

データベースを用いたコミュニケーションロボットシステムの構築

佐竹 聡[†] 川島 英之[†] 今井 倫太^{†,††}

[†] 慶應義塾大学理工学部

^{††} 科学技術新興事業団さがけ研究 21

E-mail: [†]{satake,kawasima,michita}@ayu.ics.keio.ac.jp

あらまし 本稿はコミュニケーションロボットに適した DBMS の作成を目的とする。しかし DBMS の従来手法である空間インデックス法をロボットに適用した場合は、コミュニケーション履歴を利用した行動選択および文脈を反映した反射行動の生成を実現できない。この問題を解決するために本研究は CDBMS を提案する。CDBMS ではこの 2 つの問題を解決するために、コミュニケーションの関係を有向グラフで表現したインデックス (C-Index) とデッドラインの近いクエリを優先実行する EDF スケジューラとを組み合わせる。また評価実験により、CDBMS は反射行動に文脈を反映することが可能であることを示した。

キーワード Robovie, C-Index, EDF スケジューラ, CDBMS

The construction of communication robot system with database

Satoru SATAKE[†], Hideyuki KAWASHIMA[†], and Michita IMAI^{†,††}

[†] Faculty of Science and Technonlogy, Keio University

^{††} Precursory Research for Embryonic Science and Techonlogy, Japan Science and Technology Corporation

E-mail: [†]{satake,kawasima,michita}@ayu.ics.keio.ac.jp

Abstract This paper proposes a new database system named CDBMS for a communication robot. The difficulties with employing one of the ordinal DB index(Spatial Index) for the communication are that a DB system can't use a communication context to select an appropriate behavior and generate a reactive action in response to the context. To tackle the difficulties, CDBMS consists of C-Index and EDF scheduler. C-Index represents the communication relationship with a directed graph. EDF scheduler carries out queries in order of deadline. In particular, this paper has conducted an experiments. The results have indicated the ability of CDBMS to carry out the actions reacting to context.

Key words Robovie, C-Index, EDF Scheduler, CDBMS

1. はじめに

近年、より直感的なヒューマンインターフェイスの実現を目指し、人間と共存することを目的としたロボットの開発が行われている。日常生活において人間と共存するコミュニケーションロボットを実現するためには、状況に適した行動をロボットに実行させる必要がある。つまりロボットは多様な状況に対応するために多くの行動パターンを持つ必要がある。さらに、実世界ではロボットが予期しない出来事(人間からの割り込みや、周囲の環境の変化)が発生するため「反射行動」を実行可能でなければならない。一方、多くの行動パターンを効率的に管理するためには DBMS(Database Management System) の導入が考えられる。しかし、コミュニケーションロボットに適した

DBMS はまだ実現されていない。そこで本研究ではコミュニケーションロボットに適した DBMS を開発する。

コミュニケーションロボットに適した DBMS が満たさなければいけない条件には以下の 3 つがあげられる。1 つ目の条件はロボットがセンサを通して実世界の情報を得て実行すべき動作の決定をおこなうことである。つまり、DBMS はセンサ情報を入力とし、ロボットの動作を出力とする必要がある。2 つ目の条件はコミュニケーションの履歴を利用しデータを選択していく機能である。なぜならば、人間のコミュニケーションは過去おこなった履歴に依存して内容が変化するためである。3 つ目の条件は反射行動の必要性である。つまり、現在行っている行動を中止し実行しなければならない緊急時の回避行動である。効果的に「反射行動」をロボットに組み込んだ Subsumption [6] [7]

が知られている。しかし、Subsumption では現在のロボットの内部状態に適した反射行動を選択することができない。一方、ロボットは現在の内部状態に適した行動を選択し優先的に決定しなければならない。

そこでコミュニケーションに適した DBMS の作成のため本研究では CDBMS(Communication Database Management System) を提案する。CDBMS は C-Index(Contextual Index) と EDF(Earliest Deadline Fast) スケジューラとで構成される。C-Index とはコミュニケーション同士の関係を有向グラフで表現し、グラフをたどることでロボットが次におこなう行動を決定する。C-Index の導入により、「現在実行中の内容の把握」と「コミュニケーションの履歴の利用」が実現される。さらに EDF スケジューラの導入により検索の優先度変化を可能とし、C-Index と組み合わせることで「文脈を反映した反射行動」を実現する。

