

ニューラルネットワークによる 4脚歩行ロボットの障害物回避行動の獲得

山口 智浩*

Acquisition of Obstacle Avoidance Action for Quadruped Robots Using Neural Network

Tomohiro YAMAGUCHI*

1. はじめに

近年、産業用ロボットが工場に導入されて、生産性の向上に寄与していることは周知の事実である。当初、定位置に据え付けられるロボットが中心であったが、その後無人運搬車などの導入も進み、移動機能を持ったロボットもその重要性が認識されるようになってきた。今後さらに高度な機能を持ち、その作業内容や作業対象が広がって行くと、その作業領域も工場内だけに留まらず、人間の居住空間や自然環境下まで広がって行くものと思われる。また、原子力プラント内や災害現場、惑星探査といった人間が行くことができない場所で作業を行うロボットも期待されている。このようなロボットでは、柔軟で多様性のある移動機能が必要となる。例えば、不整地での移動や階段昇降、狭所でのその場旋回や横移動等の全方向移動、パイプや配線等の脆弱な障害物の跨ぎ越え、段差乗り越えや溝越え等の機能が要求される。

ロボットの移動機構としては、車輪、クローラ、脚が主に考えられる。車輪は機構および制御が簡単であり、エネルギー効率が良く高速な移動が可能であるが、平坦な路面でなければ移動できない。クローラは円環状の無限軌道履帯と車輪を組み合わせたもので、車輪より高い踏破能力を持っている。これらの2種類の移動機構は、基本的にロボットと地面との接点が連続して確保されなければならないという性質を持つ。これに対して、脚ではロボットと地面との接点が離散的であり、着地点を選択できる。さらには着地点の変更なしにロボット本体の位置・姿勢を変更することができるという特徴を有している。このため、脚式歩行ロボットは、平地においては移動効率

が車輪型やクローラ型より劣るものの、自分自身の高い自由度を有することにより、高い運動性能と対地適応能力を発揮することができる。¹⁾ 安定な接地点を離散的に選択して歩行するため、不整地においても安定な移動ができる。静止時には脚によって転倒しにくい姿勢をとることで安定な作業の足場を構成できるなど、いくつかの特異な機能性を有する。²⁾ しかし、反面歩行ロボットは高い自由度を有するため、歩容を決定することが非常に困難である。脚が単に固定された命令に従って動くだけのものである場合には、地面に対する適応能力が著しく制限されるだけでなく、安定な歩行を維持することさえ不可能になると考えられる。遊脚(地面に接地していない脚)の接地判断においても、指示された接地位置に障害物があるなどの理由で接地できない場合には、他の接地場所の候補から最適な接地位置を効率よく探索する必要がある。つまり、歩行ロボットでは地形状態を予測的に知覚・認識しそれに応じて適応的に地上の障害物を乗り越える等の脚の運び方³⁾を逐次決定する必要がある。これにより不整地移動や階段昇降を実現することができる。ただし、不整地移動や階段昇降の実現は、段差乗り越え可能な移動ロボットとしてクローラ型が早くから研究されているが、このクローラ型は全方向移動が十分ではなく、さらにパイプなどの脆弱な障害物を乗り越える際、自重でその障害物を破壊する危険性があるため旋回動作等により障害物を回避する必要がある。これに対し脚式歩行ロボットは障害物を非接触で跨ぎ越えることが可能であり、動物のような多様な移動機能を実現できる可能性を有している。⁴⁾ このことから、脚式歩行ロボットで障害物を回避する場合、車輪型やクローラ型とは異なりすべての障害物を旋回動作のみで避ける必要はなく、障害物の形状および自己状態によって、乗り越えや跨ぎ越えの動作を行うことにより障害物を回避することができる。また、それらの動作を行うことにより目的地までの移動効率が

*特別助手、電気電子情報工学科
Research Associate,
Dept. of Electrical, Electronics and Information Eng.

向上するものと考えられる。

本稿では乗り越えまたは跨ぎ越えを考慮した4脚歩行ロボットにおける障害物回避行動の獲得について検討する。4脚歩行ロボットの歩行では、支持脚(接地している脚)により構成される多角形内に常に重心をおき静的に安定を保つ静歩行と、静的には不安定であるが動的に安定を保つ動歩行とを実現することができる。また、その静歩行においては制御が容易であるため不整地歩行を実現し易いという特徴があり、動歩行においては移動速度や消費エネルギーの点で優れているという特徴がある。⁹⁾このため、段差などの障害物が存在する地形において4脚歩行ロボットの行動を獲得¹⁰⁾するためには、静歩行が適していると考えられる。ロボットの進行方向にある障害物がすべて乗り越え可能である場合は、前進運動のみで脚の接地位置を調整することにより回避可能である。しかし、障害物が壁で乗り越えられない場合は、前進と旋回、前進と横移動さらには前進と横移動および旋回などそれぞれの動作を組み合わせることで障害物を回避する必要がある。また、段差を乗り越える場合には脚の可動範囲により胴体の高さを調整する必要がある。さらには、障害物の端に脚を接地すると転倒する可能性がある。このため、障害物の位置などから歩行ロボットの移動距離と脚の接地位置を決定する必要がある。しかし、実機実験のみでロボットの移動距離を調整することは大変困難である。

そこで、本研究ではニューラルネットワーク(NN)の汎化能力に着目し、NNを用いて目的地とロボットの脚先および障害物の位置情報から縦と横の移動距離と旋回角を求める方法を提案する。NNの各設計パラメータの調整については、目的地と障害物の位置を変えた環境を幾通りか用意し、少ない歩行周期¹¹⁾で目的地へ移動するよう遺伝的アルゴリズム(GA)を用いてシミュレーションにより最適化を行う。ロボットが障害物を回避するためには、障害物を認識する必要がある。実機においては超音波センサを用いて障害物の位置を認識するが、シミュレーションでは障害物の初期位置を既知として与えておき、ロボットの移動に合わせて座標を変化させることにより、障害物の位置情報を更新し、これらの情報からロボットの行動を決定する。

また、4脚歩行ロボットの静歩行では遊脚の順番を切り替えることで自由歩容を実現できる。この自由歩容についてはすでに中村ら¹²⁾により、制約条件付きの最適化問題として静歩行での自由歩容計画(Free Gait Planning)が定式化され、モンテカルロ法による解法が提案されている。しかし、この方法から得られる結果は想定環境に特化され、探索環境外の摂動環境への汎化性がない。4脚歩行ロボットの静歩行では遊脚の順番は順列として24通り存在するが、リアルタイムでそれらすべてを試行する方法は大変効率が悪い。そこで、本研究ではロボットの移動量と自己状態か

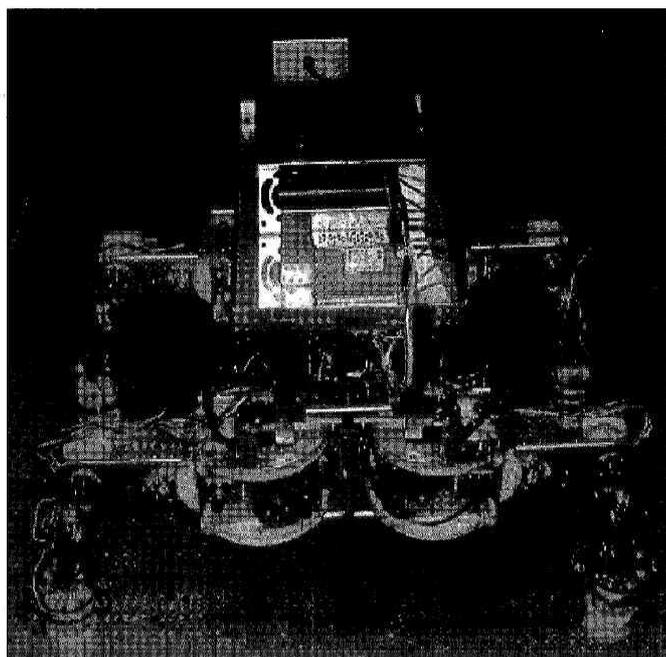


Fig.1 TITAN-VIII

らNNを用いて遊脚の順番をパターン認識的に決定する方法を提案する。このNNの各設計パラメータは胴体の移動が最小でかつ静歩行の安定条件を満足するようGAを用いて最適化を行う。訓練されたNNを用いて遊脚の順番を1歩行周期ごとに切り替えることにより自由歩容を実現できることをシミュレーションから明らかにする。さらに、この遊脚の順番決定用のNNを用いて行動決定用NNを再学習することにより、自由歩容を考慮した障害物回避行動を獲得する方法を示し、その有用性をシミュレーションで検証する。

2. 4脚歩行ロボットの概要

本研究では、4脚歩行ロボットとしてTITAN-VIII¹⁰⁾を用いる。TITAN-VIIIは移動の際のエネルギー消費を最小とすべくGDA (Gravitationally Decoupled Actuation)¹³⁾の方針で設計され、視覚センサや様々なセンサを用いて行動するのに適し、自由度と搭載能力を兼ね備えた脚式歩行ロボットの研究用プラットフォームとして、東京工業大学広瀬研究室より開発された。

TITAN-VIIIの写真を図1に示す。TITAN-VIIIは出力50 [W]のDCモーター3個で駆動する3自由度の脚を4本持ち、各脚を進行方向に対して横方向に張り出した歩行姿勢を基準歩行姿勢として歩行する。基準姿勢時の大きさは全長402 [mm]、幅602 [mm]、高さ273 [mm]、重さは22 [kg]である。各脚は基準姿勢時の状態で、鉛直方向に12~15 [kgf]程度の力を連続して発生することができ、デューティー比0.5の時0.9 [m/s]、0.75の時0.3 [m/s]程度の速度で歩行に対応した往復運動を行うことができる。このTITAN-VIIIを用い

て歩行実験を行うためには運動学を導出する必要がある。

さらに、TITAN-VIIIを用いて安定した歩行を実現するためには、歩行の安定性を考慮する必要がある。

2.1 4脚歩行ロボットの運動学

4脚歩行ロボットTITAN-VIIIの制御は、各関節の角度指令により行う。このため、指令値となる各関節角を導出する必要がある。脚については3自由度のマニピュレータとして運動学を導出することができる。

TITAN-VIIIの1つの脚に対する座標系をFig.2に示す。この座標系からTITAN-VIIIの脚の順運動学および逆運動学を求める。ここで、リンクの長さは $l_0 = 45.0$ [mm]、 $l_1 = 155.0$ [mm]、 $l_2 = 200.0$ [mm]であり、 l_3 は超音波センサ等を取り付けているために130.0[mm]となる。

順運動学では、絶対座標系 Σ_0 を設けて関節角 θ (θ_{0i} , θ_{1i} , θ_{2i})を与えた時の肩 r_{si} (x_{si} , y_{si} , z_{si})から脚先 r_{fi} (x_{fi} , y_{fi} , z_{fi})までの相対位置の導出を行う。順運動学の式は

$$r_{fi} = \begin{bmatrix} x_{fi} \\ y_{fi} \\ z_{fi} \end{bmatrix} = \begin{bmatrix} (l_0 + l_1 \cos \theta_{1i} + l_2 \cos \theta_{2i}) \cos \theta_{0i} + x_{si} \\ (l_0 + l_1 \cos \theta_{1i} + l_2 \cos \theta_{2i}) \sin \theta_{0i} + y_{si} \\ l_1 \sin \theta_{1i} + l_2 \sin \theta_{2i} - l_3 + z_{si} \end{bmatrix} \quad (1)$$

となる。ここで、添え字*i*は脚の番号を表す($i=1, 2, 3, 4$)。

また、逆運動学では、絶対座標系 Σ_0 で脚先座標 r_{fi} (x_{fi} , y_{fi} , z_{fi})、肩の座標 r_{si} (x_{si} , y_{si} , z_{si})を与えた時の関節角 θ (θ_{0i} , θ_{1i} , θ_{2i})の導出を行う。逆運動学の式は

$$\begin{bmatrix} \theta_{0i} \\ \theta_{1i} \\ \theta_{2i} \end{bmatrix} = \begin{bmatrix} \arctan \frac{y_{fi} - y_{si}}{x_{fi} - x_{si}} \\ \arctan \frac{d_{1i}}{\pm \sqrt{r_i^2 - d_{1i}^2}} - \phi_i \\ \arctan \frac{d_{2i}}{\pm \sqrt{r_i^2 - d_{2i}^2}} - \phi_i \end{bmatrix} \quad (2)$$

となる。ただし

$$\begin{cases} r_i = \sqrt{\left(\frac{x_{fi} - x_{si}}{\cos \theta_{0i}} - l_0\right)^2 + z_i^2} \\ \phi_i = \arctan \frac{1}{z_i} \left(\frac{x_{fi} - x_{si}}{\cos \theta_{0i}} - l_0\right) \\ d_{1i} = \frac{1}{2l_1} \left\{ \left(\frac{x_{fi} - x_{si}}{\cos \theta_{0i}} - l_0\right)^2 + z_i^2 + l_1^2 - l_2^2 \right\} \\ d_{2i} = \frac{1}{2l_2} \left\{ \left(\frac{x_{fi} - x_{si}}{\cos \theta_{0i}} - l_0\right)^2 + z_i^2 - l_1^2 + l_2^2 \right\} \\ z_i = z_{fi} - z_{si} \end{cases}$$

