イベント駆動型制御とその表現
—動的環境下における多自由度ロボットの運動学習を目指して—

山口 明彦* 川嶋 宏彰** 松山 隆司**
*京都大学 工学部，606-8501 京都市左京区吉田本町
**京都大学 情報学研究科，606-8501 京都市左京区吉田本町

砂地やトランポリンといった動的環境下における多自由度ロボットの運動制御は困難な課題である．運動学習によるアプローチを取る場合であっても，ロボットの運動によって環境の状態が変動するため，参照軌道によって運動を記述することは現実的でない．そこで本研究では，環境との接触といった身体の力学的拘束条件の変化と，局所的な制御信号との時間関係に基づいて，定性的な運動の知識を表現し，学習によって定量的な運動制御パラメータを求める枠組を提案する．トランポリン上での跳躍運動をタスクとして，シミュレーションにより手法の有効性を検証した．

Event-driven control and its representation
—towards a motion learning of multi-DOF robot in dynamic environment—

Akihiko YAMAGUCHI* Hiroaki KAWASHIMA** Takashi MATSUYAMA**
*Faculty of Engineering, Kyoto Univ., Yoshida-Honmachi, Sakyo, Kyoto, JAPAN 606-8501
**Grad. Sch. of Informatics, Kyoto Univ., Yoshida-Honmachi, Sakyo, Kyoto, JAPAN 606-8501

Motion acquisition of multi-degree-of-freedom (multi-DOF) robot in dynamic environment, such as sandy soil and trampoline, is very difficult. Even in approaching with motion learning, it is not effective to describe the motion by reference trajectory since the state of the environment is changed by the action of the robot. In this paper, we suggest a new framework of motion learning: representing qualitative knowledge of human with the temporal relation between local control signal and the changing of dynamical constraint of the body, such as landing or taking off, then learning quantitative control parameters. We take simulations of hopping task on a trampoline to verify our methods.

1 序論

1.1 背景と目的

近年ヒューマノイド型をはじめとする多自由度ロボットが開発され，2 足歩行など，かなり高度な運動が実現されている．しかし，それらの運動の多くは静的な環境のもとで行われることが多く，動的な環境，すなわちロボットの行動によって環境の状態（例えば砂地やトランポリンのような環境を想定）やロボットの身体と環境の関わり方（接触状態など）が変わるような環境下での運動には対応できていないのが現状である．また，歩行などに関してはかなり多くの研究がなされ，数多くの手法が提案されているが，そのほとんどが歩行に特化したものであり，汎用的に運動を実現する手法は確立されていない．

本研究の目的は，動的環境下におけるロボットの運動生成を，汎用的に実現する枠組を提案することにある．ここで実現したい「運動」とは，歩行や跳躍などのある程度複雑な運動で，跳び箱運動のような高次の運動も対象とする．このような運動の特徴として，身体の力学的拘束条件が変化する，つまりロボットと環境の接触状態や，ロボットのリンク間での接触状態，あるいは可動範囲が限定されている関節角の変化によって，微分方程式が不連続に変化することがあげられる．

ここでは運動生成を，ロボットに実現させたい運動（軌道）に関する記述が与えられたとき，それを実現するための制御信号を求める問題として考えている．このときの課題には，大きく分けて (a) 適切な運動の記述をどのように行うか，(b) 運動の記述から制御信号をどのように計算するか，(c) 環境の揺らぎ（例えば床反力の摂動）にどのように対処するかの 3 つがある．

(a) については，(a-1) モーションキャプチャなどで人間から得たデータをもとに参照軌道で与える方法，(a-2) 解析的に参照軌道を求める方法（例えば歩行パタンなど），(a-3) 実現軌道に対する評価関数を直接的に与える方法（例えば跳躍なら高さを使って表現）などが考えられている．それぞれの問題点として，(a-1) はキャプチャした軌道で必ずしもタスクが実現できない [1]，(a-2) は汎用性が無く運動によっては困難，(a-3) は軌道を与えるわけではないため (b) の問題を解析的に解くことが困難，と言ったことが考えられる．さらに，参照軌道で運動を記述する (a-1) や (a-2) は，動的な環境では環境の変化も考慮する必要があるという困難さを含んでいる．

一方 (b) の課題には，大きく分けて (b-1) 逆動力学を解析的に解くアプローチと，(b-2) 学習によって求めるアプローチがある．前者は，基本的には運動の記述が軌道で与えられることを前提としていることが多い．いずれにせよ，この問題は次のような困難さを含んでいる．すなわち，(b-a) ロボットの自由度が大きい，(b-b) 身体の力学的拘束条件が変化[1]，(b-c) 劣駆動系 [3][4]，(b-d) 環境のモデル化誤差（床や地面の反作用モデルを正確に作ることは難しい [1]），などである．解析的に解くアプローチでは，ロボットの姿勢に制約を加えて (b-a) を解決することが一般的に行われるが [5]，エネルギロスにつながることもある．解析的に解くアプローチでは，特に動的な環境下では (b-d) に大きく左右されるが，学習によるアプローチならこのような誤差を吸収するというメリットがある．

