

インタラクティブなロボットプランニング

長田 正 塚本享治
(電子技術総合研究所)

1. まえがき

ロボットプランニングとは、与えられたゴールを実現するために必要な実行手順も自動的に作り出すプロセスをいう。従来、ロボットプランニングシステムとして開発されてきたものには、定理解明プログラムをベースにした STRIPS⁽¹⁾, ABSTRIPS⁽²⁾ や、人工知能用言語である PLANNER⁽³⁾, CONNIVER⁽⁴⁾ に基づく SHRDLU⁽⁵⁾, BUILD⁽⁶⁾, HACKER⁽⁷⁾ などがある。これらは、いずれもゴールオリエンティドな方法で、GPS の影響を強く受けているといえる。また、これらのプランニングシステムは、具体的にハードウェアを動かす実行システムとは一応分離して考えられており、ロボットシステム全体は、プランニングと実行に関し階層的な構造をなしている。

しかし、実際には、ロボットの作業環境や制約条件が前もつてすべて明らかであることは少なく、また、たとえ前にもすべてのことが知られていたとしても、実行中に予期せぬ事態が発生し、与えた知識を変更しなければならないことがしばしば生じる。さらに、複数のロボットが協同作業を行なうような場合には、ゴールオリエンティドなやり方のみでプランニングを進めることは、条件が複雑となり非常に困難を伴う恐れがある。

したがって、ロボットのプランニングは、固定された知識に基づくゴールオリエンティドな方法だけでなく、現時点において最も好ましいと思われる行動の選択・実行と外界からの反応による誤り訂正という前向きな試行錯誤的な方法をも何らかの形で取り入れることが望ましいと思われる。具体的には、ゴールオリエンティドな方法で大まかな方針を定め、細部に試行錯誤的な方法を加味する混合方式が望ましい。プランニングにおける前者の付きまとい「グローバルモード」、後者のものを「ローカルモード」とよぶことにする。ローカルモードでのプランニングの場合には、オペレータや実行部とのインタラクションは、欠くことのできない機能となる。

本論文は、上述のような考え方に基き、インタラクティブな機能も有するプランニングシステムの基本的な構成法と二・三の応用例について考察したものである。

2. インタラクティブなプランニングシステム

プランニングにおけるバックトラッキング、外界とのインタラクションおよび誤って実行したときの再プランニング (re-planning) の諸機能も容易に実現するために、プランニングはすべて状態を基準として行ない、実行部に移すときにはじめて行為の命令に変換することにする。プランニングは、計算機内で作られている環境のモデルに対する操作 (シミュレーション) という形で行なわれる。

2.1 システム構成の概要

プランニングシステムを中心としたロボットシステムの構成図は図1に示される。ただし、ここでは簡単のために、一台のロボットに限定して話を進めることにする。プランニング部は、プランを作る問題解決システム (PSS) とオペレ

ータや実行部と交信し、結果的には、システム全体の動きを制御する制御システム(CS)とから構成されている。PSSは、可能性を見出すシステム(PFS)とプランの遂行に伴う状態変化をモデル上にシミュレートするためのシミュレータ(SIM)とで構成される。また、PFSは、必要に応じてゴールオリエンティドな方法で大まかな道筋を定める部分(グローバルモード)と、限られた範囲内で前向きに可能性を見出す部分(ローカルモード)に分けられる。SIMは、モデル上での仮想ロボットの