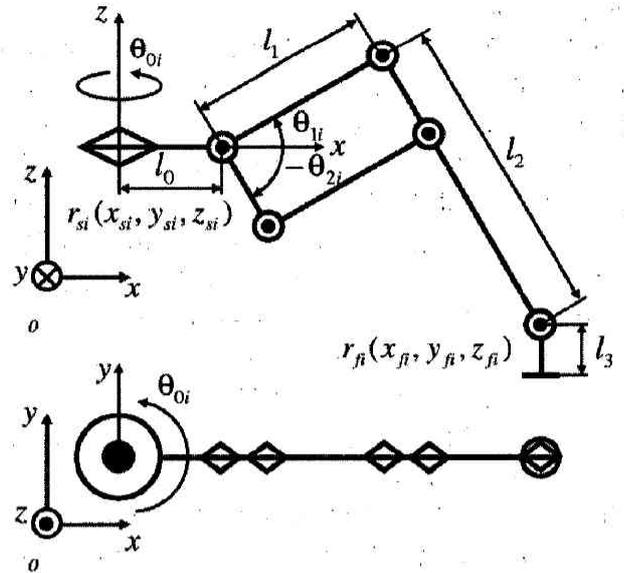


Fig.2 Coordinate system of TITAN-VIII's leg

である。

式(2)により、各脚先および肩の座標から各関節角が求められるが、各関節の可動角 [deg] 範囲は $(90.0 \leq \theta_{0i} \leq 245.0)$ 、 $(-65.0 \leq \theta_{02} \leq 90.0)$ 、 $(-115.0 \leq \theta_{03} \leq 270.0)$ 、 $(65.0 \leq \theta_{04} \leq -90.0)$ 、 $(-65.0 \leq \theta_{1i} \leq 65.0)$ 、 $(-65.0 \leq \theta_{2i} \leq 90.0)$ と定められていることを付記しておく。

ここで導出した運動学は実機であるTITAN-VIIIへ指令値として出力するための関節角の算出に用い、さらに、この運動学はシミュレーションでのロボットモデルとしても用いる。

2.2 静的安定性

歩行ロボットが歩行を行う上での重要な課題の一つとして、いかに安定な歩行が実現可能かということがあげられる。ここでは、4脚歩行ロボットにおける静的安定性の評価基準⁽¹²⁾について述べる。静的安定性の評価基準としては安定余裕⁽¹³⁾、転倒安定性⁽¹⁴⁾、Tipover Stability Margin⁽¹⁵⁾、Energy Stability Margin⁽¹⁶⁾、Dynamic Energy Stability Margin⁽¹⁷⁾などが提案されている。本研究ではこの中で最も基本的な評価基準である安定余裕を用いることとした。

安定余裕は移動体が水平面上に立っているという姿勢を前提とされた安定性評価であり、ロボットの重心の水平面への投影点から支持多角形の境界までの距離を s_i ($i=1, 2, 3$)としたとき、 s_i の最短距離を安定余裕と呼ぶ(Fig.3)。この重心の投影点が支持多角形の内部にあれば、静的安定性が保たれている。

また、ロボットの重心の水平面への投影点から、進行方向(負方向も含む)での支持多角形の境界までの距離を s_i ($i=1, 2$)とし、 s_i の最短距離を進行方向安定余裕と呼び(Fig.4)、前方の境界までの距離と後方の境界までの短い

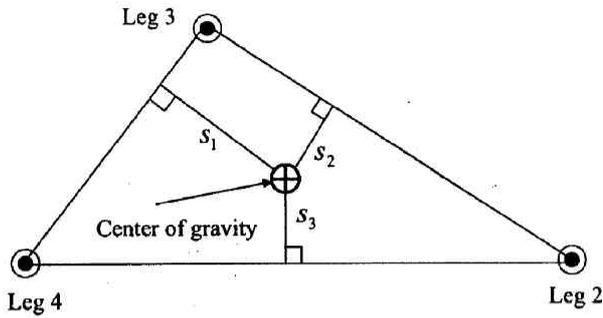


Fig.3 Stability margin

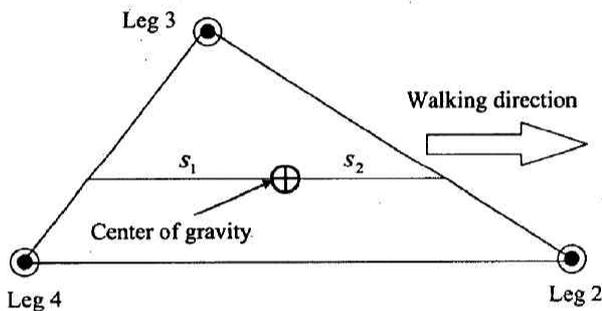


Fig.4 Longitudinal stability margin

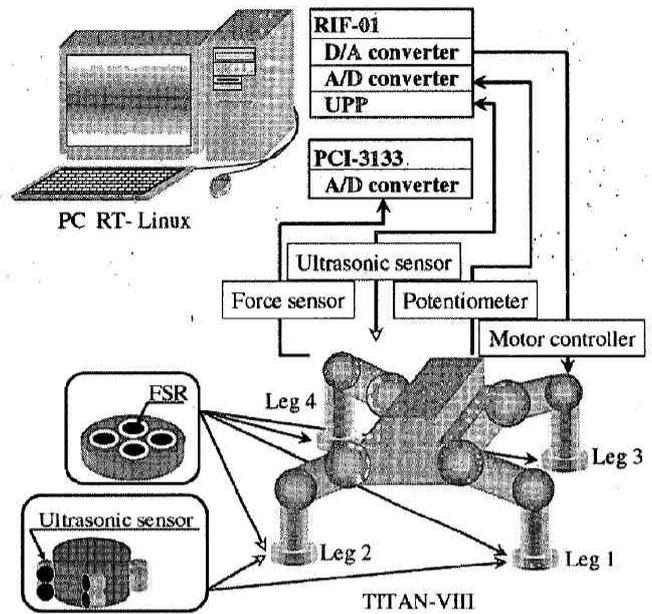


Fig.5 Robot control system

方である。

このため、歩行の安定性はロボットの重心が常に支持多角形の内部に存在するか否かにより判断する。

3. 障害物の知覚・認識

本研究で用いる実験装置のシステム構成をFig.5に示す。4脚歩行ロボットTITAN-VIIIには移動時の脚の関節角を検出するために、各関節にポテンシオメータが装備されている。脚先端部には各脚の接地状態を認識するために感圧抵抗素子(インターリンク株式会社, FSR Part # 402)を各脚に4個取り付けている。さらに、障害物を認識するために前脚の2脚の脚先端部に超音波センサ(NiceRa, T/R40-16)をそれぞれ3個取り付けている。ここで、ポテンシオメータの値は、ロボットインターフェイスボード(富士通, RIF-01)のA/D変換部、感圧抵抗素子の値は、A/D変換ボード(株式会社インターフェイス, PCI-3133)をそれぞれ通じてコンピュータに送られる。さらに、超音波センサではRIF-01のユニバーサル・パルスプロセッサ(UPP)部から超音波を発射して、障害物から反射して戻ってくる迄の時間を計測する。なお、コンピュータはRIF-01のD/A変換部を通してロボット本体のモータドライバ(岡崎産業, Titech moter driver)に関節角の指令値を送る。さらに、パソコンのOSは、リアルタイム処理を行うために、RT-Linuxを採用している。

移動ロボットで障害物を回避するためには、障害物の有無や高さなどを何らかの知覚情報を用いて認識する必要

がある。本研究で使用するロボットでは超音波センサにより障害物を知覚・認識する。しかし、超音波センサは障害物に対してほぼ垂直の位置で超音波を発射しないと障害物を正確には計測できず、ロボットの胴体に超音波センサを設置した場合、超音波を発射する方向が限定されることになる。本研究では、超音波センサを脚先に設置することにより、超音波を発射する方向を変化させロボットの進行方向に対して垂直でない障害物も検出できるようにしている。

3.1 超音波センサによる障害物の計測方法

超音波センサは前脚となる2脚の脚先端部にそれぞれ3個設置しており、超音波センサは進行方向にある障害物を検出するためにロボットの前方と両側の方向、さらに、接地面の凹凸を検出するために脚から下向きの方に設置している(Fig.6)。ここで、前方と両側の方向に設置した超音波センサは肩関節角 θ_0 によって超音波の発射方向がFig.7のように変化するため、ロボットの進行方向に対して垂直でない障害物も認識可能である。さらに、凸の段差に対して認識可能で、脚が遊脚の状態にある場合、Fig.8のように超音波センサに反応がない位置まで脚を上げることにより、乗り越え可否の判断および障害物の大まかな高さを計測することもできる。

また、脚から下向きの方に設置した超音波センサは脚先と地面との距離を計測するため、Fig.9のように脚先を一定の高さで動かした場合の地面との距離変化により障害物

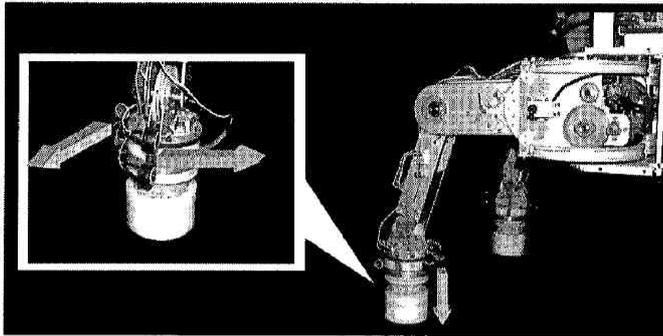


Fig.6 Directions of ultrasonic sensors

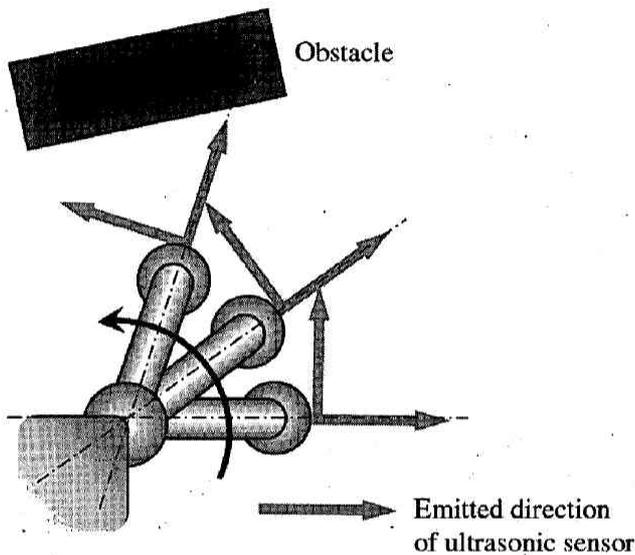


Fig.7 Recognition of the obstacles to X-Y plane

の端を認識することができる。さらに、脚先と地面との距離であるため、凹の障害物も認識でき、超音波センサの計測距離によって脚の接地可否を判断することができる。

3.2 実験環境

障害物認識の実験を行う環境をFig.10に示す。ここで、 y 方向を進行方向とし、4脚歩行ロボットの重心位置(0.0,0.0,0.3) [m]をスタート地点とし、重心が $y=1.2$ [m]に達したときゴールとする。ここでは、障害物の認識のみを行うため、歩行ロボットは y 方向への前進移動のみとし、旋回動作や横移動は考慮していない。このため、障害物はロボットが乗り越えられるものを用意している。ただし、障害物の乗り越えを行う場合、胴体の高さを一定とすると遊脚の高さよりも高い障害物に対しては乗り越えることができない。このため、胴体の高さは障害物の高さに合わせて変化させる必要がある。そこで、脚の接地位置が障害物の手前300 [mm]以内に来たとき胴体の高さを障害物の高さに合わせて上げている。遊脚の接地状態は脚底に設置している力覚センサにより判断する。ここで、4脚歩行ロボットの歩行については、本研究では超音波センサにより障害物を認識

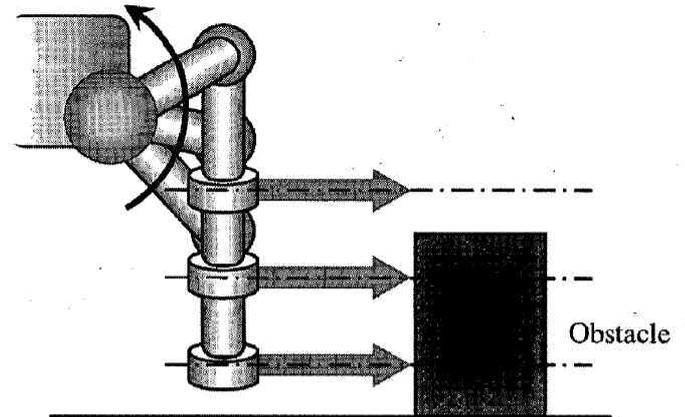


Fig.8 Recognition of the obstacles to Y-Z plane

するため、胴体が傾いてしまうと障害物までの距離を計測できない。4脚支持の場合は胴体の上下方向の揺れは起きないが、1本の脚が遊脚時に胴体移動を行うとロボットの重心位置によっては胴体が上下方向に揺れる可能性があるため、脚の遊脚時は胴体は移動しないものとする。

3.3 障害物認識

ロボットの胴体中心を原点とした際に検出される障害物の位置(x_{ob} , y_{ob})は、ロボットの前方向のセンサでは

$$x_{ob} = x_{si} + (l_0 + l_1 \cos \theta_{1i} + l_2 \sin \theta_{2i}) \cos \theta_{0i} - L_{so} \sin \theta_{0i} \quad (3)$$

$$y_{ob} = y_{si} + (l_0 + l_1 \cos \theta_{1i} + l_2 \sin \theta_{2i}) \sin \theta_{0i} + L_{so} \cos \theta_{0i} \quad (4)$$