(c) の課題については，例えば歩行なら ZMP を安定領域に収まるように制御するなどの手法が提案されているものの，汎用的な手法はまだ確立されていない．

動的な環境においては，環境自体も変動するために，参照軌道を与えることは困難である．また，仮に与えられても，環境は直接制御できない．このため，(b-1) の解析的に逆問題を解く手法も同様に困難となり，運動を学習することが有効なアプローチのひとつであると結論される．

1.2 運動学習と従来手法

運動学習には，実現軌道側を学習するか，制御信号側を学習するか，2 通りのアプローチがある．

実現軌道側を学習するとは，(a) 運動の記述としてある評価関数を与えてそれをもとに参照軌道を学習し，(b) 逆問題については解析的に解く，という方法であり，ロボットの運動学習ではこのタイプの手法が多い．この場合でも (b-d) の問題は学習によって吸収される．最適化アルゴリズムには強化学習 [6] や遺伝アルゴリズム [7] などが用いられている．

一方，制御信号側の学習とは，(a) 運動の記述として評価関数を与え，(b) それを最大にするような制御信号を探索する，という方法である．もっとも単純な方法としては，制御信号を時間方向に離散化し，ベクトルとして最適化するなどが考えられるが，離散化を粗くすると運動を実現できる解が存在しなくなる一方で，細かすぎると学習に時間がかかり，あまり適切ではない．

しかし，一般に学習においては探索空間をいかに小さく設計するかが重要な課題となることから考えると，制御信号側の学習の方が好ましい．実現される軌道には身体と環境がつくり出す複雑さが含まれているからである．

もうひとつ観点として，人間が持つ知識をどのように伝えるかがある．人間が持つ運動知識を伝えることによって，学習を早めることを期待する．

1.3 提案手法の概要

1.3.1 制御則とその表現

本研究では，(1) 制御信号側を学習する，(2) 人間が持つ運動知識を伝える，というふたつの要求を満たし，かつ (3)(b) の問題を解決するために，「イベント駆動型制御」と名付ける制御則を提案する．

イベント駆動型制御は，人間の定性的な運動知識（身体の各関節に力を加えて行く手順）を直接的に反映可能な制御則であり，身体のある関節に対する局所的な制御信号—プリミティブ—を出力するタイミングを，ほかのプリミティブや，観測軌道から安定して抽出される点—イベント—との時間関係から決定して出力する．ここでイベントとは，具体的には身体の力学的拘束条件が変化した瞬間などである．よってイベント駆動型制御は逆動力学問題を解かずに直接制御信号を出力する制御則で，(1) の要求を満たす．

イベント駆動型制御で参照するイベントは，着地といった身体の力学的拘束条件の変化を含んでいる．つまりイベントの前後でダイナミクスが変化し（運動方程式が変化），イベントの前と後では制御信号が運動に与える影響が大きく異なる．イベント駆動型制御ではこの事実を利用して，制御信号の複雑さ（具体的にはプリミティブの数や形状）を達成すべき運動の複雑さに合わせて調整することが可能であり，(b-a) の問題や，学習における探索空間の爆発といった問題を克服している．

(b-b) の拘束条件が変化するという問題については，むしろ積極的にその事実を利用している．このことは，Kuniyoshi らの “invariant features” に関する研究 [1] から得られた知見に一致する．この知見とは，

- K1 身体の力学的拘束条件を利用すれば，常に精密な制御をしなくても，ツボ (knacks) を押えた制御で大域的にはタスク（運動）が実現できる
- K2 タスクを成功させるために重要なのは，トータルエネルギや関節角ではなく，局所的なエネルギを出力するタイミングである

というものであり，イベント駆動型制御が有効に機能する根拠のひとつである．

また，(b-c) の問題については，ロボットの身体特性（リンクの構造やアクチュエータの数など）により運動を実現不可能な場合を除き，学習（探索）

[1] ハイブリッドシステムの枠組でこの問題を解決し，フィードバック制御によってロボットに運動を生成させる手法などが提案されている [2] が，フィードバック制御では平衡状態を目標とするため，複雑な運動実現には無理がある．

