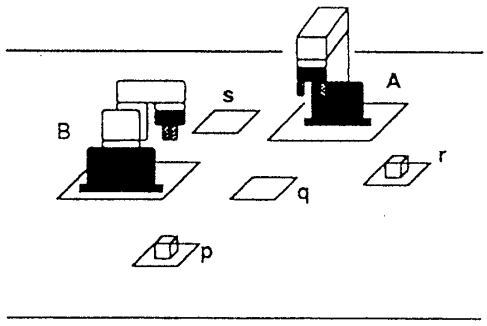

図1: ロボットとブロックの配置

まず，ロボットAはrからq，ロボットBはpからqへブロックを運ぶ作業を繰り返す．そして，外部から作業の変更の要求があった場合などの状況に合わせ，その作業を変更する．

外部からのロボットへの要求とは，ブロックのsへの移動などである．

4　システム構成

2台のロボットは，それぞれプリンター回線により計算機(PC-9801)に接続されており，計算機2台はRS-232C回線により接続される．計算機の1台をホストとし，作業全体の流れを管理する．ホスト計算機側は，ロボットAへの命令，ロボットB側の計算機との通信，外部からの入力を受けつけるモジュールを持つ．B側の計算機はA側と通信を取りながら，ロボットBに命令を送信する．

5　実行の流れ

ホスト計算機における，プログラムの主要な部分，および，その実行の流れ（図2）を以下に示す．

```
start:- true | stream(prn,rw,R),
               stream(aux,rw,X),
               stream(tty,rw,U),
               process(R,X,U,... ).
process(...):- true | douki(FLA,FLB,A,B),
                      proc_a(A,...),
                      proc_b(B,...),
                      inkey(...).
douki(wait,wait,A,B):- true | A=ok,B=ok.
douki(ok,FL,A,B):- true | A=ok,B=FL.
```

*Cooperative operation with KL1 processing system
†Fumio MIZOGUCHI, Hayato OHWADA, Yutaro OKAMOTO
†Faculty of Sci. and Tech. Science University of Tokyo

```
douki(FL,ok,A,B):- true | A=FL,B=ok.

proc_a(ok,...):- true | rocom(C,...).
proc_b(ok,...):- ...

rocom([]   ,...):- true | wchk(CF,CN...).
rocom([H|T],...):- true | cchk(H,...).

wchk(0,1,BR,BQ,BS,...):- (ブロックの数) |
                         CN:=... ,comm(CN,C),
                         process(...,CN,C).
cchk(wa,...):- true | process(...,wait,...).
cchk(H,...):- H\=wa | R=[write(H),nl|R1],
                      process(R1,...).
comm(CN,C) :- true  | C=['mp -300,...].


inkey(...,U,...):- true | U=[read(Iu)|U1],
                          u_com(Iu,U1,...).
u_com(s,U1,...):- true  | cnchk(s,CN,CF),
                  process(...,U1,...,CF...).
cnchk(s,CN,CF):-  (CN チェック) | CF=....
```

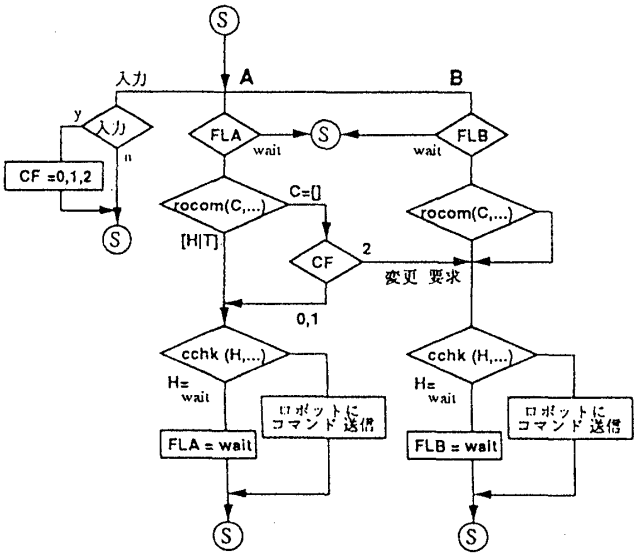

図 2: 実行の流れ

デバイスに対する入出力は，述語streamにより，デバイスごとに割当てられた変数に命令を送ることで行なう．ロボット A，B，外部入力に対応するデバイスはそれぞれ，プリンター回線，RS-232C，キーボードであり，変数R，X，Uを割当てている．

処理の流れは述語processから始まり，それを再帰的に呼び出すことで，実行が繰り返される．processからはロボット A，B，外部入力の３つの処理に分岐する．

ロボットの動作のためのコマンドは，全体の作業を区切り，コマンドリストにしておき，リストから１つずつ送る．リストは，述語comm(CN,C)のリスト番号(CN)を呼べば得られる．リストからコマンドを送り，リストが空になったら次のリストを選択する．選択は，作業変更のフラグ(CF)，作業場のブロックの数を条件に行なう(wchk)．

また，B側計算機のプログラムは，上記のロボット B の処理に分岐した後，先頭に戻るまでの部分である．ロボット A からの通信を受け，その際，作業変更の指示，ブロックの数などの情報を受けとり，ロボットにコマンドを送る処理をする．処理の完了後，B側計算機は A 側計算機に情報を返す．

5.1 ロボットの同期制御

前述の３つの処理は，疑似的に並列に実行される．ここで，ロボット A，B の干渉の回避のために，A，B のモジュール間で同期制御をする必要がある．そのために，各ロボットへの命令を送っていく際，同期を取る部分，即ち，相手のロボットの動作を待つ必要があるところで，そのロボットへの処理を断つようにする．これにより，次に続く処理は必然的にもう一方のロボットの処理をすることになる．

プログラムでは，各処理の先頭で変数（FLA,FLB）を調べ，FLA(FLB)=waitならば，その処理には分岐しないようにする．また，両方の変数がwaitになった際は，それを解除(okに)する(douki)．

同期を取る部分はコマンドwaitとしておき，リストからwaitが現れたら，FLAにwaitを代入し，処理の先頭に戻る(cchk)．

5.2 外部入力と作業の割り当て

作業中，その次の作業をするロボットを，状況により決定するようにする．例えばsにブロックを運ぶことが外部より要求された場合，ロボット A，Bのうち，sに移動しやすいほう，即ち，qに移動中のロボットが，その作業を次に行なうこととする．

外部入力は述語inkeyにおいて，受付け，その内容により，処理を分岐させる(u_com)．作業変更の要求であれば，現在の動作番号により，そのロボットと他方のロボットのいずれがその作業につくか判断し，変数（CF）に値を代入する（述語cnchk)．

作業の変更は，コマンドリストを選択するとき(wchk)，変数（CF）の値により，行なう．

6 まとめ

本研究では，ロボットの協調作業の制御を，並列言語 KL1 処理系によりインプリメントし，実際にロボットを動かした．今後，再帰的処理と同期制御を組合せかた，適切な処理の割り付け方法をを検討したい．それにより，３台以上のロボットや，複雑な作業に対する制御を考えていく．

参考文献

[1] 新世代コンピュータ開発機構 KLIC 講習会テキスト -KL1 言語編- ,ICOT,1993.

[2] Tomas Lozano-Pérez,HANDY (A Robot task planner). The MIT Prress ,1992