となり、ロボットの両サイドのセンサでは

$$x_{ob} = x_{si} + (l_0 + l_1 \cos \theta_{1i} + l_2 \sin \theta_{2i}) \cos \theta_{0i} + L_{so} \cos \theta_{0i} \quad (5)$$

$$y_{ob} = y_{si} + (l_0 + l_1 \cos \theta_{1i} + l_2 \sin \theta_{2i}) \sin \theta_{0i} + L_{so} \sin \theta_{0i} \quad (6)$$

となる。ここで、添え字 i は超音波センサを取り付けている脚の番号を表しており、 $i=1,2$ である。このように、検出される障害物の位置は超音波センサにより測定された距離 L_{so} と胴体の位置、前脚の各関節角により算出することができる。

また、本研究で用いている超音波センサは取り付け位置から約1 [m]の範囲内に反射物があれば距離を計測することが可能である。しかし、この範囲はロボットが静止している場合に計測できる範囲であり、ロボット移動中に計測できる範囲は約0.6 [m]である。このため、超音波センサの取り付け位置から0.5 [m]を計測範囲と設定している。ただし、超音波センサを脚先に取り付けているため、実際にはロボットの胴体中心から約0.7 [m]の範囲で障害物を認識することが可能である。

歩行ロボットが直進移動した際に検出された障害物の位置をFig.12とFig.13に示す。Fig.12は、ロボットの前方と両側の超音波により検出した障害物の認識情報であ

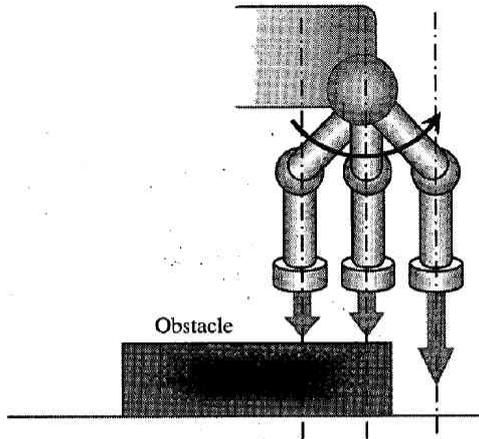


Fig.9 Recognition of the ground

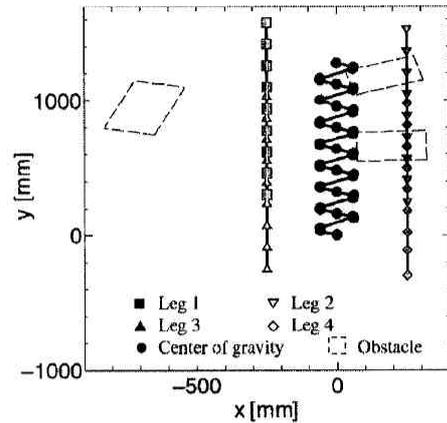


Fig.11 Movement path of quadruped robot

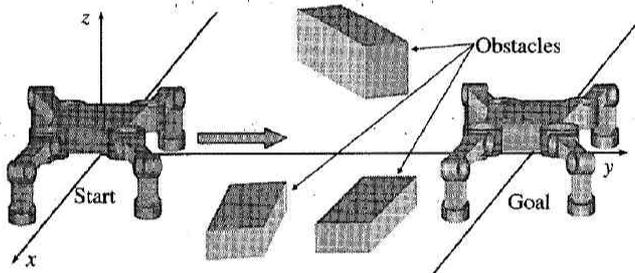


Fig.10 Setup of the environment

り、障害物の一辺を計測していることが確認できる。Fig.13は脚の下方の超音波センサにより検出された情報であり、障害物の上を脚が通過したときに障害物の両端の位置と高さを検出していることが確認できる。これらの情報を用いて次節では歩行ロボットの行動としての移動量および脚の接地位置の決定を行う。ただし、本研究の障害物認識には超音波センサを用いているため、円筒などの超音波をうまく反射できないものは認識できない。複雑な形状をした障害物を認識するためには視覚センサなどの他のセンサが必要となる。このため本研究では超音波センサで形状を認識することができる四角形のもののみを障害物として扱う。

3.4 障害物の座標設定

障害物と各脚先の座標はFig.14に示すように、ロボットの左前脚を (x_{f1}, y_{f1}) 、右前脚を (x_{f2}, y_{f2}) 、左後脚を (x_{f3}, y_{f3}) 、右後脚を (x_{f4}, y_{f4}) とし、ロボットに対して右側の障害物座標を $\{x_{o1}, y_{o1}\}, \dots, \{x_{o4}, y_{o4}\}$ 、左側の障害物座標を $\{x_{o1}, y_{o1}\}, \dots, \{x_{o4}, y_{o4}\}$ と設定する。ここで、障害物がロボットの前方にある場合、超音波センサでは障害物の1辺(or2-or3とo12-o13)を計測することができる。また、各脚先の位置はポテンショメータの値と運動学により求めることができる。このため、この1辺

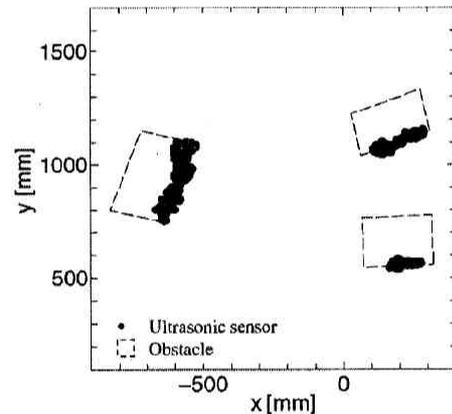


Fig.12 Recognition result of the obstacles in the forward direction

の情報と脚先の位置情報を用いて行動を決定する必要がある。ここで、障害物情報は脚2の超音波センサにより認識した障害物のx方向の最大値と最小値 $\{x_{or\ max}(k), x_{or\ min}(k)\}$ 、y方向の最大値と最小値 $\{y_{or\ max}(k), y_{or\ min}(k)\}$ 、z方向の最大値と最小値 $\{z_{or\ max}(k), z_{or\ min}(k)\}$ 、同様に脚1の超音波センサにより認識した障害物についてのそれらの値を $\{x_{ol\ max}(k), x_{ol\ min}(k)\}, \{y_{ol\ max}(k), y_{ol\ min}(k)\}, \{z_{ol\ max}(k), z_{ol\ min}(k)\}$ とする。

4. 4脚歩行ロボットの障害物回避行動決定

4脚歩行ロボットで障害物を回避する場合、前進と旋回または前進と横移動との組み合わせなど回避動作がいくつかが存在するため、NNを用いて目的地とロボットの脚先および障害物の位置情報から縦と横の移動距離と旋回角を求める方法を提案する。実際のロボットを用いた場合、センサから得られる情報は、障害物の位置、高さ、および4脚歩行ロボットの各関節角である。また、移動ロボットにおいては、目的地への到達も重要な課題の一つであるが、実際に用いるロボットからは目的地の情報を得ることができない。このため、目的地の情報について

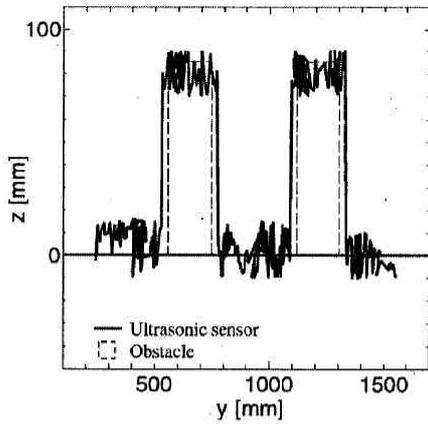


Fig.13 Recognition-result of the ground

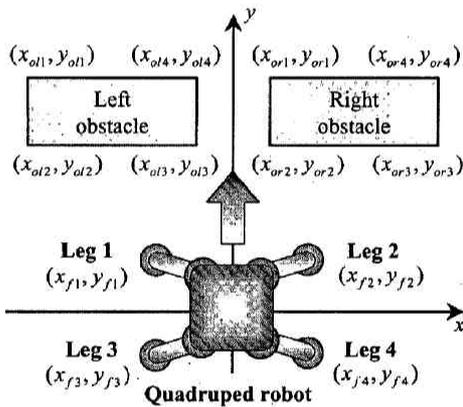


Fig.14 Coordinates of the robot's legs and the obstacles

てはあらかじめ与えてあるものとし、これらの情報をもとに歩行ロボットの移動量を決定する。ここで、4脚歩行ロボットの行動決定にはNNを用いるが、このNNの設計パラメータの調整には、少ない歩行周期回数でかつより目的地にロボットが近づきようGAを用いて最適化を行う。さらに最適化を行ったNNを実機実験に適用し、本手法の有効性を検証する。

歩行ロボットの行動決定にはFig.15に示すような3層のNNを用いる。NNへの入力値は、4脚歩行ロボットの各脚の位置 $\{x_{f1}(k), y_{f1}(k)\}, \dots, \{x_{f4}(k), y_{f4}(k)\}$ と胴体の高さ $Zr(k)$ 、ロボットに対して右の障害物のx方向の最大値と最小値 $\{x_{or\max}(k), x_{or\min}(k)\}$ 、y方向の最大値と最小値 $\{y_{or\max}(k), y_{or\min}(k)\}$ 、z方向の最大値と最小値 $\{z_{or\max}(k), z_{or\min}(k)\}$ 、同様に左の障害物についてのそれらの値 $\{x_{ol\max}(k), x_{ol\min}(k)\}$ 、 $\{y_{ol\max}(k), y_{ol\min}(k)\}$ 、 $\{z_{ol\max}(k), z_{ol\min}(k)\}$ 、さらにロボット重心から目的地までのx方向の距離誤差 $x_{de}(k)$ 、同じくy方向の距離誤差 $y_{de}(k)$ とロボットの進行方向に対する目的地方向の誤差角 $\theta_{de}(k)$ とする。

ここで、ロボットの各脚と障害物の位置は胴体中心を原点とした位置である。また、各出力は、歩行ロボット

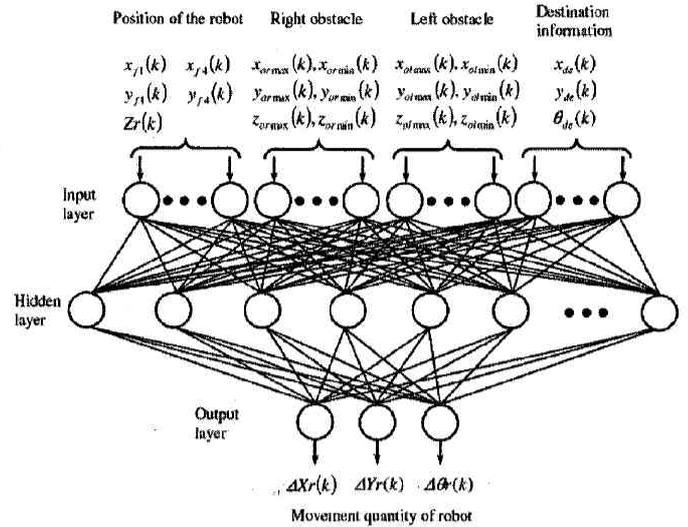


Fig.15 Three layered neural network for determining the action of the quadruped robot

のx方向、y方向の移動量 $\{\Delta Xr(k), \Delta Yr(k)\}$ および旋回角 $\Delta \theta r(k)$ とする。

また、ロボットが障害物を通過した場合など、ロボットの前方に障害物が存在しない場合の各障害物についての値は

$$(x_{or\max}(k), y_{or\max}(k), z_{or\max}(k))=(1.0, -0.99, 0.0)$$

$$(x_{or\min}(k), y_{or\min}(k), z_{or\min}(k))=(0.99, -1.0, 0.0)$$

$$(x_{ol\max}(k), y_{ol\max}(k), z_{ol\max}(k))=(-0.99, -0.99, 0.0)$$

$$(x_{ol\min}(k), y_{ol\min}(k), z_{ol\min}(k))=(-1.0, -1.0, 0.0) \text{ [m]}$$

の一定値とし、ロボットの後方に障害物が存在するものとしている。

本研究では、ニューラルネットワークの1つであるラジアル基底関数ニューラルネットワーク(Radial Basis Function neural network: RBFNN)^{18,19)}を用いる。このRBFNNは、非線形関数を円形の等高線を持つ基底関数で展開する方法であり、関数近似やパターン識別法として利用されている。一般の階層型NNと比較してRBFNNの利点は、ネットワークの一部だけを学習することが可能なこと、ユニットの配置に関するパラメータと重みのパラメータを別々の手法で学習可能であること、後者だけの問題ととらえれば簡単な線形式の最小2乗問題に帰着されることなどが挙げられる。しかし、一般にRBFNNではより多くのユニットが必要とされる問題点もある。

RBFNNの中間層のユニット関数には

$$\phi_i(\mathbf{x}) = \exp \left\{ -\frac{\|\mathbf{x}(k) - \mathbf{c}_i\|^2}{\sigma_i^2} \right\} \quad (7)$$