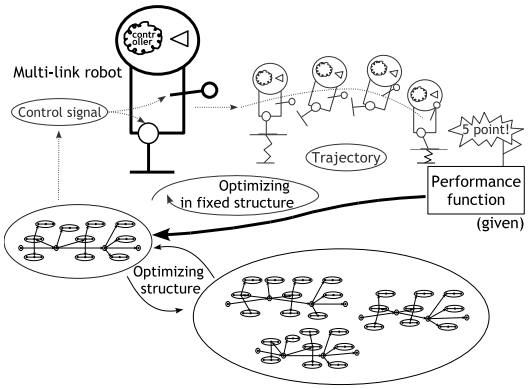


Fig. 1: Motion learning of multi-link robot in a dynamic environment

によって解決できる．(b-d) の誤差についても，学習によって吸収される．

本稿では，イベント駆動型制御の表現として，イベントやプリミティブといった要素間の依存関係及び時間関係を記述したグラフ構造—EventGraph—を導入し，イベント駆動型制御の有効性を検証する．

1.3.2 学習手法

EventGraph の学習は，

1. グラフの構造を固定したパラメタ（プリミティブの形状など）の最適化
2. グラフの構造自体の最適化

の 2 段階に分けられる（Fig.1 参照）．

一度最適化されたグラフの構造は環境や身体の状態の変動に対して大きく変化しないため，異なった環境下でもグラフ構造を固定した最適化のみ行えばよく，効率的であると考えられる．また，身体性が同じロボットはもとより，身体性が異なるロボット（ただしリンクの構造は同じという想定）に対しても，あるロボットが学習した制御表現 (EventGraph) を初期値にして探索することで効率的な学習が行えると考えられる．

本稿では，上記 1 のグラフの構造を固定した学習を中心に議論を深め，2 の，グラフの構造自体の最適化に関しては基礎的な検討を行う．なお，グラフの構造については人間の定性的な運動知識に基づいて与えており，人間と類似した身体性をもつロボットに対しては，ある程度最適であると考えられる．

グラフの構造を含めた最適化を実現することで，未知の（学習していない）動的環境への適応や，人間とは異なる身体性をもつロボットが運動学習をすることにつながるだろうと期待される．また，グラフ構造に対する演算（ノードの追加や削除，グラフ同士の連結など）を制御信号に反映させることが実現できれば，高次の運動（跳び箱運動など）を動的に生成することが可能になると考えられる．本研究の提案手法は，そのための基盤となる技術である．

2 イベント駆動型制御

2.1 イベント駆動型制御の概要

まず,「人間の定性的な運動知識」について,「跳び箱運動（開脚跳び）」の身体制御の手順を例に取って説明する．

> 踏切板まで助走し，踏切板を蹴って跳ぶ．跳び箱で手を着いて，手で体を前方に加速させると同時に足を開く．体が跳び箱の真上に来たら跳び箱を手で押して前方に跳び，空中でバランスを整えつつ着地．

この例からわかることは，制御（体の各部位に力を加えること）の手順は,〔踏切板に到達する〕や〔跳び箱に手が着く〕と言った「イベント」と関係していることである．この関係とは，イベントよりも前に制御が位置したり，イベントと同期，あるいはイベントより後に制御が位置するといったものである．また,「手で体を前方に加速させると同時に足を開く」など，体の各部位に対する局所的な制御の間でも同期や遅延の関係がある．

以下では，局所的な制御信号（プリミティブ）を出力するタイミング，及びその形状について議論する．

2.2 プリミティブの出力タイミングの決定

イベント駆動型制御では，前述した制御の「手順」は変わらないと仮定する．このことは，プリミティブやイベントの間に，何らかの，環境や身体状態の変化に依らない構造が存在することを意味する．プリミティブの出力タイミングの決定は，この構造に基づいて行われる．

前節で述べた跳び箱運動の例からわかるように，あるプリミティブを出力するタイミングに関係しているものには，ほかのプリミティブとイベントがある．ここで「ほかのプリミティブ」には，同じ関節に対するほかのプリミティブだけでなく，異なる関節に対するプリミティブも含まれている．すべてのリンクが連結している以上，関節が異なっていてもそこに何らかの因果関係が存在すると考えられるからである．

イベントとは,〔踏切板に到達する〕や〔跳び箱に手が着く〕と言った，運動の実現軌道上の点（瞬間[2]）である．これらのイベントは，床から踏切板，あるいは空中から跳び箱に身体の一部が移行する瞬間であるから，その物理的意味は「力学的拘束条件の変化」である．

よって，このようなイベントを基準にして制御を

[2]物理的にはおそらく連続的に変化していると考えられるが，この場合，瞬間として捉える（変化が急峻で，その区間幅が十分に短いため）．

するタイミングを決める目的は，身体の力学的拘束を安定に利用するためであると考えられる．この目的を「プリミティブを出力するタイミングの安定化」に拡張し，これに合わせてイベントの定義を「ある運動の実現軌道上から，安定して検出される点（瞬間）」とする．