本稿の構成は以下の通りである。本稿における研究の前提を 2. 節でのべる。そして CDBMS を 3. 節で提案する。CDBMS を評価するために行った評価実験について 4. 節でのべる。そして CDBMS を用いず、空間インデックス法をロボットに適用した場合の議論を 5. 節にのべる。今後の課題を 6. 節で述べ、最後に結論を 7. 節でのべる。

2. CDBMS の前提条件

本節では本稿の前提についてのべる。まず本研究で使したロボットである Robovie についてのべる。そして、論文で使される単語と記号の定義を行う。

本研究では、人間との対話を行うロボットとして Robovie [9] を用いた。ハードの仕様 [10] を表 1 に示す。Robovie とはコミュニケーション機能に重点を置いて設計されたヒューマノイド型ロボットである。そして人間に対して圧迫感を与えないように人間よりも一回り以上小型に設計されている。

ここからは本稿で使用する記号および定義についてのべる。本研究において「コミュニケーション」とは人間から見てロボットと人間の動作が自然に繋がって見える状態のことだと定義する。このとき、ロボットの動作が終了し人間の反応を待つまでを「コミュニケーションの最小単位」もしくはノードと定義する。そして j を任意定数としたとき、ノードを $Node_j$ と表記する。 $Node$ に対応づけられる情報にはロボットのアクチュエータ情報があり、それは $actInfo_i$ で参照される。ただし i は任意の整数である。また、現在実行中のコミュニケーションを $CurNode$ と定義する。

また $Node_j$ のすぐ後に $Node_k$ を実行可能であるとき、 $Node_j$ と $Node_k$ は連結していると定義し、有向辺 v_m を使い次式で表記する。ただし m は任意の正整数である。

$$v_m = Node_j \rightarrow Node_k$$

つぎに v_m の有限集合を任意の正整数 n を使い V_n とする。 V_n のうち、 $Node_j$ を始点とする辺の集合を $V_n^*(Node_j)$ と表記する。

またノードをたどるときの条件として $evalfunc_j$ を定義す

表 1 Robovie の仕様

重量	39kg
移動速度	1.6m/sec.(最大)
腕の運動速度	200 °/sec.(最大)
バッテリー電力	DC12V 21Ah
台車駆動方式	2 輪独立駆動 1 キャスター
最低地上高	3cm
ステレオカメラ	SONY EVI-G20 ×2 個
ステレオマイク	SENNHEISER MKE104 ×1 個
超音波センサ	24 個
全方位センサ	1 個
温度センサ	1 個
赤外線センサ	2 個
パンパーセンサ	スイッチタイプ × 10 個
指先センサ	スイッチタイプ × 2 個
皮膚センサ	感圧導電性ゴムタイプ × 16 個
腕用モータ	ハーモニックドライブ DC モータ × 8 個
首用モータ	ハーモニックドライブ DC モータ × 3 個
画像入力装置	4 画像混合装置 画像処理ボード
通信装置	無線 LAN
OS	Linux 2.2.16-rtl2.2

る。ただし j は任意の正整数である。 $evalfunc_j$ はセンサを入力にとり実数を返す関数であり、ユーザの定義した条件を満たすならば正の実数を、満たさなければ負の実数を返す。そして、ユーザの条件を多く満たしているほど 0 に近い実数を返す。

また反射行動の検索に利用される表が存在し、任意の正整数 j を用いて $reflectTbl_j$ と表現する。そして反射行動の決定には、どの行動をおこなうかを決定するまでの時間制限としてデッドラインが必要である。デッドラインにはハード、ファーム、ソフトの 3 種類 [8] のリアルタイム性が知られているが、本稿ではファームデッドラインを想定する。