を用いる。ここで、 ϕ_i は中間層のi番目のユニットの出力であり、RBFの設計パラメータとして、各入力に対する中心値

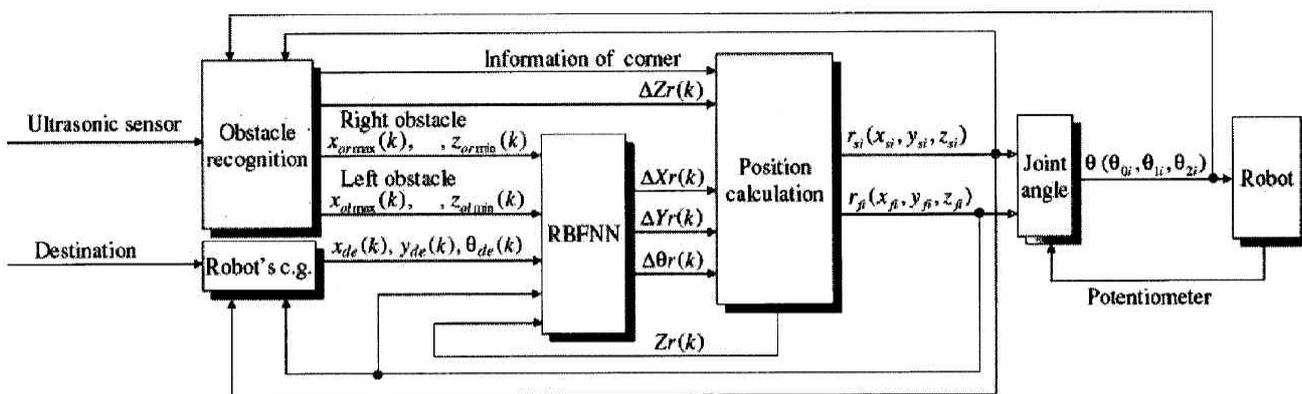


Fig.16 Obstacle avoidance control system with consideration to the destination

ベクトル C_i と標準偏差 σ_i を与える。出力層の j 番目のユニットの出力 o_j は

$$o_j(k) = \sum_{i=1}^m w_{ij} \phi_i(x) \quad (8)$$

とし、中間層の出力の線形結合により計算する。ここで、 w_{ij} は i 番目の中間層ユニットから j 番目の出力層ユニットへの結合荷重、 m は中間層のユニット数である。中間層のユニット数を入力数の約2~3倍に設定した場合のNNの学習結果では十分な結果が得られず、ユニット数を多く設定した場合は学習時間が長くなる。このため、中間層のユニット数は試行錯誤により決定した。その結果、入力数の約4倍のユニット数としたときが最も良い結果であったため、 $m=100$ と設定した。

4.1 目的地移動および障害物回避行動の獲得

本章で提案するRBFNNを用いた障害物回避制御システムの流れをFig.16に示す。障害物が存在した場合、旋回のみで回避するのではなく、乗り越えや跨ぎ越え動作によっても回避を行うため、もし胴体の高さを一定とすると遊脚の高さよりも高い障害物に対しては乗り越えや跨ぎ越えで障害物を回避することができない。そこで、障害物を認識した段階で胴体の高さを障害物の高さに合わせて変化させる。歩行移動の際に得られた障害物の位置情報、各脚先位置と胴体の高さおよび目的地の情報からRBFNNによりロボットの移動量を決定し、この移動量からロボットの肩の位置および脚先の位置を算出し、各関節角を求めロボットに出力する。

ここで、目的地までの距離誤差は肩の位置と脚先の位置からロボットの重心位置を求め、その変化量から目的地までの誤差距離の各値を更新する。障害物の位置情報は、実際のロボットでは超音波センサにより計測するが、シミュレーションではロボットの重心の変化量から障害物の位置情報を更新する。さらに、各脚はロボット

の移動量に合わせて変化させる。ただし、脚を地面に接地する場合は障害物の角から ± 50 [mm] の範囲を、また、障害物の上部に接地する場合は角から ± 30 [mm] の範囲をそれぞれ脚が接地できない範囲と設定し、脚の接地予定位置がこの範囲内にある場合は、接地可能な範囲内で接地予定位置から最も近い位置に設置する。

4.1.1 シミュレーション条件

シミュレーションを行う環境をFig.17に示す。ここで、 y 方向を進行方向とし、4脚歩行ロボットの重心位置 $(0.0, 0.0, 0.3)$ [m] をスタート地点とし、その重心が与えられた目的地の半径 0.2 [m] の円内の任意の地点に達したときをゴールとする。ここで、スタート時に与える目的地までの距離誤差は

$$(x_{de}, y_{de}) = (0.0, 1.8), (0.15, 2.2), (-0.15, 2.2) \text{ [m]}$$

の3種類とする。障害物は左右に1個づつ存在するものとする。ここで、ロボットの各脚の初期位置[m]は

$$(x_{f1}, y_{f1}) = (-0.25, 0.3), (x_{f2}, y_{f2}) = (0.25, 0.25)$$

$$(x_{f3}, y_{f3}) = (-0.25, -0.25), (x_{f4}, y_{f4}) = (0.25, -0.3)$$

とする。また、障害物の位置はロボットの胴体中心を原点とした場合の座標をNNの入力値とし、ロボットの移動に応じて障害物の x 座標、 y 座標は変化する。ここで、スタート時に与える障害物の x 座標、 y 座標はTable 1に示すような座標を用意し、それぞれを組み合わせる。障害

Table.1 x-and y-directional coordinates of obstacled

No	x_{or1}	x_{or2}	x_{or3}	x_{or4}	x_{ol1}	x_{ol2}	x_{ol3}	x_{ol4}
1	0.5	0.1	0.1	0.5	-0.5	-0.1	-0.1	-0.5
2	0.5	0.1	0.1	0.5	-0.5	-0.2	-0.2	-0.5
3	0.5	0.2	0.2	0.5	-0.5	-0.1	-0.1	-0.5
4	0.5	0.2	0.2	0.5	-0.5	-0.2	-0.2	-0.5
No	y_{or1}	y_{or2}	y_{or3}	y_{or4}	y_{ol1}	y_{ol2}	y_{ol3}	y_{ol4}
1	0.85	0.65	0.55	0.75	0.86	0.66	0.65	0.85
2	0.85	0.65	0.66	0.86	0.75	0.55	0.65	0.85

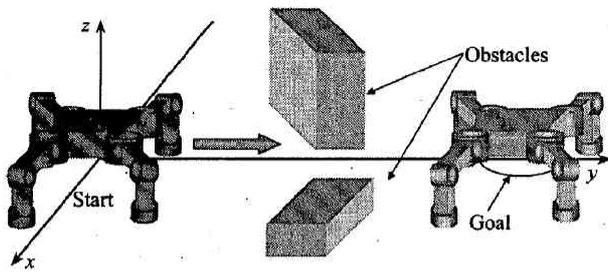


Fig.17 The environmental setup for the acquisition of quadruped robot's action

物の高さを表す Z_{or}, Z_{oi} は, $\{0.06, 0, 12, 0.3\}$ [m] の3種類をそれぞれ用意する. ただし, $Z_{or}(Z_{oi})=0.3$ [m] は越えることができない障害物であり, $Z_{or}=0.3$ [m] と $Z_{oi}=0.3$ [m] の組み合わせは行わないものとし, またどちらか片方は乗り越えられない障害物を設定する. このように, 位置や高さが異なる障害物を96通り用意しシミュレーションを行う.

歩行は, 常に3脚以上の脚で胴体を支持するクロール歩容を行い, 遊脚の順番は右後脚, 右前脚, 左後脚, 左前脚とする. クロール歩容のような静歩行で安定な歩行を実現するためには, 支持多角形の内部に重心があればよいとされている. このため, ロボットの重心が支持多角形の内部に常に存在するように胴体を移動させる. また, 胴体は地面に対して平行であると仮定する. ここでも前章と同様に, 障害物の座標と脚の接地位置により障害物の乗り越えを判断する.

4.1.2 適応度の設定

RBFNNの学習においては, 目的地に対して少ない歩行周期回数で移動するように, GA^[20,21]を用いてNNの結合荷重およびRBFの設計パラメータ(中心値, 標準偏差)の最適化を行う. ただし, 個体の適応度 $fitness$ は

$$fitness = \sum_{i=1}^{ob_n} (fitness_o + fitness_a + fitness_c) \quad (9)$$

とする. ここで, ob_n は最適化を行った環境の数である. $fitness_o$ は障害物への衝突を示す評価関数で

$$fitness_o = \begin{cases} 0.0, & \text{if there is no collision} \\ [x_{dc}^2(k) + y_{dc}^2(k)] \times 10, & \text{otherwise} \end{cases} \quad (10)$$

とし, 衝突した段階で歩行移動を中止する. また, $fitness_a$ は各関節角が歩行ロボットの許容範囲内であるか否かを示す評価関数で

$$fitness_a = \begin{cases} 0.0, & \text{if there is no outside of the range} \\ [x_{dc}^2(k) + y_{dc}^2(k)] \times 10, & \text{otherwise} \end{cases} \quad (11)$$

とし, ロボットの可動範囲を越えた段階で歩行移動を中

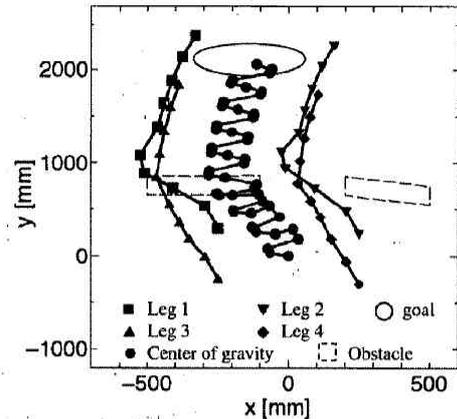


Fig.18 Movement path of quadruped robot in the case where a right obstacle is a wall

止する. さらに, $fitness_c$ は目的地に移動したか否かを示す評価関数で

$$fitness_c = [x_{dc}^2(k) + y_{dc}^2(k)] \times \frac{T}{50} \quad (12)$$

とする. ここで, T は安定した歩行で障害物を回避し, スタートから目的地までの移動に要した歩行周期回数である. ただし, 1つの環境に対する歩行周期回数の最大数を50回とし, 歩行周期回数が50回を越えたら歩行移動を中止する. また, GAパラメータの設定は個体数100, 選択の手法は10個体によるエリート選択と3個体によるトーナメント選択を併用, 交叉確率は0.6の一樣交叉とした.

4.2 シミュレーション結果

GAの最適化により得られたNNを用いて, 障害物回避のシミュレーションを行った. ロボットに対して左右それぞれに乗り越えられる障害物と壁を設定した場合の結果を以下に示す.

4.2.1 右の障害物を壁とした場合

スタート時のロボットの重心から目的地までの距離誤差を

$$(x_{de}, y_{de}) = (-0.15, 2.2) [\text{m}]$$

とし, 2つの障害物の x 座標, y 座標, z 座標を

$$(x_{or1}, y_{or1}) = (0.2, 0.85), (x_{or2}, y_{or2}) = (0.2, 0.65)$$

$$(x_{or3}, y_{or3}) = (0.5, 0.55), (x_{or4}, y_{or4}) = (0.5, 0.75)$$

$$(x_{oi1}, y_{oi1}) = (-0.5, 0.85), (x_{oi2}, y_{oi2}) = (-0.5, 0.65)$$

$$(x_{oi3}, y_{oi3}) = (-0.1, 0.66), (x_{oi4}, y_{oi4}) = (-0.1, 0.86)$$

$$z_{or} = 0.3, z_{oi} = 0.12 [\text{m}]$$

としたときの移動経路を Fig.18 に示す. そのときの胴体の高さや脚の接地位置を Fig.19 に示す.