以上で「環境や身体状態の変動に依らない運動の構造」の要素，すなわち全関節に対するプリミティブ及びイベントについて述べた．次に，これら要素間の関係について論じる．

前述した「手順」は，要素（プリミティブやイベント）の間の時間的な位置関係（前・後・同期）を定めるものである．この関係は，毎回のタスクで変わらない．一方，この位置関係には時間的な距離が含まれていないが，実際に制御する場合には時間的な距離が必要になる．この時間的な距離は，基本的に，毎回のタスクで変動する．環境や身体の状態が変動するからである．

プリミティブやイベントと言った要素間に時間関係（位置+距離）が与えられたとき，その関係を使ってプリミティブの出力タイミングを決めるわけだから，両者の間にはどちらを基準にするか，決められている必要がある．つまり，時間関係に伴って依存関係が存在する（なお，イベントとイベントの間にも時間的な位置関係が存在するが，これらの間に依存関係はない）．

以上をまとめると，次のようになる．運動の手順を構成する要素（プリミティブやイベント）の間には，毎回のタスクで変わらない構造が存在する．この構造を構成するのは，要素間の時間的な位置関係（前・後・同期）であり，これらの関係は，それぞれが依存関係を含む（イベントとイベントの組合せを除く）．一方，毎回のタスクで，要素間の時間的な距離は変動する．

2.3 プリミティブの形状

Kuniyoshi らの研究から得た知見より，運動タスクを実現する上では,（局所的な）制御信号の波形よりもむしろそのエネルギが重要なファクタとなっており，細かな形状を定める必要はないと考えられる．また，エネルギ投入のタイミングが重要であるから，信号のピークを明示できることが望ましい．これらのことから，トルクの時間積分の大きさを明示でき，かつ波形のピークを明示できるシンプルな制御信号の波形として，本研究では Fig.2 に示すような釣鐘型のトルク波形を用いることにする（これ以外の定義も可能である）．この波形を特徴付けるパラメタは，後方向の分散 Δt_{b}，前方向の分散 Δt_{f}，及びピーク *peak* であり，具体的にはピークを中心に，

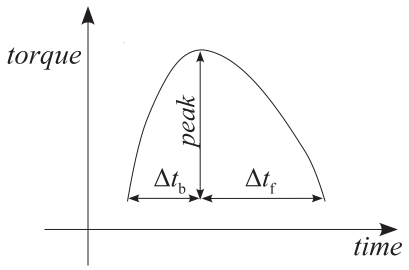


Fig. 2: Control primitive

$$T(\tau) = \begin{cases} peak \cdot \exp\left(\frac{-\tau^2}{2\Delta t_{\mathrm{b}}^2}\right) & (\tau < 0) \\ peak \cdot \exp\left(\frac{-\tau^2}{2\Delta t_{\mathrm{f}}^2}\right) & (\tau \geq 0) \end{cases} \quad (1)$$

によってトルクが計算される（τ はピークからの時間差）．

3 EventGraph: イベント駆動型制御の表現

本節ではイベント駆動型制御の表現，及びそれに基づいた制御の実行方法について述べる．

3.1 EventGraph

3.1.1 グラフ構造の構成

既に述べたように，運動を構成するイベントやプリミティブといった要素の間には，毎回のタスクで変化しない構造が存在し，個々の関係は二項関係であるから，イベントやプリミティブをノードとする，グラフ構造によってイベント駆動型制御を表現できることが分かる．このグラフ構造を EventGraph と名付ける．EventGraph のエッジは要素間の関係，つまり時間的な位置関係，時間的な距離関係，及び依存関係を表現することになる．

ところで，プリミティブは局所的な制御信号であるから，区間幅（時間幅）をもつ．また，2.3 項で述べたように釣鐘型の形状でピークの位置を明示するため，プリミティブは開始点，ピーク点，終了点の 3 つの点（これらを「制御点」と呼ぶことにする）から構成される．よって「プリミティブがイベントに依存する」といった表現は正確でなく,「プリミティブのピーク点がイベントに依存する」などと表現する必要がある．これより，エッジがつなぐのはプリミティブの個々の制御点及びイベント（点）である．

このように「点と点の関係」を基本要素として時間的な位置関係を表現することにより，区間幅をもつプリミティブ間のすべての時間関係を表現できるようになる．区間と区間の時間的な位置関係は，開始点と終了点の位置関係（前・後・同期）で分類す

ると 13 通りになるが，プリミティブの場合区間に中間点（=ピーク点）をひとつ含むため 63 通りになり，そのすべてを表現できる．

次に依存関係の表現だが，これはエッジに向きをもたせることで表現する．矢の先端が，もう一端（基準点）に依存している．依存関係は，基本的にはイベントからエッジを辿ることでわかるが，グラフがループしている場合などは知ることができない．

