

# LLP スーパーバイザとステートネットに基づくロボットの行動実現 第2報 時間付きペトリネットを用いた行動計画

○ 仲谷 篤人\* 高橋 秀行\* 小林 啓吾\* 潮 俊光\*

\* 大阪大学 大学院基礎工学研究科

## Motion Realization of Humanoid Robots Based on LLP Supervisory Control and State Nets (2) – Motion Planning Using Timed Petri Nets –

○ Atsuhito Nakatani\* Hideyuki Takahashi\* Keigo Kobayashi\* Toshimitsu Ushio\*

\* Graduate School of Engineering Science, Osaka University

**Abstract:** In this report, we propose a supervisory control system for motion planning of humanoid robots. The proposed system is hierarchically structured into two levels. The upper system takes charge of the motion planning, and the lower system controls and monitors the motion. In this second report, we explain the upper system. The upper level generates an optimal sequence of actions using timed Petri nets.

### 1 はじめに

ロボットの腕、脚などといった各要素ごとの動作をセンサ空間内の状態遷移グラフで表現するモジュラステートネットという手法を第1報で提案した<sup>[1]</sup>。しかし全身の動作がそれらの組み合わせにより表現されるため、組み合わせによっては動作の干渉が起こるという問題がある。本報告ではモジュール化されたステートネットを時間付きペトリネットを利用して管理することにより、干渉が起こらない行動を計画するシステムの構築について述べる。

### 2 システム全体構成

本研究で提案する、モジュラステートネットを管理し干渉の生じる動作の組み合わせを避けて行動をとらせるシステムを Fig. 1 に示す。このシステムは2層構造を持っている。上位層では全身の最適な行動の計画を作成している。下位層はモジュラステートネットにより構成されており、各モジュラステートネットはヒューマノイドロボットの各部の動作の監視と制御を行っている。上位レベルの各部分の働きについて、次に説明する。**path planning** では各部分のモジュラステートネットをペトリネットで離散モデル化しており、これを用いて最適な行動計画を作成している。ペトリネットによるモデルについては第3節で、行動計画については第4節で説明する。**feasibility check** は作成された行動計画が、既に実現可能、もしくは実現不可能と記録されているかどうかを判断し、まだ実現可能かどうか不明の場合は、作成された経路が実現可能かどうかを**collision check** でモジュラステートネットからセンサ情報を得てそれらを統合することにより検証を行っている。作成された

行動計画に干渉が発生した場合、ペトリネットに禁止アークを追加して経路の再検索を行う。干渉が無ければ行動計画列は **path planning** に送られ、**command server** からそれぞれのモジュラステートネットに1命令ずつ送られる。各モジュラステートネットは動作開始命令を受け取るとその命令に対応した行動を行い、行動が終了するまで次の命令を受け取ることはない。

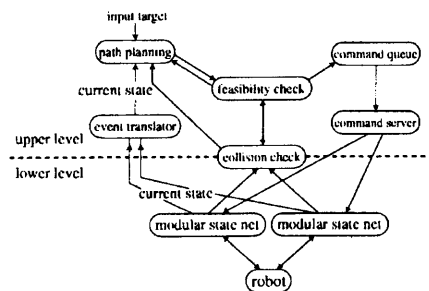


Fig. 1: A proposed system.

### 3 時間付きペトリネット

**path planning** では、時間付きペトリネット<sup>[2]</sup>を操作し各モジュールの動作軌道の組み合わせのうち干渉しないもので、実行時間が最小のものをを選択する。モジュラステートネットをモデル化するために、次のような時間付きペトリネットを導入する。

トレースに関しては、動作の開始を表す開始トランジションと動作の終了を表す終了トランジションを定義する。ブレースに関しては、静的ブレースと動的ブレースを定義する。静的ブレースはモジュラステートネットのノードに対応し、静止状態を表す。動的ブレースはモジュラステートネットのアークに対応し、行動実行状態を表す。モジュラ

ステートネットにおける現在の状態を、ベトリネット内のトークンの位置で表現する。動的ブレースには決められた停留時間が設定されており、その時間が経過すると終了トランジションを発火させる。こうしてモジュラステートネットにおける状態を離散化する。なお動的、静的ブレースはそれぞれ“□”と“○”で表すことにする。

動作の干渉が検出された組み合わせを禁止アークによって回避する。干渉回避の方法を Fig. 2,3 の例で示す。Fig. 2 の例では、右腕の動作がスタートするまで左腕の動作を開始することが禁止されている。Fig. 3 の例では、右腕の動作が終了するまで左腕の動作を開始することが禁止されている。

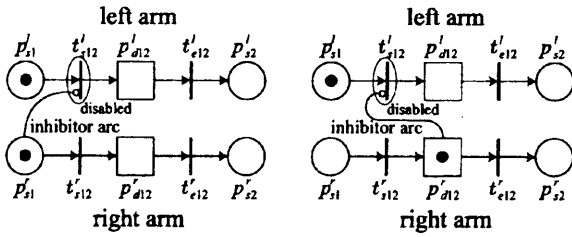


Fig. 2: An inhibitor arc from a static place. Fig. 3: An inhibitor arc from a dynamic place.