右側の障害物が越えられない高さであるため, ロボットは左方へ移動していることが分かる. ここで, 右側の障害物に対しては, 脚1, 脚3ともに乗り越え動作により

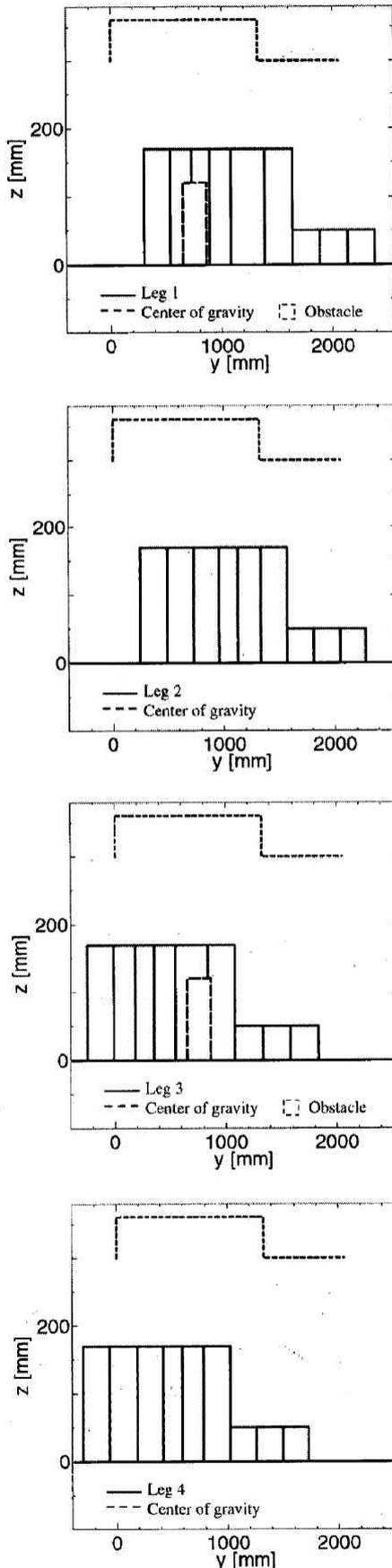


Fig.19 Motion of robot's leg in the case where a right obstacle is a wall

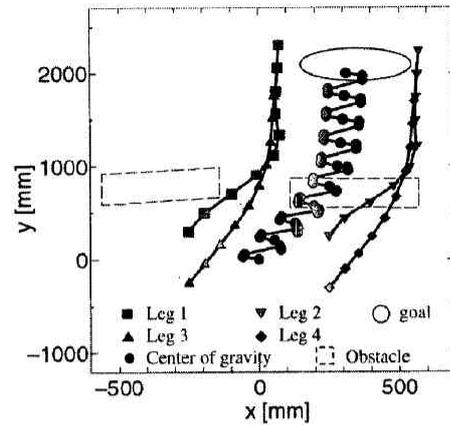


Fig.20 Movement path of quadruped robot in the case where a left obstacle is a wall

障害物を回避し、ロボットが目的地まで移動していることが確認できる。また、脚2、脚4は障害物の間を通過しているため、脚は地面に接地している。さらに、このときゴールまでに要した歩行周期回数は9回であり、ゴール時のロボットの重心から目的地点までの距離誤差は

$$(x_{de}, y_{de}) = (-0.037, 0.046) \text{ [m]}$$

であった。

4.2.2 左の障害物を壁とした場合

スタート時のロボットの重心から目的地点までの距離誤差を

$$(x_{de}, y_{de}) = (0.35, 2.1) \text{ [m]}$$

とし、2つの障害物のx座標、y座標、z座標を

$$(x_{or1}, y_{or1}) = (0.11, 0.87), (x_{or2}, y_{or2}) = (0.11, 0.56)$$

$$(x_{or3}, y_{or3}) = (0.57, 0.55), (x_{or4}, y_{or4}) = (0.57, 0.86)$$

$$(x_{ol1}, y_{ol1}) = (-0.56, 0.92), (x_{ol2}, y_{ol2}) = (-0.56, 0.59)$$

$$(x_{ol3}, y_{ol3}) = (-0.14, 0.66), (x_{ol4}, y_{ol4}) = (-0.14, 0.99)$$

$$z_{or} = 0.07, z_{ol} = 0.3 \text{ [m]}$$

としたときの結果をFig.20に、胴体の高さや脚の接地位置をFig.21に示す。このシミュレーションでは、左側の障害物が越えられない高さであるため、ロボットは右方へ移動していることが分かる。また、右側の障害物に対しては、脚2、脚4ともに乗り越え動作により障害物を回避し、ロボットが目的地まで移動していることが確認できる。ここでは、脚1と脚3は脚を地面に接地している。さらに、このときゴールまでに要した歩行周期回数は9回であり、ゴール時のロボットの重心から目的地点までの距離誤差は

$$(x_{de}, y_{de}) = (-0.048, 0.024) \text{ [m]}$$

であった。

4.3 実機実験

歩容はクロール歩容とし、歩行中は脚先に取り付けた超音波センサにより障害物を認識し、シミュレーションによ

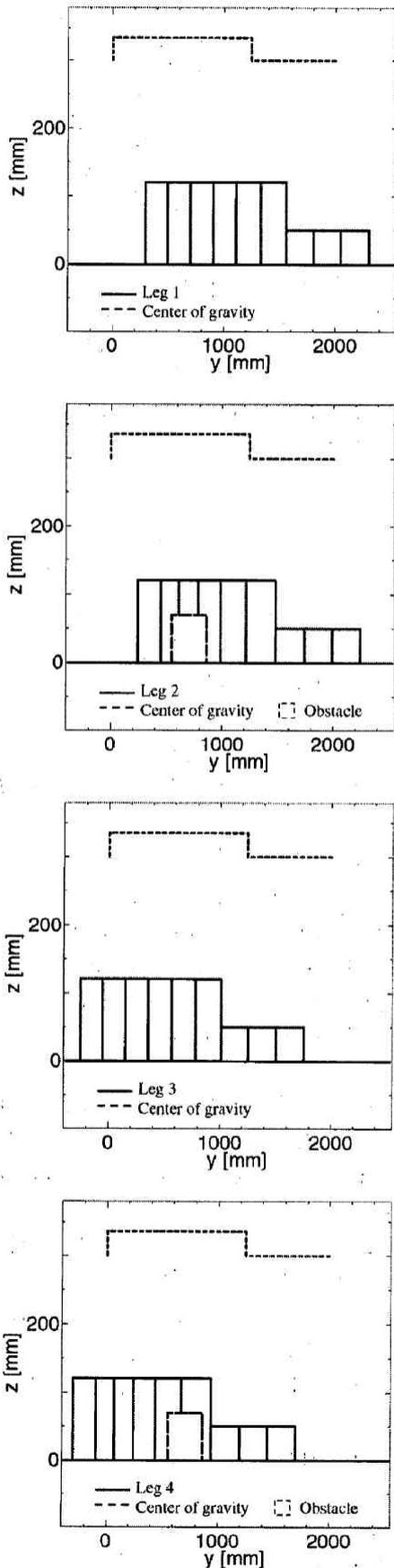


Fig.21 Motion of robot's leg in the case where a left obstacle is a wall

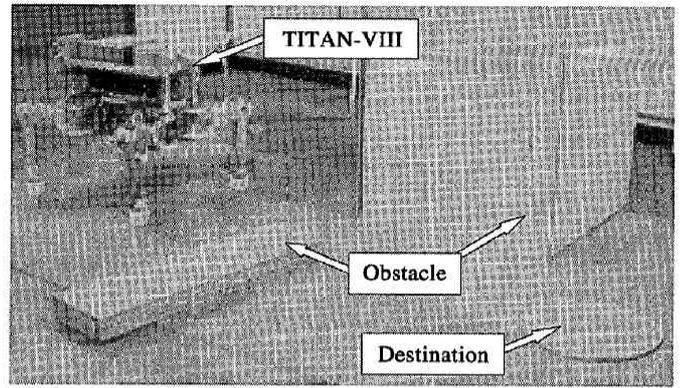


Fig.22 Environment of an experiment

り求めたNNを用いて歩行ロボットのx方向, y方向の移動量および旋回角を決定する。遊脚の接地の可否は、脚の下方の超音波センサにより計測された情報をもとに接地面が障害物の角であるか否かにより判断する。さらに、脚の遊脚および接地の状態は力覚センサにより判断する。このため、超音波センサと力覚センサの値および脚先の位置から障害物の乗り越えを判断することが可能である。ここで、遊脚の高さは初期状態で地面に接地している位置を高さ0 [mm]とする。

実験を行った環境をFig.22に示す。障害物はロボットの左右にそれぞれ1つずつ存在し、ロボットに対して左側の障害物は乗り越えられない障害物である。スタート時のロボットの重心から目的地までの距離誤差は

$$(x_{de}, y_{de}) = (-0.0, 1.9) \text{ [m]}$$

とした。このときの実験の様子をFig.23に示す。ここではFig.22を初期位置とし、Fig.23は1歩行周期で移動した後の状態を表している。このときゴールまで要した歩行周期回数は9回であり、このうち1回目の歩行は障害物を認識するためにy方向へ直進移動するものとしている。

本実験で歩行移動した際に検出された障害物の位置をFig.24とFig.25に示す。Fig.24は、ロボットの前方と両側の超音波により検出した障害物の認識情報であり、障害物のある一辺を検出している。Fig.25は、脚の下方の超音波センサによる検出結果であり、障害物の上を脚が通過したときに障害物の両端の位置と高さを検出している。これらの障害物に対するロボットの移動経路をFig.26に、胴体の高さや脚の接地位置をFig.27に示す。ここで、左側の障害物はロボットが乗り越えられない高さであるため、右方へ移動し、最終的には目的地である円内にロボットの重心が達していることが確認できる。右側の障害物は乗り越え可能であり、脚2, 脚4ともに障害物の上を脚を接地していることから障害物を乗り越えていることが確認できる。脚1と脚3は障害物の間を通過しているため、脚を地面に接地している。また、ゴール時の

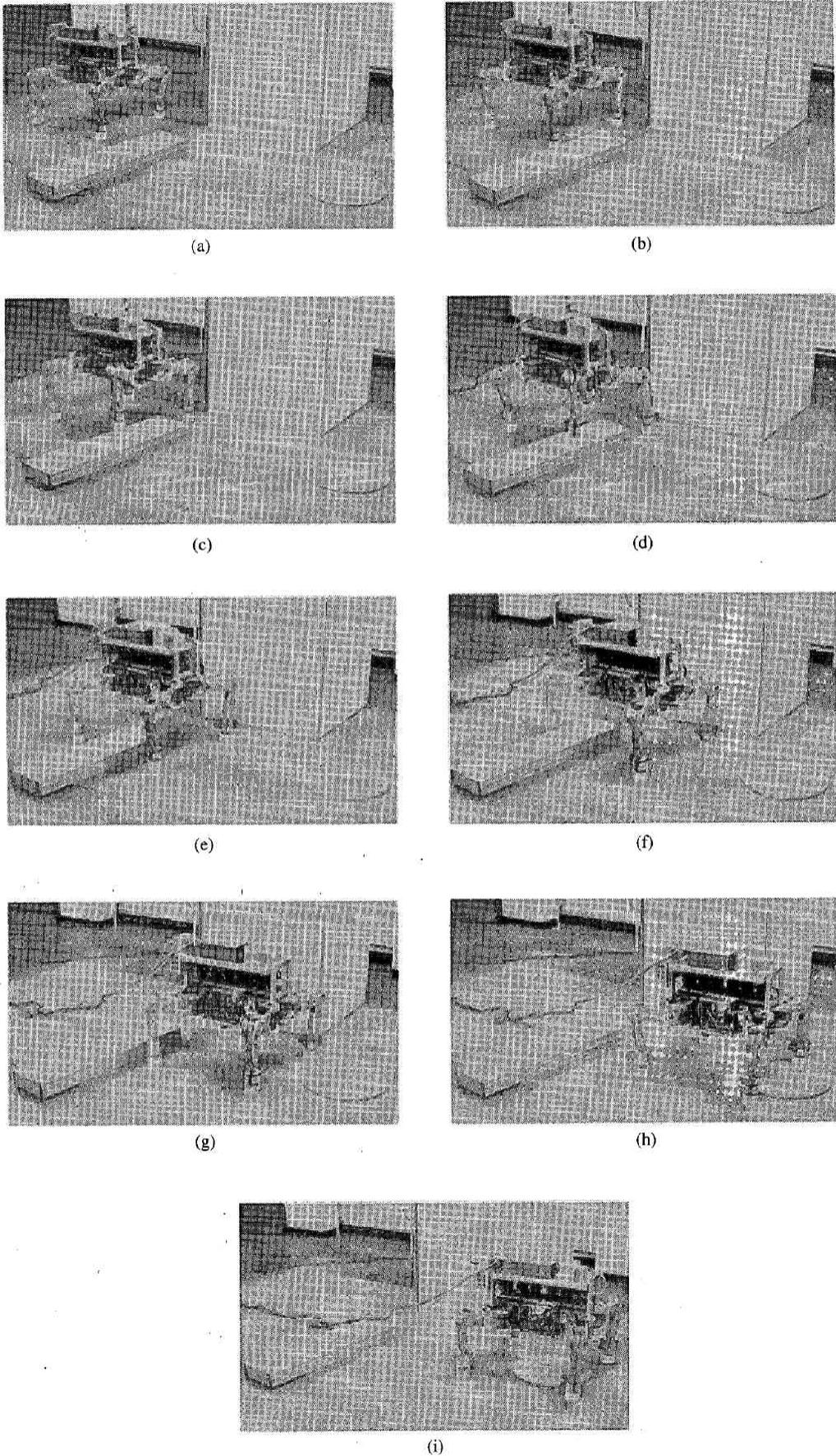


Fig.23 Experimental view of obstacle avoidance with consideration to the destination

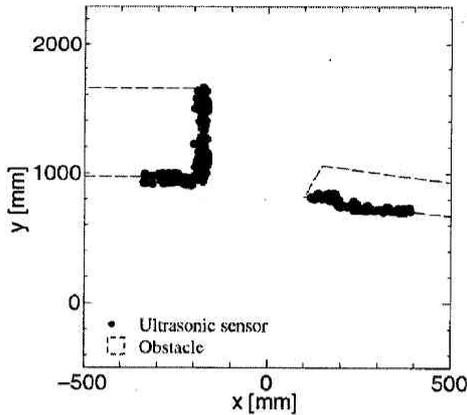


Fig.24 Recognition result of the obstacles in the forward direction

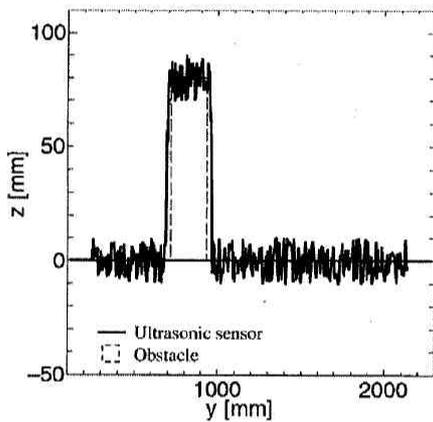


Fig.25 Recognition result of the ground

ロボットの重心から目的地点までの距離誤差は $(x_{de}, y_{de}) = (0.137, 0.075)$ [m] であった。

RBFNNの設計パラメータは複数の環境を用意し、少ない歩行周期回数で目的地へ移動できるようGAを用いて最適化を行い、シミュレーションにより目的地へ移動できる行動が獲得できた。さらに、最適化を行ったRBFNNを実機に適用し、障害物回避行動の実現を行った。しかし、本研究ではどちらか一方はロボットが乗り越えられる障害物としているため、ロボットの進行方向の障害物が完全な壁などの場合は障害物に衝突する可能性がある。障害物を回避するためのRBFNNへの入力にはロボットの胴体中心を基準としているため、スタート地点、ゴール地点は自由に変更可能ではあるが、本研究では、ゴール地点はロボットの前方にあり、さらにスタート地点からゴール地点の間には障害物(壁)が存在するものとしてロボットの行動を獲得しており、極端に異なる環境、たとえばロボットの後方にゴール地点を設定した場合などはその位置へ到達できない可能性がある。