3.1.2 メトリック

要素間の時間的な距離関係は，一般的にはロボットの入力（例えばセンサーなどによる観測値）の関数で表現される．ロボットがある運動を実現するために出力すべき制御信号は，環境や身体の状態によって決定されるからである．そこで，イベント駆動型制御を汎用的に表現するため，この関数形式がどのようなものかは定めずに，何らかの関数としてエッジの要素に含める．これを「メトリック」と呼ぶことにする．

さらに，メトリックを実数全体（正・負・ゼロ）に対して定義することによって，前項で述べた時間的な位置関係（前・後・同期）をも表現することにする．このようにすることで，エッジは向きとメトリック（関数）で構成され，要素（プリミティブ，イベント）間の時間的な位置関係，時間的な距離関係，及び依存関係を表現できるようになる．

3.1.3 EventGraph の構成

以上の議論から，EventGraph はプリミティブ，イベント，エッジ（メトリックを含む）から構成される有効グラフである．エッジがつなぐのは点と点であるから，要素として「ポイント」を導入し，3 つのポイントとプリミティブをあわせて「制御ノード」，1 つのポイントとイベント（正確には，イベントの条件である）をあわせて「イベントノード」として定義する．

3.2 EventGraph に基づく制御

EventGraph はイベント駆動型制御の表現という位置付けだが，イベント-アクション型の表現という捉え型もできる．このとき，ほかの表現形式（ペトリネットなど）との主な違いは，アクションに区間幅を考慮していること，及びイベントに関連づけられたアクションが，イベントよりも時間的に前に実行されるという表現も許容していることである．このため「イベントの待機 →(遅延)→ 動作」という実行の流れでは制御信号を生成できない．

「イベントよりも前」という表現をもとにして制御信号を生成する（プリミティブの出力タイミングを決める）とき，イベントが発生するよりも前にプリミティブを出力しなければならないから，イベントの発生時刻を「予測」する必要がある．この「予測」とはイベントとイベントの時間的な距離の予測であるから，前節の定義に従うとエッジのメトリックの役割であり，具体的にはメトリックの関数によって実現される．

イベントの予測が正確になされ，かつ予測不可能な外乱が生じなかったら，各エッジのメトリックの具体的な値は，一度計算するだけでよい．しかし現実的には予測を正確にすることは難しく，外乱も生じる．一般的に，イベントが近付く程そのイベントが生起する時刻の予測が正確になるはずだから，予測の情報が更新されたときに，各プリミティブの生起すべき時刻を更新する必要がある．この更新は，グラフ構造を辿って伝搬される．

このように，イベントの予測と更新の伝播が，EventGraph に基づいて制御をするアルゴリズムの中核となる．

4 イベント駆動型制御に基づく運動学習とその評価

4.1 目的と方針

イベント駆動型制御に基づく運動学習とは，与えられた評価関数を最大にする EventGraph を求めることである．この学習は 1.3.2 で述べたように，(1) グラフ構造を固定したパラメタ最適化と (2) グラフ構造自体の最適化の 2 段階から成る (Fig.1)．

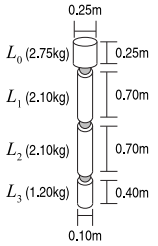
本研究の最終的な目標は，多様な動的環境下においてこれらの最適化を実現し，多自由度ロボットに運動を学習させることである．そこで本節では，人間と類似の身体性をもったロボットに，人間の定性的な運動知識に基づいて EventGraph の構造を与え，(1) のグラフ構造を固定した最適化を実現することで，イベント駆動型制御が制御則として有効に機能することを明らかにする．また，イベント駆動型制御が，身体の力学的拘束条件の変化をイベントとして利用することが妥当であることを実証する実験を行う．さらに，(2) のグラフ構造自体の最適化についての基礎的な検討として，ロボットの身体性とグラフ構造をともに変化させた場合のタスクの達成度を調べる．

具体的な実験としては，トランポリン上で，4 リンクの多自由度ロボット（Fig.3）に跳躍運動を学習させる．トランポリンは，ロボットの運動によって状態（位置や速度）が変化する，動的な環境であり，このため 1 節で述べたような，制御の困難さがある．

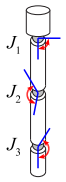
なお本実験は，Featherstone の順動力学アルゴリズム [8] をもとに，トランポリン（平面で大幅に変形しない）と多自由度ロボットの動力学シミュレーション環境を構築して行った．

4.2 グラフ構造を固定したパラメタ最適化の手法

「構造を固定した最適化」は，EventGraph のエッジやノードの数，連結関係を固定し，個々のエッジ



(a) Size and weight of links



(b) Joint constraint

Fig. 3: 4-link robot structure.