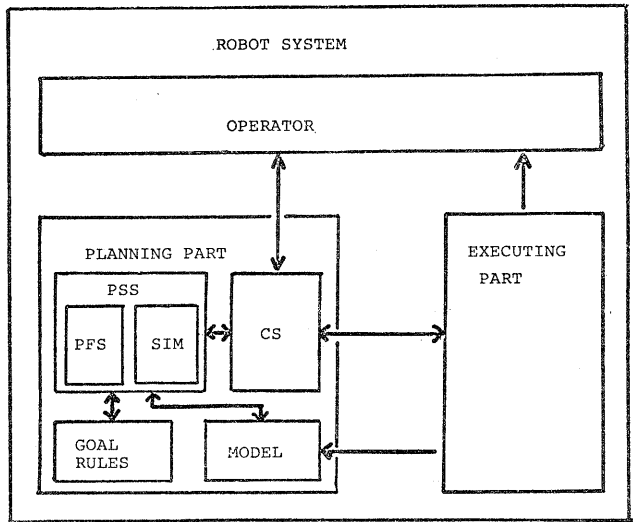


図 1

の行動を表わし、ローカルモードにおけるPFSで求められた幾つかの可能性が適用され、適当な評価の下でその中の一つが選ばれる。他の可能性は、バックトラッキングに備えてスタックに保存される。選ばれる可能性の状態は行為に変換され、実行システムからの要求に応じて送り出される。

実行システムは、プランニングが完了する前に動作することが可能である。ただし、実行システムが動作することは環境に実質的な変化を与えることになり、これに伴いモデルが変更され、PFSで求められた可能性のスタックは消去される。したがって、バックトラッキングは不可能となり、モデルの状態を現状として再プランニングが行われる。ただし、一度立てられた計画は、何らかの形で保存され、再プランニングの効率化のために利用することも可能である。なお、PSSが解を出す前には、実行システムを動作させるか否かは、与えられた目標や周囲の状況に依存して決められるべき問題で、オペレータの判断に任されている。

2.2 プランニングアルゴリズム

PSSで用いられるプランニングアルゴリズムの概要は次の通りである。ただし、プランはモデルの主体者である仮想ロボットの状態遷移で表われ、仮想ロボットの状態[*current state*]は、実ロボットの状態[*present state*]と区別して使用されている。

0. ロボットの *present state* をスタックする。
1. *current state* がゴール状態を含まば終了。
2. もし可能なら、サブゴールを設定する。特にサブゴールを設定しない場合には、ゴール状態をサブゴールとする。
3. *current state* がサブゴールを含まばステップ1へ。
4. 新たに適用できる候補のルールがなければ、*current state* をポップしてステップ7へ[バックトラッキング]、ポップできなければステップ0へ[再プランニング]。
5. *current state* に新たなルール R_j を適用し、遷移状態の可能性リスト $R_i = \{R_i, R_j \ (j=1, \dots, n)\}$ を作る。

* たとえば、ある行為の適用可能条件とそれを実行したときのモデルへの効果

6. もし評価基準が与えられれば, それに従って P_i を並べ替え, *current state* S_i に P_i を追加する.
 7. P_i が空ならステップ4へ, 空でない場合は, P_i の最初の要素 P_{ij} をとり出し, それを P_i からとり除く.
 8. P_{ij} をシミュレータに渡し, 次の決断点まで状態の遷移を行なう. 同時に, P_{ij} に対応する行為を派生させ, *current state* に付加しておく.
 9. 決断点に達すれば, その時の状態 S_{i+1} をスタックし, ステップ3へ.
- 複数台ロボットによる協同作業の場合にも, 以上で示したアルゴリズムを修正する事により, 試行錯誤的な方法に基づくプランニングも行なう事ができる.

2.3 プランの実行とインタラクション

プランの作成と実行を平行して進め, プランが少しでもできりゃ実行して外界からの情報を取り入れ, その後のプラン作りに反映させる事も可能にするためには, プランニング部で作成された結果の連鎖は, バックトラッキングに備えて LIFO の形で記憶され, FIFO の形で実行されなければならない. 具体的には, 図2に示すような *double ended queue* のメカニズムが適用されている.

プランニングにより求められた状態の連鎖と, それをロボットが実行可能な形に変換した行為の連鎖とをそれぞれ DEQ_p と DEQ_e に記憶する. DEQ_e からの要請により, DEQ_p の状態遷移の形で表わされたプランニングの結果は, 状態と行為の組の連鎖に変換されて DEQ_e に転送される. ロボットは, DEQ_e から一つづつとり出して順次実行する. u に応じて実行されると, それに対応する状態は DEQ_p より捨てられ, バックトラッキングの対象から除かれる. また, プランニングの途中でバックトラックの必要が生じた場合には, DEQ_p の最新値(左端)から順次バックトラックし, 新たな選択が決められると, それ以後のプランは更新され, 対応する DEQ_e の内容も変えられる. DEQ_p の右端まで進んでもバックトラックが完了しない場合には, 再プランニングが行なわれる. 図2のメカニズムは, 少し修正すると, オペレータとプランニング部との間のインタラクションのメカニズムにもなる.