3. CDBMS

本節ではコミュニケーションロボットに適した DBMS の 3 条件を満たすものとして CDBMS を提案する。CDBMS は C-Index と EDF スケジューラで構成され、現在実行中の内容を把握する機能とコミュニケーション履歴を利用する機能を C-Index が提供し、検索の優先度を変化させる機能を EDF スケジューラが提供する。そして本稿において全インデックス構造はメモリ上置かれる。この節ではまず CDBMS 全体について述べる。つぎに C-Index について述べ、最後に EDF について述べる。

3.1 CDBMS

本小節では CDBMS 全体と CDBMS にアクセスするプロセスおよびクエリについて述べる。CDBMS の全体図を図 1 に示す。まず CDBMS にアクセスするプロセスについて述べる。CDBMS はクライアントプロセスとして「通常プロセス」と「反射プロセス」の 2 種類を想定している。通常プロセスとはロボットに唯一存在しコミュニケーションの進行を行うプロセスであり、一方、反射プロセスとはロボットの状態を監視し必要ならば危険回避行動を行うプロセスである。

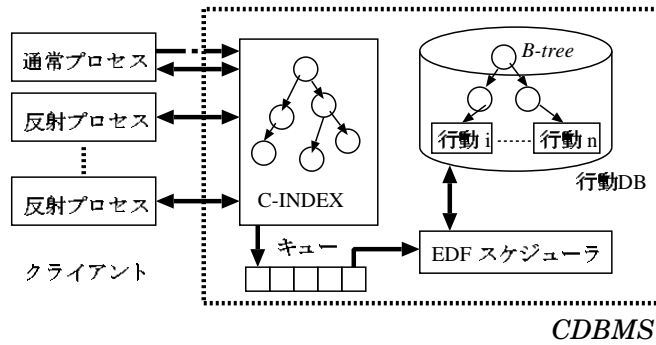


図1 システム構成

つぎに CDBMS に対して発行されるクエリについて述べる。クエリは 3 種類が存在する。1 つ目が通常プロセスが次に実行するコミュニケーションを決定するための「次行動決定クエリ」であり、次行動決定クエリのみが $CurNode$ を変更することができる。2 つ目が通常プロセスがコミュニケーション履歴の反映を行うための「グラフ変更クエリ」である。3 つ目が反射プロセスが緊急時の行動決定を行うための「反射クエリ」である。次行動決定クエリと反射クエリは結果としてロボットのアクチュエータ情報を取得し、グラフ変更クエリは結果を取得しない。

全てのクエリはまず C-Index により処理される。そして、次行動決定クエリと反射クエリは C-Index で処理された結果、ロボットのアクチュエータ行動のキーである $actInfo$ を取得する。この 2 種類のクエリは自分自身のデッドラインを入力とすることで、EDF スケジューラにより決定される優先順位を取得し、最後に最高優先度を持つクエリのみが $actInfo$ に対応するロボットのアクチュエータ情報を取得する。

3.1.1 通常プロセス

通常プロセスは、コミュニケーション実行中においてただ 1 つ実行されているプロセスである。そして通常プロセスは次行動決定クエリとグラフ変更クエリとを発行し、「コミュニケーションの進行」及び「コミュニケーション履歴の反映」を行う。

次行動決定クエリが発行されるのは、Robovie の行動終了時であり、グラフ変更クエリが発行されるのは、特定の $Node_a$ が $CurNode$ でなくなったときである。

3.1.2 反射プロセス

反射プロセスはコミュニケーション実行中に複数存在し、それぞれ別のポリシにしたがって、センサデータを読みながら危険状態がないかを監視する。そして反射プロセスが危険状態を認識した場合、ユーザ定義のアルゴリズムにより危険度からデッドラインを生成し、そのデッドラインと自分に登録されている反射行動の種類とを入力とする「反射行動決定クエリ」を発行する。このとき危険度が高い程生成されるデッドラインは短いものとなり、より高い優先度においてデータベースの検索が行われる。

3.1.3 アクチュエータ情報

CDBMS におけるアクチュエータ情報としてロボットのアクチュエータ情報を相対的な値である $ModifyAction$ と、絶対

- (1) $V_k^*(CurNode)$ の全要素の $evalfunc$ で $Sensor$ を評価
- (2) その中で 0 以上の最小値をもつ $Node_i$ 次の $CurNode$ に設定、ただし存在しなければ $NoAction$ を返す
- (3) $CurNode$ に対応づけられている $actInfo_i$ を EDF スケジューラで処理したものを返す