また、ここでは4脚歩行ロボットの遊脚の順番を右後

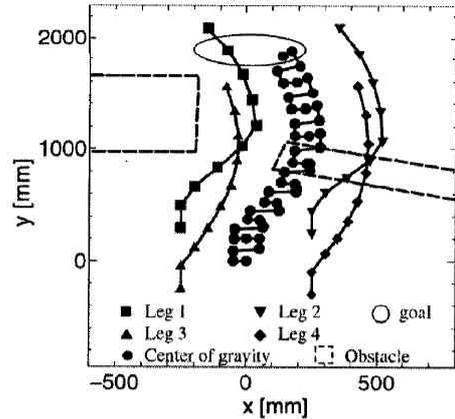


Fig.26 Movement path of quadruped robot in an actual experiment

脚、右前脚、左後脚、左前脚と一定にしたが、これによりロボットの移動範囲に若干の制約があるものと考えられる。このため、自由歩容を実現できるシステムを構築する。

5. 自由歩容となる遊脚の順番決定

4脚歩行ロボットの静歩行では遊脚の順番によって胴体の移動量が異なり、これによりロボットの移動可能範囲も異なる。このため、少ない胴体移動で静歩行の安定性が保たれれば各脚の可動範囲が大きくなり、これによりロボットの移動可能範囲も大きく設定することができると考えられる。このため、静歩行で移動する場合、歩行するごとに遊脚となるの脚の順番を決定する必要がある。4脚歩行ロボットの静歩行では遊脚の順番は順列として24通り存在する($P_4=4!$)。歩行するごとに24通りのすべての順番を試すことは可能ではあるが、効率が悪い。そこで、本研究ではNNを用いて遊脚の順番をパターン認識的に決定する。

遊脚の順番決定にはFig.28に示すような3層のNNを用いる。ここでNNの入力値は4脚歩行ロボットの胴体の高さ $Zr(k)$ 、歩行ロボットのx方向とy方向の移動量 $\{\Delta Xr(k), \Delta Yr(k)\}$ および旋回角 $\Delta \theta r(k)$ とする。また、出力はユニット数を24とし、24通りの遊脚の順番に番号をつけ、各ユニットと対応させる。4脚歩行ロボットへ出力する遊脚の順番は出力ユニットの中で出力値が1に最も近いユニットの番号を用いる。出力ユニットと遊脚の番号の対応表をTable2に示す。ここで、遊脚を表す脚の番号は左前脚を1、右前脚を2、左後脚を3、右後脚を4とする。

遊脚の順番決定においてもRBFNNを再び用いることとする。このRBFNNの設計パラメータ(中心値、標準偏差)の調整は教師信号の入力と出力の関係を満足するようGAを用いて最適化を行う。

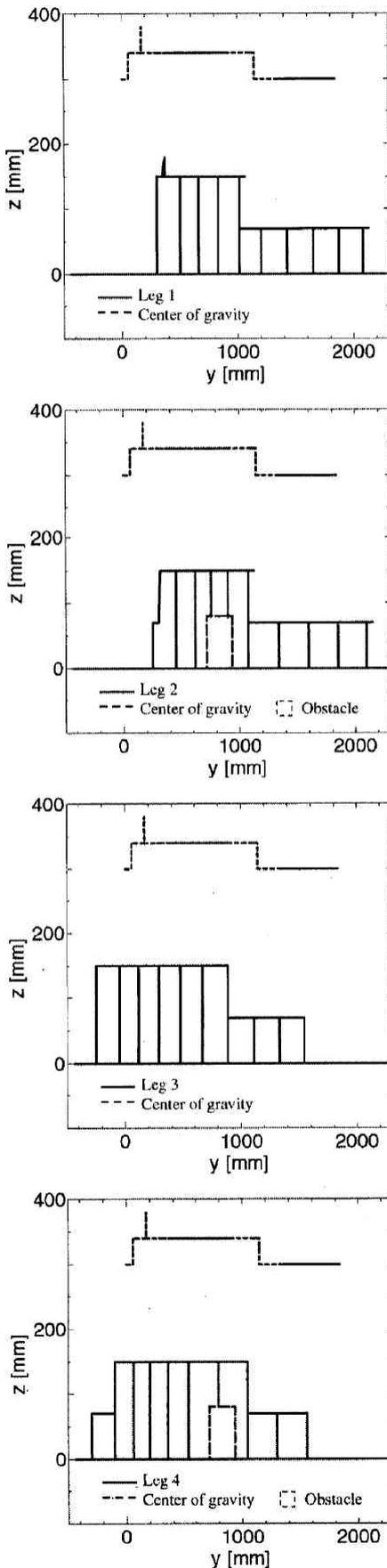


Fig.27 Motion of robot's leg in an actual experiment

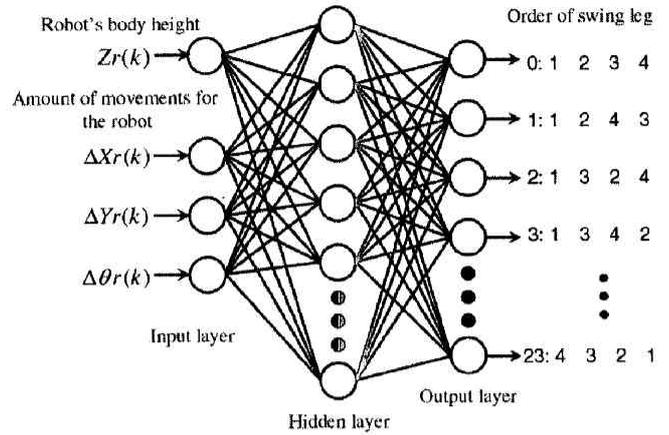


Fig.28 Three layered neural network for determining the order of swing leg

Table.2 Order of swing leg to each output of NN

Output number	Order of swing leg	Output number	Order of swing leg
0	1 → 2 → 3 → 4	12	3 → 1 → 2 → 4
1	1 → 2 → 4 → 3	13	3 → 1 → 4 → 2
2	1 → 3 → 2 → 4	14	3 → 2 → 1 → 4
3	1 → 3 → 4 → 2	15	3 → 2 → 4 → 1
4	1 → 4 → 2 → 3	16	3 → 4 → 1 → 2
5	1 → 4 → 3 → 2	17	3 → 4 → 2 → 1
6	2 → 1 → 3 → 4	18	4 → 1 → 2 → 3
7	2 → 1 → 4 → 3	19	4 → 1 → 3 → 2
8	2 → 3 → 1 → 4	20	4 → 2 → 1 → 3
9	2 → 3 → 4 → 1	21	4 → 2 → 3 → 1
10	2 → 4 → 1 → 3	22	4 → 3 → 1 → 2
11	2 → 4 → 3 → 1	23	4 → 3 → 2 → 1

5.1 遊脚の順番獲得

本研究で用いる教師信号は、ロボットの胴体の高さ $Zr(k)$ を300~370[mm], x 方向の移動量 $\Delta Xr(k)$ を-100~100[mm], y 方向の移動量 $\Delta Yr(k)$ を150~350[mm], 旋回角 $\Delta \theta r(k)$ を-15~15 [deg] の間で変化させた場合、胴体の移動が最小でかつ静歩行の安定性を満足する遊脚の順番をい1とし、それ以外の順番を0としている。ここで、ロボットの胴体の移動量が最小となる遊脚の順番は19通りあった。しかし、 $Zr(k)$, $\Delta Xr(k)$, $\Delta Yr(k)$, $\Delta \theta r(k)$ の変化量を一定にした場合、遊脚の順番によって選択される回数が異なる。そこで、選択される回数が多い場合は $Zr(k)$, $\Delta Xr(k)$, $\Delta Yr(k)$, $\Delta \theta r(k)$ の変化量を大きくし、選択される回数が少ない場合は移動の変化量を小さくして、それぞれの遊脚の順番に対して20個のデータを用意する。また、遊脚の順番を決定するRBFNNの設計パラメータ（中心値、標準偏差）の調整は入力と出力の関係を満足するようGAを用いて最適化を行う。 $Zr(k)$, $\Delta Xr(k)$, $\Delta Yr(k)$, $\Delta \theta r(k)$ をそれぞれ変化させたときに選択される遊脚の順

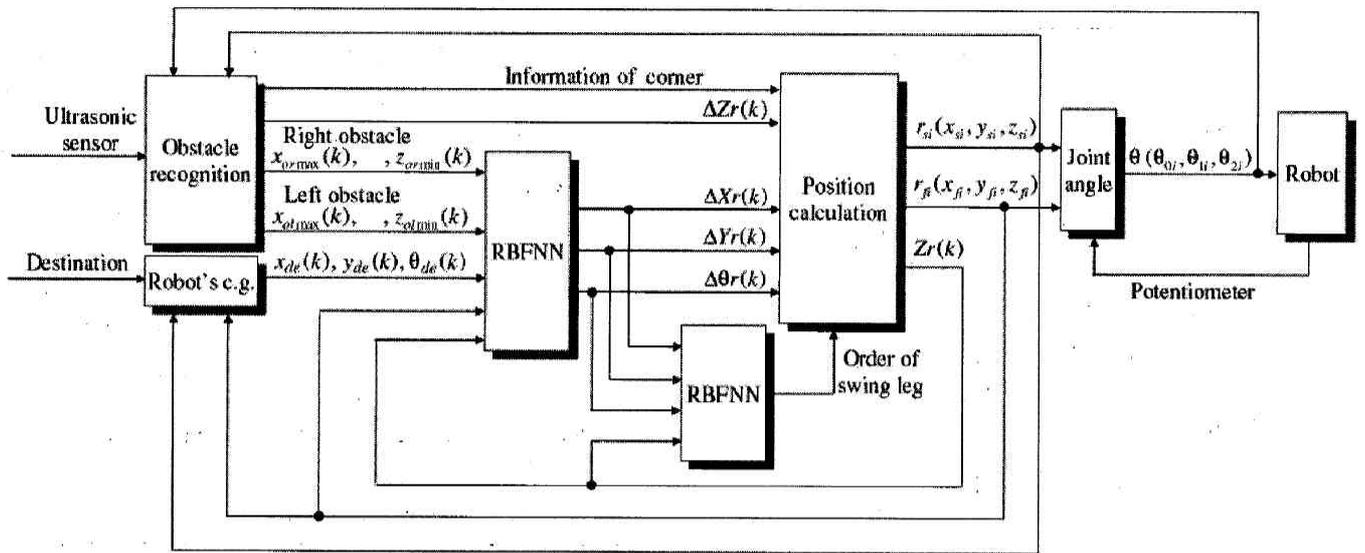


Fig.29 Obstacle avoidance control system with consideration to the order of swing leg

番は19通りであり、各遊脚の順番に対して20個のデータを用意している。今回、GAを用いて最適化を行った組み合わせは総計380通りである。ここで、GAの個体の適応度fitnessは

$$fitness = \sum_{i=1}^n (fitness_A + fitness_B) \quad (13)$$

とし、最小化問題として解を探索する。ここで n は最適化を行った組み合わせの総数である。 $fitness_A$ は教師信号 ts_j が1の場合に適用する評価関数であり

$$fitness_A = \sqrt{(o_j - ts_j)^2} \Big|_{j=j_1} \quad (14)$$

とする。ここで、添え字 j は出力層のユニット番号を表し、 j_1 は $ts_j=1$ となるユニット番号を表す。また、 $fitness_B$ は $ts_j=0$ の場合に適用する評価関数であり

$$fitness_B = \sqrt{\text{MAX}\{(o_1 - ts_1)^2, \dots, (o_j - ts_j)^2, \dots, (o_{24} - ts_{24})^2\}} \quad (15)$$

とし、 o_j と ts_j の差が最も大きかった値を評価する。

5.2 自由歩容を用いた障害物回避

本研究の障害物回避制御システムの流れをFig.29に示す。歩行移動の際に得られた障害物の位置情報、各脚先位置と胴体の高さおよび目的地の情報からロボットの縦と横の移動量および旋回角を決定し、このロボットの移動量から遊脚の順番を決定する。遊脚の順番を一定にした場合、すべての順番を試した場合、RBFNNを用いて順番を決定した場合それぞれについて検討する。

シミュレーション環境は前章と同じものを用いるため、 y 方向を進行方向、4脚歩行ロボットの重心位置(0.0,0.0,0.3)[m]をスタート地点として、その重心が与えられた目的地の半径0.2[m]の円内の任意の地点に達したときをゴールとす