のメトリックやプリミティブのパラメタをひとまとめにしたパラメタベクトルを $\mathbf{X}$ とおいたとき，

$$\mathbf{X}^* = \arg\max_{\mathbf{X}} \mathsf{Evaluate}(\mathit{EventGraph}) \qquad (2)$$

のように定式化される．なお，本実験では簡単のためにメトリックの関数形は定数項のみとし，この定数項を最適化対象のパラメタとする．

最適化手法としては，パラメタと評価関数の関係が不明であるから微分情報を用いられないこと，また，構造の最適化も視野に入れていることから遺伝アルゴリズム (GA) を用いた．GA はメタアルゴリズムであり，具体的な構成は，世代交代モデルには MGG (Minimal Generation Gap[9])，交叉オペレーションには SPX（シンプレックス交叉 [10]）を用いた．

4.3 グラフ構造を固定したパラメタ最適化の評価実験

ここでは 4.2 項で述べた，EventGraph の構造を固定した最適化を行い，解の収束性を調べることによって最適化が成功するか評価する実験を行う．

4.3.1 状況設定

達成すべき運動（タスク）を「一定の高さ[3]から落下させ，トランポリン上で跳躍し，できるだけ高い位置まで跳ぶ」という設定にした．このタスクを実現するために獲得すべきパラメタは毎回の状況が同じだから，4.2 で述べたようなメトリックやプリミティブ（関数形が定数）でもタスクの実現には影響を与えない．

4.3.2 EventGraph の構造

EventGraph の構造は人間の定性的な運動知識に基づいて Fig.4 のように与え，これを固定して 4.2 項で述べたパラメタ最適化を行う．「定性的な運動知識」と *EventGraph* の関係は，次のようなものである．

運動開始 (イベント E_S) 後，着地 (E_1) の前に足を曲げ (制御 C_{11}, C_{21}, C_{31})，着地の直後に足をふんばり (C_{12}, C_{22}, C_{32})，トランポリンが十分に沈みきってから足に力を加えて (C_{13}, C_{23}, C_{33}) 跳躍する．離陸 (E_2) したらバランスを取る制御 (C_{14}, C_{24}, C_{34}) をする．高さがもっとも高くなったら (E_E) 終了．

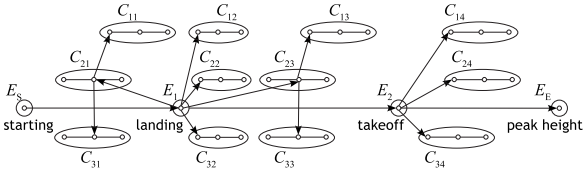


Fig. 4: *EventGraph* for trampoline hopping.

4.3.3 評価関数

評価関数は，跳躍後の高さと跳躍後のバランスで定義する．つまり，跳躍後の高さが高いほど評価値が高く，跳躍後のバランスが取れているほど評価値が高くなるようにする．また，L_0 の角速度，各関節の角速度が小さいほど評価が大きくなるようにした．具体的には，

$$\begin{aligned}\mathsf{Evaluate} = \frac{5}{2}\cdot \mathit{after\text{-}height} + 2(1+R_{33})^2 + \frac{4}{1+|\boldsymbol{\omega}_0|} \\ + \sum_{i=1}^{3} (1+\cos q_i)^2 + \frac{2}{(1+\dot{q}_i{}^2)}\end{aligned} \qquad (3)$$

とした．ここで *after-height* は跳躍後の高さを，R_{33} はエージェントの姿勢行列の (3,3) 要素を[4]，$\boldsymbol{\omega}_0$ はリンク L_0 の角速度ベクトル，q_i は関節 J_i の角度をそれぞれ表す．

4.3.4 実験結果と考察

1 回の学習を「GA の世代が 400 回交代するまで」と定義して[5]，学習を 15 回行った．各世代において個体がもつ評価値から最大のものを選び，15 回の学習での平均と分散を世代番号に対してプロットしたグラフを Fig.5 に示す．この結果から，初期の段階では初期値をランダムに与えているためにばらつきがあるが，最終的には収束する方向に向かっていることがわかり，学習が毎回のタスクで成功していると判断される．

学習によって得られたタスクの様子を，Fig.6 に示す（軸も表示されている）．この図からも運動を

[3]具体的には，トランポリンからロボットの頂点までの距離が 2.7m.

[4]Z-Y-X オイラー角表現における x 軸回転を φ，y 軸回転を θ としたとき $R_{33} = \cos\theta\cos\varphi$ である．

[5]1 個体にひとつの EventGraph が割り当てられており，評価値はタスクを 1 回行って計算する．また，1 回の世代交代において 8 個の子孫を生成している．「400 回」という数字は，何度か予備実験を行って判断した．

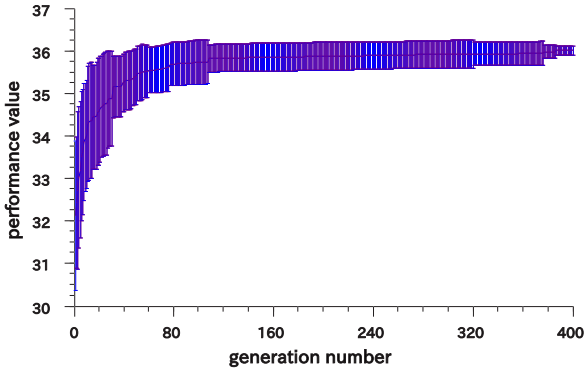


Fig. 5: Result of learning trampoline hopping.