図2 次行動決定アルゴリズム

的な値である $BaseAction$ を定義する。クエリの結果得られる情報は $ModifyAction$ と $BaseAction$ とに分類され、1 つの $BaseAction$ に正整数個の $ModifyAction$ を足し合わせた結果によりロボットの行動は決定される。

複数の形式の定義は一般的な反射行動を実現のために必要であり、Subsumption [6] [7] では実現不能であった「文脈に依存した反射行動」を実現する。

3.2 C-Index

C-Index ではコミュニケーション履歴利用の低コスト実現と、 $CurNode$ 実行中の内容を認識する機能とを提供する。まず C-Index とはコミュニケーション同士の関係を有向グラフで表現したものであり、グラフをたどることでコミュニケーションの進行を実現し、そしてグラフ変更によりコミュニケーション履歴利用を実現する。

3.2.1 ノード

ノードはグラフにおける点であり次の要素をもつ。ただし、 id とはノードの識別子であり、 $actInfo_i$ とはロボットのアクチュエータ情報であり、 $reflectTbl_l$ とは反射行動のノード毎の集合である。また、 id 、 i 、 k 、 l は任意の正整数である。

$$Node_{id} = \{id, actInfo_i, V_k^*(Node_{id}), reflectTbl_l\}$$

3.2.2 辺

ノード間を有向辺で結んだものを辺と定義する。また $Node_j$ を始点とする辺はすべて、 $Node_{id}$ の $V_k^*(Node_{id})$ の要素である。そして各辺は終点に進むための条件として、 k を任意正整数とする $evalfunc_k$ を持つ。本研究において、辺は始点ノードからのポイントで表現されており、その情報は始点ノードである $Node_{id}$ の $V_k^*(Node_{id})$ に格納される。

3.2.3 クエリ

ここでは 3 種類のクエリそれぞれにおいて C-Index がどのような処理を行っているかについて述べる。

a) 次行動決定クエリ

C-Index が次行動決定クエリを受け取り $CurNode$ の行動が終了していればクエリが処理される。もし $CurNode$ の行動が終了していなければ、行動が終了していないことを示す $NoAction$ が返される。

そして行動決定クエリは現在のセンサ情報である $Sensor$ で次行動を決定する。このアルゴリズムを図 2 に示す。

b) グラフ変更クエリ

グラフ変更クエリはコミュニケーション履歴を利用した結果をコミュニケーション自体に反映するためのクエリである。本稿では、 $CurNode$ から接続されるノードの一群は今後おこな

いうコミュニケーションの集合を形成している。このため「コミュニケーション履歴」の反映は、「辺の付け替え」または「evalfuncの付け替え」のいずれかを実行し、未来の行動プランの変更を行う。

「辺の付け替え」とは、今後おこないうるコミュニケーション群の追加および削除をおこなうグラフ変更クエリである。これはコミュニケーション A をおこなったのちコミュニケーション B が実行可能になるとき、コミュニケーション B に対応する Node を目標とする辺に付加することで実現される。本研究での「辺」は始点からのポイントで表現されているため、「辺の付け替え」はポイント付け替えという低コストで実現できる。

「evalfuncの付け替え」とは、辺に対応している evalfunc の変更を行うクエリである。これはコミュニケーション A の行われる回数が少ない時に、A を行われやすくするために実行される。

c) 反射クエリ

反射クエリは反射行動を問い合わせるクエリである。反射クエリは、反射行動の種類とデッドラインを入力として持ち、Robovie の行うべき行動を出力とする。そのとき反射行動は、CurNode とそれに対応づけられている reflectTbl_i から決定される。

reflectTbl_i は Node_i ごとに存在し、その実体は中身に actInfo をもつ表である reflectLine_m を反射行動の種類だけ集めたものである。reflectLine_m の一例を図 3 に示す。1 行目は与えられた「デッドラインの範囲」を示しており、左側が最小値 (以上)、右側が最大値 (未満) を意味する。そして右側に数字が与えられないときは無限大を意味する。図 3 の reflectLine_m は、デッドラインが 0 以上 dl_1 未満の場合は反射行動として actInfo_i を取得し、デッドラインが dl_{k-1} 以上 dl_k 未満の場合は反射行動として actInfo_k を取得し、デッドラインが dl_k 以上の場合は反射行動として actInfo_l を取得することを意味する。