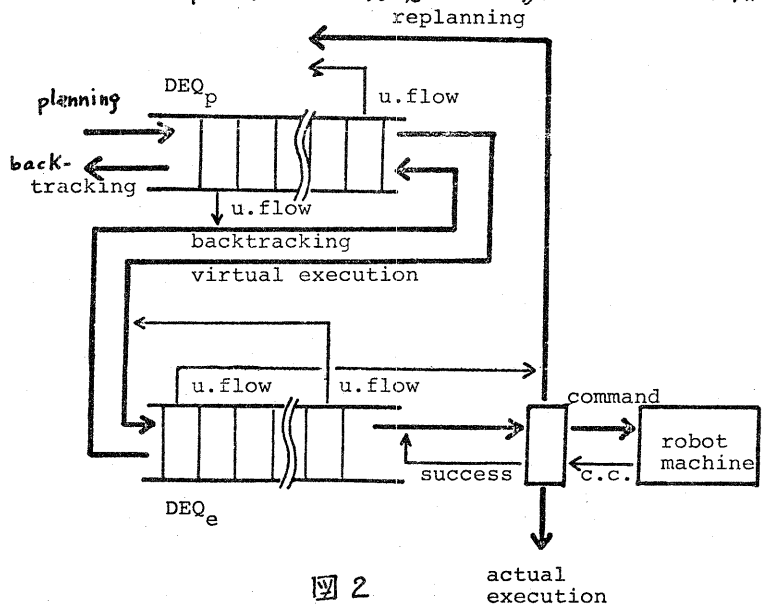


図2

actual execution

また, プランニング部とロボットの間のインタラクションのメカニズムにもなる. ルールや評価基準の変更追加, 異常事態に対する指示などは DEQ_p に入れ, プランニング部に割り込ませかけると, 適当な時点でプランニング部に取り込まれる.

なお, 本章で述べたプランニングシステムは, MELCOM 7500 (37X4E) 60KW, 32bit/w の Foreground LISP⁽⁸⁾ 上にインプリメントされており, 実行システムは

, PDP12 (17メモリ 20kw, 12bit/w) が分担している。

3. インタラクティブなロボットプランニングの応用例

3.1 局所的な認識技能をもった移動形ロボットのルート探索問題

(環境に関する知識が不完全な場合のプランニング)

格子状のルートで構成された図3に示す環境におかれた移動形ロボットを任意の目的地へガイドする問題を考えてみよう。ただし、ルート上には障害物があったり、ルートが途中で消滅していたりして、通行不能になる事態が発生するかもしれないとする。ロボットは、環境の全体を知ることができないが、搭載したTVカメラによって局所的な視覚情報もえ、ビジュアルフィードバックによってルートの上を進むことは可能である。なお、このロボットは、制御用計算機(PDP12)で制御され、データチャネルを介してプランニング用計算機(MELCOM7500)に接続されている。