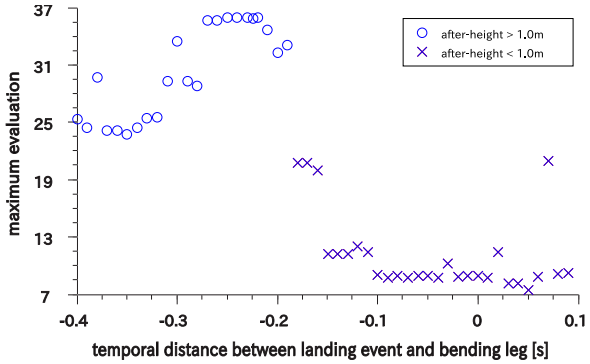


Fig. 7: Temporal distance between landing event and bending leg – maximum evaluation graph. O means *after-height* $> 1.0[m]$, and X means *after-height* $\leq 1.0[m]$.

実現できていることが確認され，イベント駆動型制御の有効性を確かめられたと言える．

4.4 拘束条件の変化の利用に関する実証実験

1.3.1 で述べたように，イベント駆動型制御では，身体の力学的拘束条件が変化する前後で，制御信号が運動に与える影響が大きく異なるという事実を利用している．ここでは，その実証実験を行う．

具体的には，4.3 項と同じ状況・タスクで，着地イベント (E_1) と屈伸アクション (C_{21}) をつなぐメトリックを変化させ，それぞれについてそのほかのパラメタを最適化し，メトリックに対する最大の評価値を調べた．その結果を Fig.7 に示す．

メトリックが -0.2s あたりで急激に評価値が減少しているのは，屈伸アクションが着地イベントよりも前に終了していないためである．このことは，イベントの前後でプリミティブが運動に与える影響が大きく異なることを示している．この物理的原因は，着地イベントが力学的拘束条件の変化であるから，言い替えると微分方程式が不連続に変化しているからである．

4.5 EventGraph の構造最適化の基礎的検討

4.3 項の実験で用いた *EventGraph*（Fig.4）は，人間の定性的な運動知識に基づくものであり，必ずしも適切な知識をロボットに与えられる分けではない．したがって，この定性的な運動知識（EventGraph の構造で表現される）そのものも最適化する必要がある．本節では，そのための基礎的な検討を行う．

注目するべきであるのは，イベントの数及び種類，プリミティブの数，及びそれらの連結関係である．イベントについては，人間が最適に与えられると仮定して，残りの項目のうち，運動にもっとも大きく影響を与えるプリミティブの数について考察する．

プリミティブの数は，達成できる運動の複雑さに比例する．つまりプリミティブの数が多い程，細かな制御ができる．しかしその半面，数が多すぎると，パラメタの最適化に時間がかかったり，局所解に陥る可能性が出て来る．

そこで Fig.4 の EventGraph の，E_2 に接続されているプリミティブの数を 0 個，3 個，6 個に変化させ，それぞれについて関節 J_1 の可動範囲（関節制約）を $\pi/2$ から π まで変化させて，その各々についてパラメタ最適化を行うことにより，タスクの困難さ（可動範囲が大きい程困難になる），プリミティブの数に対する達成度（学習後の評価関数の値）を調べた．その結果を Fig.8 に示す．

このグラフより，可動範囲が小さい領域ではいずれのプリミティブの数でも十分学習できているが，可動範囲が大きい領域では，プリミティブが 0 個の場合達成度が低い．また，プリミティブの数を増やすと達成度は高くなっているが，プリミティブが 6 個の場合などはバラツキが大きく十分学習できていないと考えられる（あるいは局所解に陥っている）．このことより，グラフ構造の最適化においては，プリミティブの数をなるべく小さくすることが望ましく，数を増加させることが必ずしも適切ではないことが言える．

5 結論と課題

本研究では，動的な環境下で多自由度ロボットに運動学習をさせるため，人間の定性的な運動知識を直接的に反映可能な制御則，イベント駆動型制御を設計し，その表現として EventGraph を提案した．また，シミュレーションによる実験により動的環境下における運動学習に対する有効性を示し，構造最適化の基礎的な検討として，プリミティブの数の選び方について指標を見出した．これらのことから，イベント駆動型制御に基づく運動学習は有効な手段と