反射クエリ				
デッドラインの範囲	0- dl_1	...	$dl_{k-1} - dl_k$	$dl_k -$
アクチュエータ情報	actInfo _i	...	actInfo _k	actInfo _l

図 3 reflectLine_m の構造

つぎに反射行動決定アルゴリズムをのべる。反射クエリは、「反射行動の種類」と「デッドライン」とを入力としてもつ。反射行動決定アルゴリズムを図 4 に示す。反射クエリは複数のプロセスから同時に発行されることを想定しているため、反射クエリは矛盾するアクチュエータ情報を返す可能性がある。この状態を回避するため反射クエリを実行したあと自分自身より優先度の低いクエリをキャンセルする「クエリキャンセル機能」を CDBMS は持つ。

3.3 EDF スケジューラ

EDF スケジューリングはデッドラインの早い順に優先度を付ける方法であり、一般的にプリエンブション可能な条件で使用される。この手法は、タスク間が独立であるとき最大遅れ時間を最小化することができる最適化スケジューリング法である。

- (1) CurNode の reflectTbl を検索対象とする
- (2) reflectTbl のなかで、「反射行動の種類」と同じ種類の reflectLine_m を検索対象とする
- (3) reflectLine_m の中で「デッドライン」が「デッドラインの範囲」内にあるものを探し、「デッドラインの範囲」に対応付けられている行動 actInfo_k を検索する
- (4) 検索結果として得られた actInfo_k と「デッドライン」とを EDF スケジューラが処理しアクチュエータ情報を得る
- (5) 得られたアクチュエータ情報を検索結果として返す

図 4 反射行動決定アルゴリズム

本研究では、EDF スケジューラの処理は C-Index の後に行われる。EDF スケジューラに送られるクエリは、入力にデッドラインと actInfo_k をもち出力にロボットのアクチュエータ情報を返す。各々のクエリは同時に実行され、デッドラインを用いた EDF スケジューラで優先度を変更しながら行動 DB の検索が行われる。EDF で決定された唯一つのクエリのみが行動 DB にアクセスでき、かつ検索の途中においてもプリエンブション可能な機能を本研究の EDF スケジューラは提供している。

4. 評価実験

本節では CDBMS を Robovie に搭載し、Robovie が状況に応じた反射行動を決定可能かを評価するために行った 4 つの実験について述べる。実験 1 では異なる数のデータを CDBMS に登録したときの、データ数と検索時間を計測することで CDBMS の探索速度の評価を行う。実験 2 では異なるデッドラインの反射クエリを発行し、より緊急度の高い行動の優先検索が実施可能であるかを評価する。実験 3 では複数の ModifyAction を返すクエリを発行を行い、反射行動の自由度を上げられているかを評価する。実験 4 では、コミュニケーションモデルを Robovie に搭載し Node_i 毎に異なる反射行動が実施されているかを評価する。

また各々のモデルにおいて競合状態を発生させるため、10000 回のクエリを繰り返し発行させた。これは測定時間をより高い精度で求めるためである。そして実験 1~3 において、Robovie の首の Pitch 方向 (垂直方向を回転軸とした頭部の移動方向) の電圧を減少および増加させるアクチュエータ情報を返すようにクエリを設定した。

実験 1 は登録データ数と検索時間との関係を評価するために行った。この実験は、Robovie の首に Pitch 方向の電圧 0.5V 下降させるアクチュエータ情報を返すクエリを 10msec 毎に発行することで検索時間の評価を行う。このときの結果を図 5 に示す。このグラフの横軸は 100msec 単位で計測された実行時間であり、縦軸は現在の電圧である。そして、同じ電圧に到達するまでの時間により検索速度が計測される。