る。ここで、スタート時に与える目的地までの距離誤差は

$$(x_{de}, y_{de}) = (-0.15, 2.2) [\text{m}]$$

とする。障害物は左右に1個ずつ存在するものとする。2つの障害物の x 座標、 y 座標は

$$(x_{or1}, y_{or1}) = (0.1, 0.85), (x_{or2}, y_{or2}) = (0.1, 0.65)$$

$$(x_{or3}, y_{or3}) = (0.5, 0.66), (x_{or4}, y_{or4}) = (0.5, 0.86)$$

$$(x_{ol1}, y_{ol1}) = (-0.5, 0.75), (x_{ol2}, y_{ol2}) = (-0.5, 0.55)$$

$$(x_{ol3}, y_{ol3}) = (-0.2, 0.65), (x_{ol4}, y_{ol4}) = (-0.2, 0.85) [\text{m}]$$

とし、障害物の高さである z 座標は

$$z_{or} = 0.12, z_{ol} = 0.3 [\text{m}]$$

とする。ここで、右の障害物はロボットが乗り越えられるが、左の障害物は乗り越えることができないものとする。このため、ロボットの胴体の高さである $Zr(k)$ は乗り越えられる障害物の高さに合わせて変化させる。

5.2.1 遊脚の順番を一定にした場合

遊脚の順番を一定にして障害物を回避したときの4脚歩行ロボットの移動経路をFig.30に、そのときの胴体の移動量 Br をTable3に示す。ここでは、遊脚となる脚は右後脚、右前脚、左後脚、左前脚の順番で移動するものとする。

5.2.2 24通りすべての順番を試した場合

24通りすべての遊脚の順番を試しながら障害物を回避したときの4脚歩行ロボットの移動経路をFig.31に、そのときの胴体の移動量 Br をTable4に示す。遊脚の順番を一定とした場合と比べ、胴体の移動量が小さくなっていることがとの比較から確認できる。

5.2.3 RBFNNにより遊脚の順番を決定した場合

学習したRBFNNを用いて遊脚の順番を決定し、障害物を回避したときの4脚歩行ロボットの移動経路をFig.32に、そのときの胴体の移動量 Br をTable5に示す。RBFNNによ

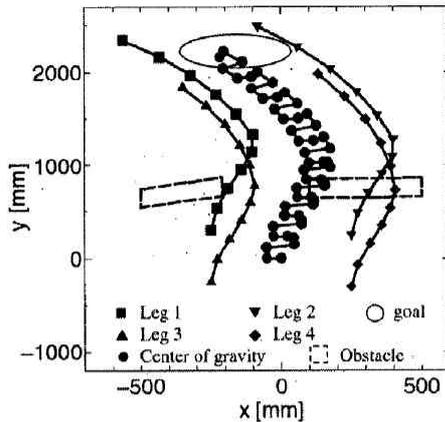


Fig.30 Movement path of the quadruped robot when fixing the order of swing leg

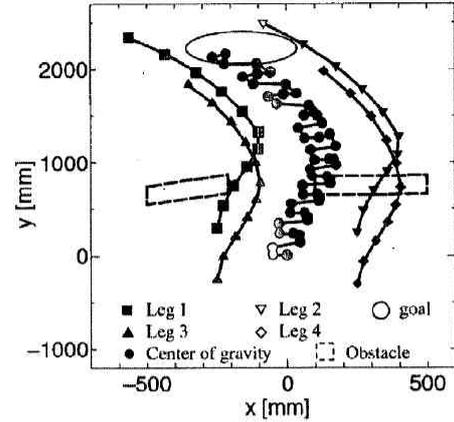


Fig.31 Movement path of quadruped robot when trying 24 kinds of the order

Table.3 The amount of movements of the robot's body when fixing the order of swing leg

$\Delta Xr(k)$ [mm]	$\Delta Yr(k)$ [mm]	$\Delta\theta r(k)$ [deg]	$Zr(k)$ [mm]	B_r [mm]	Order of swing leg
22.026	237.002	0.809	300.0	418.090	4→2→3→1
43.958	217.174	-0.818	360.0	372.904	4→2→3→1
45.347	200.484	-0.639	360.0	354.896	4→2→3→1
32.227	189.676	1.413	360.0	362.871	4→2→3→1
12.261	186.691	4.182	360.0	390.217	4→2→3→1
-15.944	241.234	3.894	360.0	520.653	4→2→3→1
-17.443	240.270	4.289	360.0	528.034	4→2→3→1
-18.958	239.279	4.695	300.0	535.676	4→2→3→1
-20.522	238.245	5.122	300.0	543.786	4→2→3→1
-22.187	237.142	5.586	300.0	552.667	4→2→3→1

Table.4 The amount of movements of the robot's body when trying 24 kinds of the order

$\Delta Xr(k)$ [mm]	$\Delta Yr(k)$ [mm]	$\Delta\theta r(k)$ [deg]	$Zr(k)$ [mm]	B_r [mm]	Order of swing leg
22.026	237.002	0.809	300.0	417.078	2→4→3→1
43.958	217.174	-0.818	360.0	372.904	4→2→3→1
45.347	200.484	-0.639	360.0	354.896	4→2→3→1
32.227	189.676	1.413	360.0	362.871	4→2→3→1
12.261	186.691	4.182	360.0	389.931	2→4→3→1
-15.944	241.234	3.894	360.0	520.653	4→2→3→1
-17.443	240.270	4.289	360.0	528.034	4→2→3→1
-18.958	239.279	4.695	300.0	448.019	3→1→4→2
-20.522	238.245	5.122	300.0	446.065	3→1→4→2
-22.187	237.142	5.586	300.0	443.984	3→1→4→2

り遊脚の順番を決定した場合は遊脚の順番を一定とした場合に比べ胴体の移動量が小さくなっていることが確認できる。しかし、24通りすべての順番を試した場合とでは、1回目と5回目の歩行周期で遊脚の順番が異なり、胴体の移動量が若干大きい結果となった。これは、RBFNNの学習に用いたデータが少なかつたためだと考えられる。このため、胴体の移動量をより小さくするためには、RBFNNの学習に用いるデータを増やす必要がある。

6. 自由歩容を考慮した障害物回避行動

これまでの障害物回避行動は遊脚の順番は一定(または固定)としたものであった。ここでは遊脚の順番決定用のRBFNNを用いて、障害物回避行動決定用のRBFNNを再度学習する方法を示し、さらに遊脚の順番変更(つまり、自由歩容)による行動決定への影響についてシミュレーションを通して検討する。

6.1 自由歩容を考慮した障害物回避行動の獲得

障害物が存在した場合、その障害物の高さに合わせて胴体の高さも変化させる。歩行移動の際に得られた障害

物の位置情報、各脚先位置と胴体の高さおよび目的地の情報から行動決定用のRBFNNによりロボットの移動量を決定する。さらに、このロボットの移動量と胴体の高さから遊脚の順番決定用のRBFNNを用いて自由歩容を行う。また、行動決定用のRBFNNはGAを用いて、少ない歩行周期回数でより目的地に近づくよう再度学習を行う。

6.2 シミュレーション結果

再学習した行動決定用のRBFNNを用いてシミュレーションを行った。ロボットに対して左右それぞれに乗り越えられる障害物と壁を設定した場合の結果を以下に示す。

6.2.1 右の障害物を壁とした場合

スタート時のロボットの重心から目的地点までの距離誤差を

$$(x_{de}, y_{de}) = (-0.15, 2.2) [\text{m}]$$

とし、2つの障害物のx座標、y座標、z座標を

$$(x_{or1}, y_{or1}) = (0.2, 0.85), (x_{or2}, y_{or2}) = (0.2, 0.65)$$

$$(x_{or3}, y_{or3}) = (0.5, 0.55), (x_{or4}, y_{or4}) = (0.5, 0.75)$$

$$(x_{ol1}, y_{ol1}) = (-0.5, 0.85), (x_{ol2}, y_{ol2}) = (-0.5, 0.65)$$

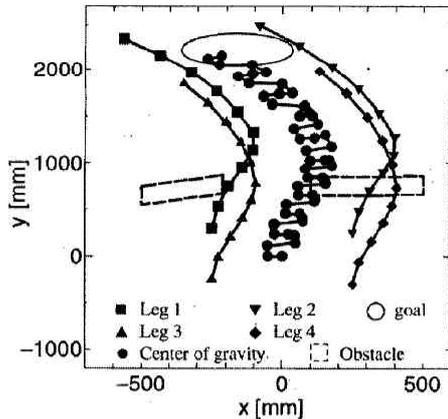


Fig.32 Movement path of the quadruped robot when determining the order of swing leg using RBFNN

Table.5 The amount of movements of the robot's body when determining the order of swing leg using RBFNN

$\Delta X_r(k)$ [mm]	$\Delta Y_r(k)$ [mm]	$\Delta \theta_r(k)$ [deg]	$Z_r(k)$ [mm]	B_r [mm]	Order of swing leg
22.026	237.002	0.809	300.0	418.090	4→2→3→1
43.958	217.174	-0.818	360.0	372.904	4→2→3→1
45.347	200.484	-0.639	360.0	354.896	4→2→3→1
32.227	189.676	1.413	360.0	362.871	4→2→3→1
12.261	186.691	4.182	360.0	390.217	4→2→3→1
-15.944	241.234	3.894	360.0	520.653	4→2→3→1
-17.443	240.270	4.289	360.0	528.034	4→2→3→1
-18.958	239.279	4.695	300.0	448.019	3→1→4→2
-20.522	238.245	5.122	300.0	446.065	3→1→4→2
-22.187	237.142	5.586	300.0	443.984	3→1→4→2

$$(x_{o13}, y_{o13})=(-0.1, 0.66), (x_{o14}, y_{o14})=(-0.1, 0.86)[m]$$

$$z_{o1}=0.3, z_{o1}=0.12[m]$$

としたときの移動経路をFig.33に示す。また、胴体の高さや脚の接地位置をFig.34に示す。

右側の障害物が越えられない高さであるため、ロボットは左方へ移動していることが分かる。左側の障害物に対しては、脚1は乗り越え動作により、脚3は跨ぎ越え動作によりそれぞれ障害物を回避し、ロボットが目的地まで移動していることが確認できる。さらに、このときゴールまでに要した歩行周期回数は10回であり、ゴール時のロボットの重心から目的地までの距離誤差は

$$(x_{de}, y_{de})=(-0.017, -0.027)[m]$$

であった。障害物回避時のロボットの移動量と遊脚の順番をTable6に示す。ここで、遊脚を表す脚の番号は左前脚を1、右前脚を2、左後脚を3、右後脚を4とする。ロボットの移動量に対して遊脚の順番が変化していることが確認できる。

また、再学習を行わない行動決定用のRBFNNを用い、遊脚の順番を固定した場合のシミュレーション結果は4.2.1節で示した。このとき、ゴールまでに要した歩行周期回数

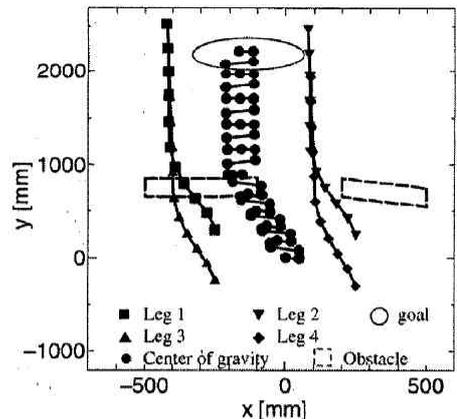


Fig.33 Movement path of quadruped robot when changing the order of swing leg using RBFNN (in the case where a right obstacle is a wall)

Table.6 The amount of movements of the robot in the case where a right obstacle is a wall

$\Delta X_r(k)$ [mm]	$\Delta Y_r(k)$ [mm]	$\Delta \theta_r(k)$ [deg]	$Z_r(k)$ [mm]	Order of swing leg
-26.645	183.469	0.751	300.0	3→1→4→2
-35.059	157.334	0.497	360.0	3→1→4→2
-32.240	162.725	0.311	360.0	3→1→4→2
-24.792	181.329	0.089	360.0	3→1→4→2
-14.077	211.723	-0.166	360.0	3→1→4→2
2.710	274.449	-0.255	360.0	4→2→3→1
2.834	266.891	-0.172	300.0	4→2→3→1
1.787	265.774	-0.159	300.0	4→2→3→1
0.661	264.499	-0.143	300.0	4→2→3→1
-0.516	263.101	-0.126	300.0	4→2→3→1

は9回であり、ゴール時のロボットの重心から目的地までの距離誤差は

$$(x_{de}, y_{de})=(-0.037, 0.046)[m]$$

であった。

6.2.2 左の障害物を壁とした場合

スタート時のロボットの重心から目的地までの距離誤差を

$$(x_{de}, y_{de})=(-0.15, 2.2)[m]$$

とし、障害物のx座標、y座標、z座標を

$$(x_{o11}, y_{o11})=(0.1, 0.85), (x_{o12}, y_{o12})=(0.1, 0.65)$$

$$(x_{o13}, y_{o13})=(0.5, 0.66), (x_{o14}, y_{o14})=(0.5, 0.86)$$

$$(x_{o11}, y_{o11})=(-0.5, 0.75), (x_{o12}, y_{o12})=(-0.5, 0.55)$$

$$(x_{o13}, y_{o13})=(-0.2, 0.65), (x_{o14}, y_{o14})=(-0.2, 0.85)[m]$$

$$z_{o1}=0.12, z_{o1}=0.3[m]$$

としたときの移動経路をFig.35に、胴体の高さをFig.36に示す。ここで、左側の障害物が越えられない高さであるため、ロボットは右方へ移動していることが分かる。また、右側の障害物に対しては、脚2、脚4ともに乗り越え動作により障害物を回避し、ロボットが目的地まで移動していること