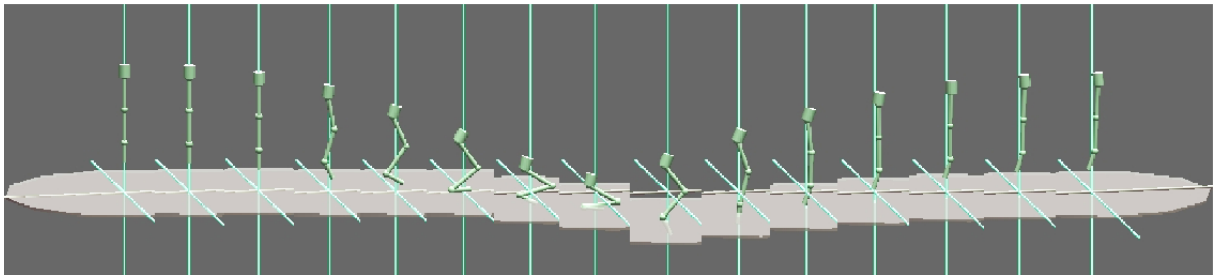


Fig. 6: A series of images in trampoline hopping motion (0.1s per frame).

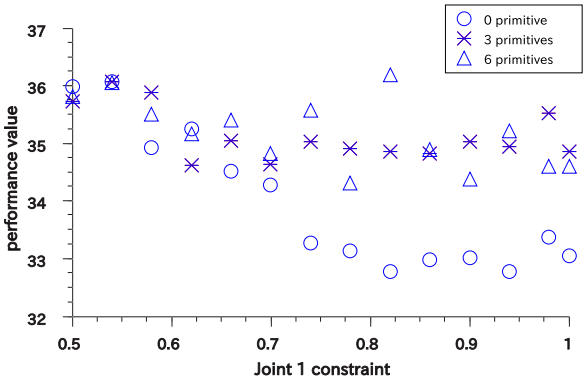


Fig. 8: J_1 constraint – performance-value graph for 0 primitive, 3 primitives, and 6 primitives.

なり得ると考えられる．

今後の課題としては，(1) イベントの抽出をどうするか，(2) 構造最適化のアルゴリズム（例えば遺伝的プログラミング），(3) プリミティブの形状，(4) より高速な学習などがあり，検討を進めている．また，次のステップとして,「動的な運動調整」がある．例えば跳び箱運動を考えたとき，踏み切り板に到達する前に歩幅を調節するが，この実現には，ロボットがダイナミクスの知識をもっていなければならない．ロボットを含む系の運動方程式は，接触状態の変化などによって不連続に変化するため，このようなダイナミクスの知識は定常的な表現ができず，環境と相互作用して決定するしくみが必要である．この動的な運動調整が実現できれば，グラフ構造に対するシンボリックな演算を物理世界に適切に反映させることができ，歩幅だけでなくステップ数のような高次の調整もできるようになると思われる．

参考文献

[1] Y. Kuniyoshi, Y. Ohmura, K. Terada, A. Nagakubo, S. Eitoku and T. Yamamoto: “Embodied basis of invariant features in execution and perception of whole body dynamic actions - knacks and focuses of roll-and-rise motion”, The Second International Workshop on Man-Machine Symbiotic Systems, pp. 289–301 (2004).

[2] M.Ishikawa, A.Neki, J.Imura and S.Hara: “Energy preserving control of a hopping robot based on hybrid port-controlled hamiltonian modeling”, IEEE Conference on Control Applications (CCA2003), No. CF002507 (2003).

[3] 美多：“非線形制御入門：劣駆動ロボットの技能制御論”, 昭晃堂 (Nov., 2000).

[4] 鈴木, 中村:“２階非ホロノミックシステムの平均化法による解析”, the 14th Annual Conference of the Robotics Society of Japan (1996).

[5] 比留川, 加賀美：“ヒューマノイドの知能—個としての知能—”, 日本ロボット学会誌, **20**, 5, pp. 478–481 (2002).

[6] 木村, 山下, 小林：“強化学習による 4 足ロボットの歩行動作獲得”, 電気学会電子情報システム部門誌, **122-C**, 3, pp. 330–337 (2002).

[7] 吉田, 水内, 稲葉, 國吉, 井上：“脊椎構造を持つ人間型ロボットにおける GA による匍匐動作の自動獲得”, 日本機械学会ロボティクス・メカトロニクス講演会’02 講演論文集, pp. 2P2–L04 (2002).

[8] R. Featherstone: “Robot Dynamics Algorithms”, Kluwer Academic Publishers (1987).

[9] 佐藤, 小野, 小林：“遺伝的アルゴリズムにおける世代交代モデルの提案と評価”, 人工知能学会誌, **12**, 5, pp. 734–744 (1997).

[10] 樋口, 筒井, 山村：“実数値 GA におけるシンプレックス交叉の提案”, 人工知能学会論文誌, **16**, 1, pp. 147–155 (2001).