グラフより、最大検索時間と最小検索時間の差がほぼ 50msec であることがわかる。クエリを 10000 回重ねて発行していることを考慮すると、1 クエリごとの検索速度差は $5 \mu \text{ sec}$ 程で

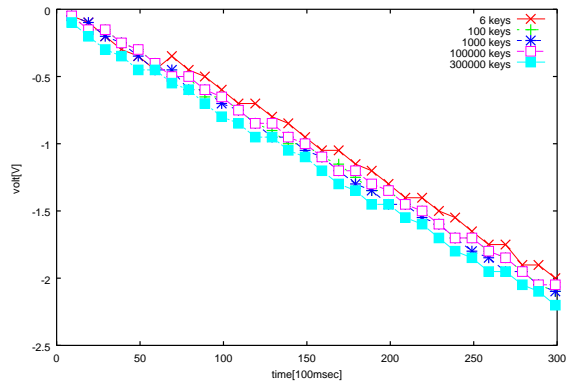


図 5 データ数と検索速度の関係 (実験 1)

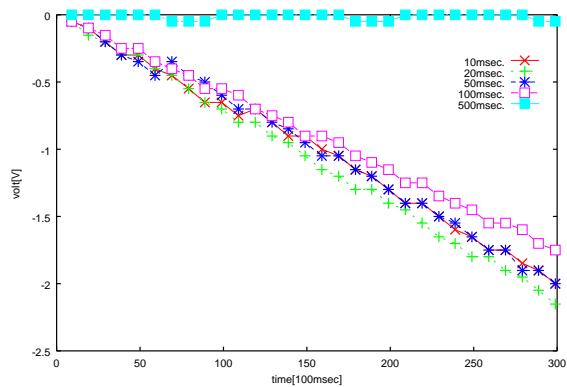


図 6 クエリ発行周期と優先実行の関係 (実験 2)

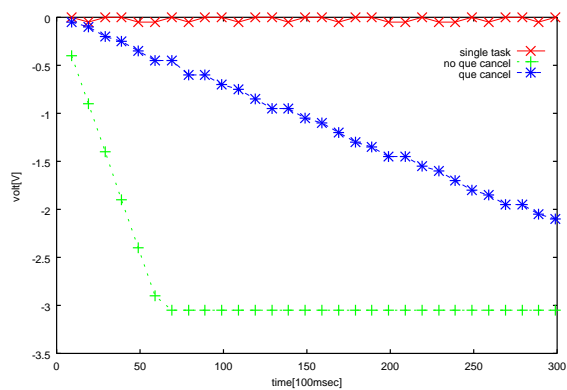


図 7 *ModifyAction* と *BaseAction* の関係 (実験 3)

あり、ほとんど差が出ないことがわかる。これにより CDBMS は、データ数増加による応答時間の劣化がほとんどないことがわかった。これは *B-Tree* インデックス構造をメモリに展開した為だろう。

実験 2 では異なるデッドラインのクエリを発行し緊急度の高いクエリを優先的に検索しているかの評価を行った。この実験は首の Pitch 方向の電圧を 0.5V 下降させる結果を返すクエリ A と 0.5V 上昇させる結果を返すクエリ B を同時に発行させる。この実験においてクエリ A は 100msec のデッドラインを設定し、クエリ B は 150msec のデッドラインを設定した。そして発行の周期を 10msec、20msec、50msec、100msec、そして 500msec の 5 種類の場合において測定を行った。さらに優先実行の確認のため、先に終了したクエリが残りのクエリをキャ

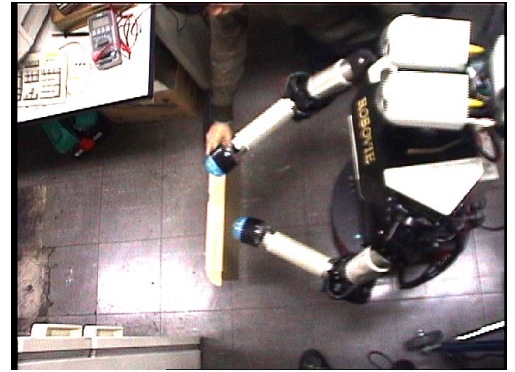


図 8 コミュニケーションモデルの適応 (実験 4)