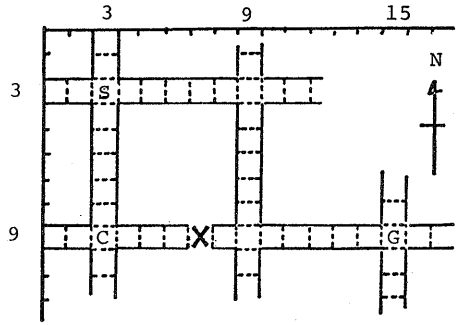


図3

システムの初期設定として、環境の地図、ロボットの動作(前進、回転)に対応する仮想ロボットの行動ルール(ロボットの行動可能条件と、行動したときのモデルへの効果)および、目標の位置(x_g, y_g)を与える。なお、ロボットの状態は $S(x, y, direct)$ で与えられる。ここで、 x, y はロボットの現在位置を示し、 $direct$ はロボットの向いている方向で、N, E, W, S のいずれかをとる。いま、ロボットの初期状態と目標状態をもとに $S_0(3, 3, S), S_g(15, 9, E)$ とする。先づ、最初の可能な行動は、 $(2, 3), (3, 4), (4, 3)$ のいずれかに進むことである。これらの可能性をゴールまでの距離の大きさによって評価し、並びかえると $(3, 4), (4, 3), (2, 3)$ となる。バックトラックに備え、2番目以降の可能性リストを DEQ_E にセーブしたのち、最も確からしい候補 $(3, 4)$ へ進むこととシミュレートに渡し、次の決断点 $(3, 9)$ までシミュレートする。このようにして、 $S_0(3, 3, S), S_1(3, 9, S), S_2(3, 9, E), S_3(9, 9, E), S_4(15, 9, E)$ が次々に DEQ_E に入れられる。制御用計算機からの要求があれば、これらの結果は、*go-ahead, turn-left, go-ahead, go-ahead, go-ahead* と交換され、 DEQ_E に入れられて実行に移される。ルートの一部、たとえば、 $(7, 9)$ が通行不能だとすると、3番目の命令である *go-ahead* の実行中に失敗し、再プランニングが要求される。システムは、失敗した時点のロボットの状態 $S(?, 9, E)$ を推定し、ヒストリマップにその地帯を記入し、 $S(?, 9, E)$ を初期状態にして再プランニングを開始する。

3.2 プランニングにおける手続(ルール)学習問題

(ロボットに関する知識が不完全な場合のプランニング)

ロボットがとり得る行為には、① 適用可能条件、② 適用可能条件が満たされない時の処置 ③ 行為適用後のモデルに与える影響、が前もって付与されており、ある目標が与えられた場合には、①~③を考慮しながら、取るべき行為の連鎖を定めるのが普通のプランニングのやり方である。

いま、もし、ある行為に関し、これらのうち①が与えられなかったとし、試

行錯誤によって命令も実行しながら、①を自ら求めるプロセスで簡単な例について考えてみよう。図4に示される環境で、ロボットが①を知らずに $puton(B, D)$ ^{*1} を実行しようとする場合を想定する。〔注：組み立て、積み上げ作業などで、問題を単純化し（行為の適用可能条件なども無視して）、プランニングの次の段階を進めるときに遭遇する状況である〕

$puton(B, D)$ は、 $liftup(B)$ ^{*2} \wedge $releason(D)$ ^{*3} のように一変数の動作に分解できる。ロボットは、先づ、実際にBを持ち上げよう ($liftup(B)$) として失敗する。そこで、「失敗の原因は、現在の環境の状態 *i.e.* (ATB1) (ONAB) (ATD2) (ONCD) が不都合だからに違いない。不都合の原因を探し出して取り除けばうまくいくかもしれない。」と考えるのは極めて自然であろう。そこで、悪い原因として、Bのセに関連した状態を抜き出し、 $liftup(B)$ の適用可能条件として次の三つの仮説を立て、検証してみる。

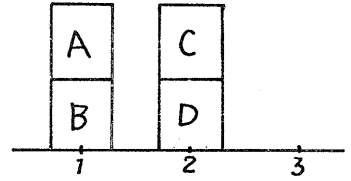


図4

- (i) $\sim(ATB1)$ (ii) $\sim(ONAB)$ (iii) $\sim(ATB1) \wedge \sim(ONAB)$

いま、状態に因り、表1のような操作テーブルが与えられていたとする。

状態	肯定の行為	否定の行為
(ATX P)	$liftup(X) \wedge releason(p)$	$liftup(X) \wedge releason(\sim p)$ ^{*4}
(ONX Y)	$liftup(X) \wedge releason(Y)$	$liftup(X) \wedge releason(\sim Y)$
⋮	⋮	⋮