#### 4 最適経路プランニング

現在の状態から任意に与えられた状態への最適経路を見つけるために、一種の limited lookahead policy スーパーバイザ制御器 [3] を用いる。予測木の長さを一段一段伸ばしていき、全ての可能な長さの有限の経路からなる木を計算し、現在の状態から最短時間で目標の状態へ至る最適経路を見つける。簡単な具体例を挙げて説明する。

Fig. 4 のような時間付きベトリネットが与えられているとする。このときのマーキングを  $(p_{s1}^l, p_{s1}^r)$  と表すことにする。動作中を表す動的ブレース  $p_{d12}^l$  及び  $p_{d12}^r$  の停留時間をそれぞれ 5, 10 とする。ゴールマーキングを  $(p_{s2}^l, p_{s2}^r)$  とすると、Fig. 5 に表される予測木が作られる。なお図中の  $\tau$  はノードに到達する最小時間である。一番上の枝は右腕の動作終了後に左腕の動作を行うことを意味している。なぜなら、各モジュラステートネットは動作開始命令を受けた後、行動が終了するまで次の命令を受けないためである。この枝の場合、動作終了時刻は 15 である。同様に、二番目の枝は左腕の動作終了後に右腕の動作を行うことを意味しており、動作終了時刻は 15 である。三番目の枝は左腕の動作を開始すると同時に右腕の動作を開始することを意味しており、動作終了時刻は 10 である。また一番下の枝では右腕の動作が先に終了しているが、この場合、左腕動作終了後のマーキングに到達する最小時間が求まらないため、この枝は刈り取られている。これらより最短時間で動作を終了

できる三番目の枝が最適経路として得られ、発火トランジション列  $t_{s12}^l, t_{s12}^r, t_{e12}^l, t_{e12}^r$  が命令列として出力される。

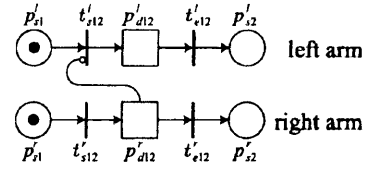


Fig. 4: A timed Petri net.

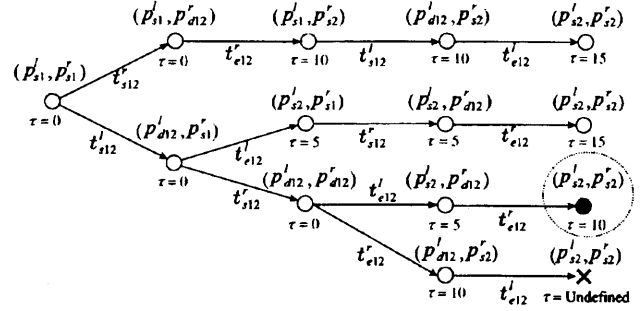


Fig. 5: A tree.

#### 5 おわりに

本報告ではヒューマノイドロボットの制御において、干渉の問題を避けつつ多彩な動きを実現するという手法の基礎として、モジュール化されたステートネットに対しその上位レベルで時間付きベトリネットを用いた行動計画を行い、各モジュラステートネットに適切な指示を与える方法を提案した。動力学的な拘束についての対処や、予期せぬ障害が実際に生じてしまった場合、どのようにベトリネットに表現し下位レベルに動作命令を送るかなどの点が今後の課題として挙げられる。

#### 謝辞

本研究は科学事業振興事業団戦略的基礎研究推進事業 (CREST) 領域「脳を創る」の補助を受けた。また多くの有益な助言をいただいた中村仁彦先生 (東京大学) に謝意を表す。

#### 参考文献

- [1] 高橋秀行, 仲谷篤人, 小林啓吾, 潮俊光: LLP スーパーバイザとステートネットに基づくロボットの行動実現 第1報 モジュラステートネットを用いた行動制御; 第3回システムインテグレーション部門講演会 2002 (2002)
- [2] J. Wang: *Timed Petri Nets: Theory and Application*, Kluwer Academic Publishers (1998)
- [3] S.-L. Chung, S. Lafortune, and F. Lin: Limited Lookahead Policies in Supervisory Control of Discrete Event Systems; *IEEE Trans. Automat. Contr.*, Vol. 37, No. 12, pp. 1921-1935 (1992)