ンセルする条件を導入した。このときの結果を図 6 に示す。この結果も実験 1 と同じく横軸が 100msec 毎の時間であり、縦軸が首にかかっている電圧である。

グラフより、周期が 100msec 以下ではクエリ A が優先的に実行されていることが得られる。一方、周期が 500msec のときは両クエリとも正しい検索結果を得ている。この理由は、500msec とあまりに広い周期幅を取ってしまったためクエリキャンセルが上手く働かなかったためである。また、10msec 周期の検索速度が 50msec 周期の速度を下まわる結果も得られた。これは 10msec 周期の場合、クエリキャンセル機構が頻繁に呼び出されオーバーロード現象をひきおこした為と推測される。以上の結果より、CDBMS は短いデッドラインのタスクを優先的に処理できることがわかった。

実験 3 では複数の反射行動を組み合わせる *ModifyAction* の有用性の評価を行った。この実験は首の Pitch 方向の電圧を 0.5V 下降させる *BaseAction* を返すクエリ A と、0.5V 上昇させる *BaseAction* を返すクエリ B、0.5V 上昇させる *ModifyAction* を返すクエリ C の組み合わせによる違いを測定することで評価を行う。このとき評価は、クエリ A のみ実行する single task と、クエリ A と B とを共に実行する no que cancel と、クエリ A と C を共に実行させる que cancel の結果を比べることでおこなう。クエリ A・B・C は同じデッドライン 100msec をもち、50msec 周期でクエリを発行を行う。この結果を図 7 に示す。図 7 も横軸が 100msec 毎の時間であり、縦軸が電圧である。

single task と no que cancel との間に que cancel が存在する結果が得られた。これは *ModifyAction* が相対的にクエリ A の結果を変更していることを示す。つまり、*ModifyAction* によって全く別の行動を生成可能であることを示し *ModifyAction* の有用性が得られた。

実験 4 は異なるノード実行中に異なる反射行動を返せるかを確認した。実験の様子を図 8 に示す。また実験の結果として、状況に適した反射行動を実行可能である結果を得た。

5. 議 論

1. 節で述べた条件を CDBMS を使わず、センサデータをキーとした空間インデックスを用いて実現する方法を議論する。空間インデックス法とは多次元データのキー検索を高速に行うためのデータ構造である。その中には検索結果に点データを扱

う *K-D-B-Tree* [3] と、範囲を扱う *R-Tree* [1]・*R*-Tree* [5]・*R+-Tree* [4]・*SR-Tree* [2] 等が存在する。このような空間インデックスは、特定のセンサ入力に対するロボットの行動の高速検索を提供するが空間インデックスの作成に大きなコストがかかるという特徴をもつ。この時、空間インデックスをただ一つ作成し、センサデータをもとにロボットの次行動を高速に検索可能とする DBMS を搭載することを「最も単純なモデル」とする。

しかし、空間インデックス法を単純に用いただけでは、条件 2 および 3 を満足することができない。「最も単純なモデル」では「ロボットが次におこなう行動とセンサ入力との関連」を記述したものが空間インデックスとなる。しかしロボットの次行動は人間の反応と過去に行ってきた「コミュニケーション履歴」で決定される。またコミュニケーション毎のロボットの反応は同じセンサ入力であっても異なる。よって「ロボットの次行動とセンサ入力との関連」は、常に同じであるとは言えない。忠実に「最も単純なモデル」でこの関連の変化を実現しようとする、コミュニケーションが切り替わるたびにインデックスの再構成を行わなければならない。インデックスの再構成には、非常に高い計算コストが必要である。従って「最も単純なモデル」で条件 2 を実現することは困難である。次に条件 3 を解決できない理由についてのべる。これには二つの理由が存在する。1 つ目の理由は空間インデックス法で現在ロボットが実行しているコミュニケーション状態を取得できないことである。このため、DBMS は文脈に依存した「反射行動」を決定することができない。2 つ目の理由は DBMS が 1 ユーザアプリケーションとして実装されるためである。ユーザアプリケーションでは、DBMS ではロボットで要求される厳しいリアルタイム性を満足するタスクスケジューリングの実行が困難である。以上の理由により「最も単純なモデル」ではコミュニケーションに適した DBMS を提供することができない。