表1

仮説の検証

(i) 表1より、 $liftup(B) \wedge releason(\sim 1)$ を行なわなければならないが、いま、 $liftup(B)$ が実行不可能なのだから、矛盾。

(ii) $liftup(A) \wedge releason(\sim B)$ は実行可能で、状態は図5か図6のいずれかに移る。次に、 $liftup(B)$ を試みて成功する。仮説 $\sim(ONAB)$ は正しいのである。

(iii) 仮説(i)の場合と同様に矛盾。

次に、状態が図5のようであったとし、 $releason(D)$ を試みるが失敗する。再び、D=関し

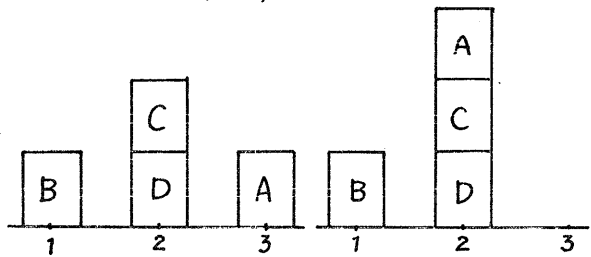


図5

図6

- (i) $\sim(ATD2)$ (ii) $\sim(ONCD)$

- (iii) $\sim(ATD2) \wedge \sim(ONCD)$

なる仮説を立て、前回と同様な手続きで検証すると、正しい仮説の候補として、(i)、(ii) が得られる。(iii) は正しいとして検証の必要はない。

そこで、“検証の手続きが最も簡単な仮説を優先する”という評価基準を導入すると、次の候補の仮説として、 $\sim(ONCD)$ が得られ、この状態で $releason$

*1 BをDの上におく、*2 Bを持ち上げる、*3 持っているものをDの上におく、

*4 持っているものをp以外の上におく。

(D)の実行に成功する。したがって, $puton(BD)$ の適用可能条件として

$$\sim(ONAB) \wedge \sim(ONCD)$$

が求められる。A, C を別の対象物で置き換えた同種の実験を重ねることで, = の条件を

$$\forall X[\sim(ONXB)] \wedge \forall Y[\sim(ONYD)]$$

のように拡張する。さらに一般化を進めて, = のブロックワールドでの $puton(MN)$ の適用可能条件として

$$\forall X[\sim(ONXM)] \wedge \forall Y[\sim(ONYN)]$$

が導かれる。

4. むすび

ロボットプランニングにおける問題として, 一般化と効率化(最適化)を両立させる推論アルゴリズムの研究や不確実な知識の下でのプランニング・実行システムの開発が最近クローズアップされつつある。本論文は, 後者の立場でのインタラクティブプランニングシステムの試みについて述べたものである。ここで取り上げたインタラクティブプランニングの考えも押し進めると, ロボットシステムも, プランニング, 実行, 監視の各サブシステムの有機的な総合体としてとらえることができ, 特に, 複数台ロボットのプランニングに有効だと思われるが, これについては, いづれ別の機会に報告したい。最後に, 3.1における実験に協力し有益な助言も頂いた当所ロボットグループの築根, 井上の両氏に深甚の謝意を表す。

文献

- (1) R. E. Fikes and N. J. Nilsson, Artificial Intelligence, Vol. 2, 189-208, (1971).
- (2) E. D. Sacerdoti, 3rd. Intl. Joint Conf. Arti. Intel., 412-422, (1973).
- (3) C. Hewitt, 1st. Intl. Joint. Conf. Arti. Intel., 295-301, (1969).
- (4) G. J. Sussman and D. V. Mcdermott, Proc. AFIPS, FJCC, 1171-1179, (1972).
- (5) T. Winograd, AI-TR-17, MIT, (1971).
- (6) S. E. Fahlman, Artificial Intelligence, Vol. 5, 1-14, (1974).
- (7) G. J. Sussman, AI-TR-297, MIT, (1973).
- (8) 塚本, 情報処理 15, 327 (1974).
- (9) 井上, 築根, バイオメカニクス 2, 19 (1973).