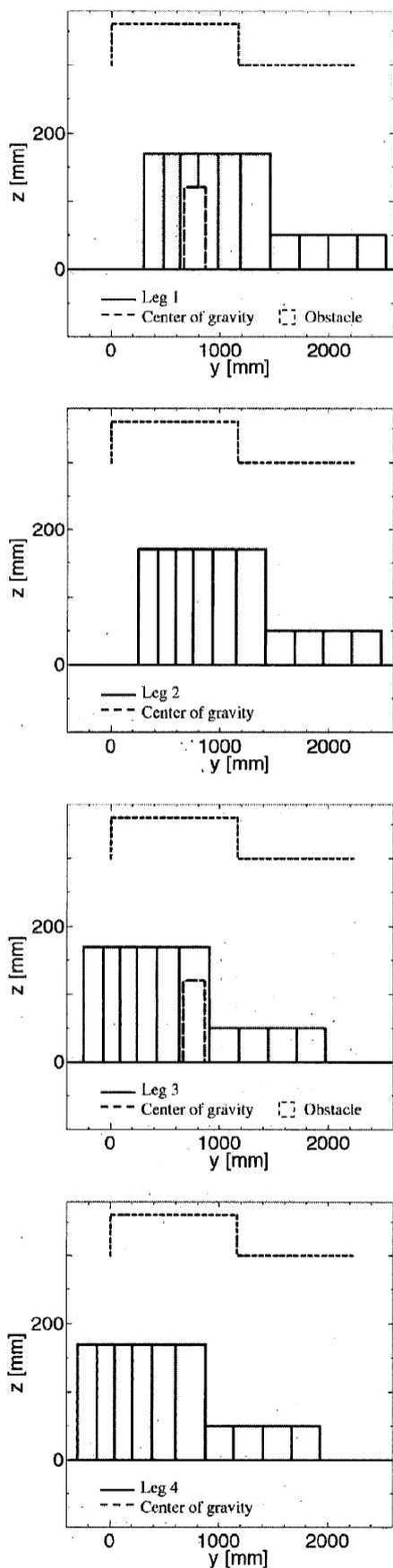


Fig.34 Motion of robot's leg in the case where a right obstacle is wall

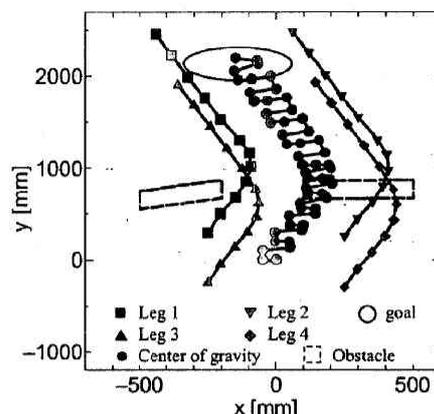


Fig.35 Movement path of quadruped robot when changing the order of swing leg using RBFNN (in the case where a left obstacle is a wall)

が確認できる。ゴールまでに要した歩行周期回数は11回であり、ゴール時のロボットの重心から目的地点までの距離誤差は

$$(x_{des}, y_{de}) = (-0.018, -0.014) [m]$$

であった。障害物回避時の遊脚の順番をTable7に示す。ここでもロボットの移動量に対して遊脚の順番が変化していることが確認できる。

4脚歩行ロボットの遊脚の順番を考慮した障害物回避行動を獲得するために、RBFNNを用いて目的地情報、障害物の位置および自己状態から4脚歩行ロボットの行動を決定した。ただし、自由歩容を実現する遊脚の順番は、4脚ロボットの移動量と胴体の高さから別のRBFNNにより決定した。ここでは、ロボットの行動を決定するためのRBFNNは、遊脚の順番を一定にした状態で構成したものを再学習することにより利用した。この結果、ゴール時の目的地までの距離誤差は小さくなったが、歩行周期回数は多くなる結果となった。これは、GAの評価が距離誤差を優先したためと考えられる。

また、遊脚の順番を固定して、再学習なしの行動決定用のRBFNNを用いた場合のシミュレーション結果は4.2.2節に示した。ゴールまでに要した歩行周期回数は10回であり、ゴール時のロボットの重心から目的地点までの距離誤差は

$$(x_{de}, y_{de}) = (0.068, -0.027) [m]$$

であった。

7. おわりに

本稿では、障害物回避行動を獲得するために、RBFNNを用いて目的地情報、障害物の位置および自己状態から4脚歩行ロボットの行動を決定した。RBFNNの設計パラメータは複数の環境を用意し、最小の歩行周期で目的地へ移動できるようにGAを用いて最適化を行った。その結果、未学習環境においても目的地への経路の獲得ができ、さらに、

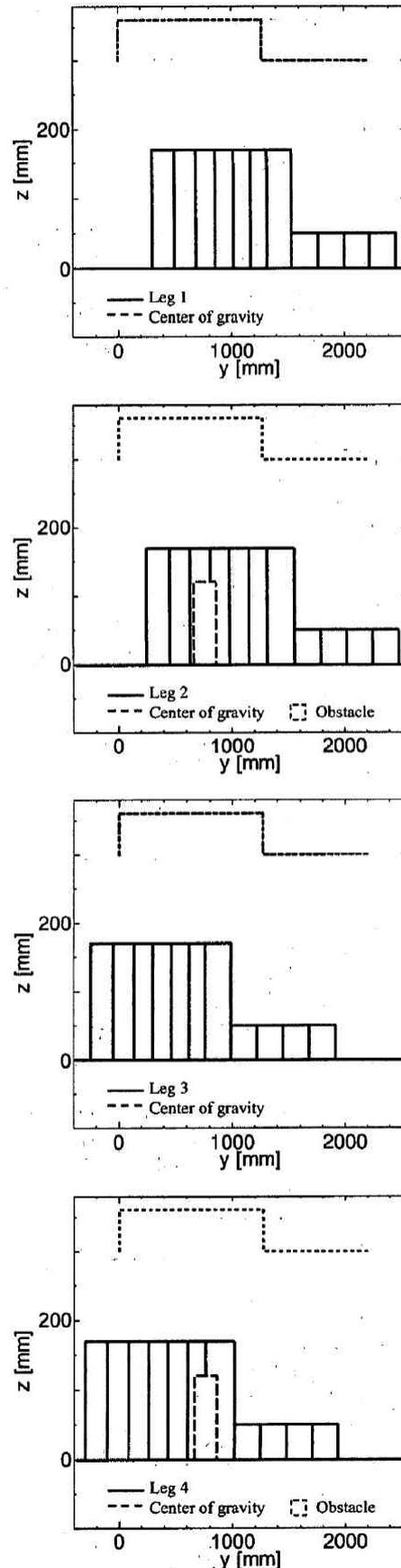
Table.7 The amount of movements of the robot in the case where a left obstacle is a wall

$\Delta Xr(k)$ [mm]	$\Delta Yr(k)$ [mm]	$\Delta \theta r(k)$ [deg]	$Zr(k)$ [mm]	Order of swing leg
45.379	198.899	-0.298	300.0	4→2→3→1
55.253	181.795	0.837	360.0	4→2→3→1
49.297	172.875	1.419	360.0	4→2→3→1
36.077	165.824	1.896	360.0	2→4→3→1
18.925	159.794	2.220	360.0	2→4→3→1
-0.386	155.208	2.319	360.0	2→4→1→3
-23.440	241.344	0.167	360.0	4→2→3→1
-23.620	239.286	0.087	300.0	4→2→3→1
-24.022	237.751	0.013	300.0	4→2→3→1
-24.397	236.259	-0.060	300.0	4→2→3→1
-24.746	234.816	-0.131	300.0	4→2→3→1

実機を用いてその有効性を示した。さらに、ロボットの移動量に応じて遊脚の順番を変更できる自由歩容獲得のためのシステムを構築した。この自由歩容での遊脚の順番は、4脚ロボットの移動量と胴体の高さから遊脚の順番決定用(または、自由歩容獲得用)のRBFNNにより決定した。遊脚の順番決定用のRBFNNの設計パラメータの学習には、それぞれの遊脚の順番に対してロボットの移動量を変化させたデータを20個用意し、このデータの入力と出力の関係を満足するようGAを用いて最適化を行った。その結果、遊脚の順番を一定とした場合と比較して胴体の移動量は小さくなったが、理想的な24通りすべての遊脚の順番を試した場合とでは若干大きくなる結果となった。これは、RBFNNの学習に用いたデータが少なかつたためだと考えられ、胴体の移動量をより小さくするためには、学習用のデータを増やす必要があると思われる。

また、行動決定用のRBFNNの再学習に関して、自由歩容獲得用のRBFNNを利用して再学習する方法を示した。行動決定用のRBFNNを再学習しない場合と比較して、自由歩容を適用し再学習したRBFNNは目的地までの距離誤差はさらに小さくなることを実証した。しかし、歩行周期回数の中では、再学習しない場合に比べて提案手法では若干多くなるという結果になった。これは、最適化に用いたGAの評価が距離誤差を小さくする点を優先したためと考えられる。したがって、歩行周期回数をより小さくするためにはGAの適応度を再検討する必要もあると考えられる。

一方、本研究では障害物の認識は超音波センサのみで行っており、障害物の形状が四角形等で超音波をうまく反射するものしか検出することができない。このため、障害物は単純な形状のみとし、複雑な形状をした障害物は考慮しなかった。複雑な形状を認識するためには、カメラ等の視覚センサと超音波センサを併用した知覚・認識システムを構築し、障害物認識の更なる向上を図る必要があると考え

**Fig.36** Motion of robot's leg in the case a right obstacle is wall

られる。

参考文献

- 1) K. K. S. Afak and G. G. Adams, "Dynamic Modeling and Hydrodynamic Performance of Biomimetic Underwater Robot Locomotion", *Autonomous Robots*, **13-3** (2002), p.223.
- 2) 広瀬茂男, 米田完, "実用的4足歩行機械の開発に向けて", *日本ロボット学会誌*, **11-3** (1993), p.360.
- 3) 安達弘典, 小谷内範穂, 中村達也, 中野栄二, "脚に設置された力センサを用いた4脚歩行ロボットの適応歩容", *日本ロボット学会誌*, **9-6** (1991), p.707.
- 4) 前田陽一郎, 津谷定廣, 萩原史明, "多機能移動ロボットの障害物またぎ越え動作解析", *日本ロボット学会誌*, **7-2** (1989), p.121.
- 5) 木村浩, "四足歩行ロボットの動歩行について", *日本ロボット学会誌*, **11-3** (1993), p.372.
- 6) X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, "An ART-Based Fuzzy Controller for the Adaptive Navigation of a Human-Coexistent Quadruped Robot", *IEEE/ASME Transactions on Mechatronics*, **7-3** (2002), p.318.
- 7) X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, "Path Tracking Based on Closed-Loop Control for a Quadruped Robot in a Cluttered Environment", *ASME, Journal of Dynamic Systems, Measurement, and Control*, **124** (2002), p.272.
- 8) 古荘純次, "歩行ロボットの研究展開", *日本ロボット学会誌*, **11-3** (1993), p.306.
- 9) 中村達也, 関光徳, 森善一, 安達弘典, "脚型ロボットの不整地歩行のためのモンテカルロ法を用いた自由歩容生成", *日本機械学会ロボティクス・メカトロニクス講演会 99 講演論文集*, (1999), 1A1-42-061.
- 10) 広瀬茂男, 有川敬輔, "研究用プラットフォームとしての普及型歩行ロボット TITAN-VIII の開発", *日本ロボット学会誌*, **17-8** (1999), p.1191.
- 11) S. Hirose, "A Study of Design and Control of a Quadruped Walking Vehicle", *Journal of Robotics Research*, **3-2** (1984), p.113.
- 12) 広瀬茂男, 塚越秀行, 米田完, "不整地における歩行機械の静的安定性評価基準", *日本ロボット学会誌*, **16-8** (1998), p.1076.
- 13) R. B. McGhee and A. A. Frank, "On the stability properties of quadruped creeping gaits", *Mathematical Biosciences*, **3** (1968), p.331.
- 14) 米田完, 広瀬茂男, "歩行機械の転倒安定性", *日本ロボット学会誌*, **14-4** (1996), p.517.
- 15) E. G. Papadopoulos and D. A. Rey, "A New Measure of Tipover Stability Margin for Mobile Manipulators", *IEEE International Conference on Robotics and Automation*, (1996), p.3111.
- 16) D. A. Messuri and C. A. Klein, "Automatic Body Regulation for Maintaining Stability of a Legged Vehicle During Rough-Terrain Locomotion", *IEEE Journal of Robotics and Automation*, **RA-1-3** (1985), p.132.
- 17) A. Ghasempoor and N. Sepchri, "A Measure of Machine Stability for Moving Base Manipulators", *IEEE International Conference on Robotics and Automation*, (1995), p.2249.
- 18) 坂和正敏, 田中雅博, "ニューロコンピューティング入門", (1997), 森北出版.
- 19) V. T. S. Elanayar and Y. C. Shin, "Radial Basis Function Neural Network for Approximation and Estimation of Nonlinear Stochastic", *IEEE Transactions on Neural Networks*, **5-4** (1994), p.594.
- 20) 北野宏明, "遺伝的アルゴリズム", (1993), 産業図書.
- 21) Z. Michalewicz, "Genetic Algorithms + Data Structure = Evolution Programs", 3rd, revised and extended edition ed., Springer, Germany, (1996).