6. 今後の課題

今後の課題には以下の 5 つが存在する。1 つ目の課題は、グラフ変更クエリを連想可能なコミュニケーションモデルに適応させることである。なぜならば、コミュニケーションは直前に行われた結果から連想を行い、次行動に反映を行うからである。しかし、本稿でのグラフ変更クエリや C-Index は、連想の実現を考慮していない。

2 つ目の課題はデータ数の多いコミュニケーションモデルでの評価である。なぜならば、現時点ではグラフ変更クエリを含んだコミュニケーションモデルでの速度測定を行っていないからである。そのため、より人間同士のコミュニケーションに近い環境で評価を行わなければならない。

3 つ目の課題はデータベースに格納されているデータ形式のモデル化である。なぜならば、コミュニケーション中のロボットの表現には画像および音声情報が必要不可欠であるにも関わらず、画像情報、音声情報を本稿は扱っていない。このため、アクチュエータ情報と画像情報および音声情報で同期モデルを作成し、CDBMS に適応させなければならない。

4 つ目の課題は EDF スケジューラの改良である。現時点での EDF スケジューラはデッドラインミスが発生した場合の処理機構が存在しない。また、デッドラインはロボットの行動が終了するまでの時間のため、データベースに格納されているアクチュエータ情報のフィードバックを行わなければならない。しかし現システムではこの点も考慮されていない。EDF スケジューラを改良しこれら 2 点を解決する必要がある。

5 つ目の課題はデータの格納場所の問題である。本稿でのコミュニケーション中のロボットでは全てのデータがメモリ上に置かれているが、今後データをディスク上に格納できるよう改善していかなければならない。

7. 結 論

本稿は、状況に適したコミュニケーション行動をロボットに提供する CDBMS を提案した。そして、コミュニケーション履歴を利用し現状の把握を行う C-Index を開発し、緊急時の行動の優先検索を提供する EDF スケジューラを導入した。評価実験により、CDBMS が文脈を反映した反射行動を実現できることを示した。

文 献

- [1] Antonin Guttman, "R-Trees, "A Dynamic Index Structure for Spatial Searching," In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.47-57, 1984.
- [2] Norio Katayama et al. "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," In the Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, , Tucson, Arizona, 13-15 May 1997.
- [3] John T. Robinson, "The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes," In the Proceedings of ACM SIGMOD Conference, 1981, Ann Arbor, MI, April 1981, pp. 10-18.
- [4] T. Sellis, N. Roussopoulos, and C. Faloutsos, "The R+-tree: a Dynamic Index for Multi-Dimensional Objects," Proc. of the 13th VLDB Conf., Brighton, England, pp.507-518, Sep. 1987.
- [5] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: an Efficient and Robust Access Method for Points and Rectangles," Proc. ACM SIGMOD, Atlantic City, USA, pp.322-331, May 1990.
- [6] Brooks, R. A., "How to build complete creatures rather than isolated cognitive simulators," in K. VanLehn (ed.), Architectures for Intelligence, pp. 225-239 , Lawrence Erlbaum Associates, Hillsdale, NJ, 1991
- [7] Brooks, R. A. , "Intelligence without representation, " Artificial Intelligence. 47, pp. 139-159, 1987
- [8] Joakim Eriksson, "Real-Time and Active Database: A Survey," In Lecture Notes in Computer Science 1553: Active, Real-Time and Temporal Database System, pp. 1-23, September 1997.
- [9] Takayuki Kanda, Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, and Ryohei Nkatsu, "Development and Evaluation of an Interactive Humanoid Robot "Robovie", " IEEE International Conference On Robotics and Automation(ICRA 2002),pp. 1848-1855,2002
- [10] "Robovie-基本仕様," <http://www.revast.co.jp/Revast%20Homepage-Dw/root/robovie/spec.html>