故障に対して冗長性を備えた仮想ロボットのニューロ進化による持続可能な行動獲得

速水陽平　辰巳嵩豊　上野史　高玉圭樹　（電気通信大学）

Sustainable manipulation by neuro-evolution for a simulator robot having robustness against sensor troubles

Yohei Hayamizu　Takato Tatsumi　Fumito Uwano　Keiki Takadama
(University of Electro-comunications)

Abstract− This paper proposes the Neuro-evolutionary method that can acquire the manipulate rules of a robot, which work well even when sensors become broken. In order to investigate the effectiveness of the proposed method, we conducted the intensive simulation with the quadruped robot in a physics simulator and revealed that the proposed method can successfully acquire the manipulate rules of a robot which derives mostly the same performance of the normal situation with sensor troubles.

Key Words: センサ故障, ニューロ進化, 冗長性, 持続可能性

1　はじめに

昨今，災害現場や宇宙環境などの人が立ち入ることが困難な環境での作業を目的としたロボットの研究が行われている [1, 4]．特に自律制御ロボットの研究ではセンサ情報に基づいて災害地等の状況を観測し，それに対して適切な制御を行うための制御則を獲得することを目的としている．このとき，地形変化などのロボットの周囲の環境変化や，センサやアクチュエータの故障による情報の劣化など様々な不測の事態が起こりうるため，ロボットの周囲の環境変化やセンサの故障などを想定した冗長性のある制御は重要である．従来，進化計算，特にニューロ進化を用いたロボット制御がある [2, 3, 7]．ニューロ進化によるロボット制御は，ロボットの設計に関するバイアスを排除した上で環境との相互作用による進化を通して制御則を獲得するため，センサの故障であってもロボットの設計を考慮することなく制御則の創発が期待できる．しかしながら，従来研究においては環境変化やセンサの故障が起きてからその変化に合わせた制御則を獲得するため，災害現場などの即応性が求められる場合において適用することが困難である．本研究では災害現場のような即応性が求められる場合へのロボットの適応を目指し，予めセンサの故障を想定した制御則を，正常時における制御則と合わせて獲得する方法を提案する．具体的にはニューロ進化によりロボット制御に関わる内部構造を多様に進化させることで，センサの故障にロバストな制御則を獲得する．提案手法の有効性を検証するために，本研究では物理シミュレーション上の 4 足歩行ロボットにおける制御則獲得に関する実験を行う．4 足歩行ロボットと目標地点までの距離を評価指標として，センサからの入力が得られている場合，センサからの入力が得られなくなった場合においてそれぞれ目標地点までの制御が可能であるかについて実験を行う．

本論文の構成は以下の通りである．第 2 章では従来手法であるニューロ進化についての説明を行い，第 3 章で提案手法のアルゴリズムについて述べる．実験及びその結果を第 4 章で触れ，第 5 章で考察を行う．第 6 章で本論文のまとめを記す．

2　ニューロ進化

ニューロ進化とはニューラルネットワーク (Neural Network: NN) の重みの更新に進化計算を用いた手法である [5]．進化計算には主に遺伝的アルゴリズムが用いられ，遺伝的アルゴリズムに従って NN の結合重みを学習していく．入力次元数が膨大となる問題において NN の結合重みを効率的に決定していく．そのためニューロ進化は，自律制御ロボットの制御系を NN で表現することで制御の最適化を行う進化ロボティクスに応用されている．

2.1　進化ロボティクス

進化ロボティクスはロボットの制御系を進化させることで効率的な制御則を獲得するものである [7]．従来における進化ロボティクスの研究では制御系として NN を用い，その進化に焦点を当ててきた [8]．また，ロボットの身体性についても焦点を当て，制御系だけではなく形態も同時に進化させるという研究がなされている [10]．生物の進化に着想を得た進化ロボティクスによるアプローチにより，目的の制御則を効率的に達成することができる．例えばロボットのセンサやモータなどに関わる形態素をモジュール化することで環境にすばやく適応する進化や，モータが故障したことを検知して新たな制御則を獲得するような進化をする研究がなされている [2, 3]．進化ロボティクスの進化論的アプローチにおいて Bongard らが用いているアルゴリズムを以下に簡単に記す (Algorithm1)．

2.2　ロボットの制御獲得方法

進化ロボティクスにおける自律制御ロボットの制御則を獲得する方法について説明する．進化ロボティクスにおいて，ロボットの制御系は NN の結合重みにより表現され，その結合重みを遺伝子と呼ぶ．入力次元 I，出力次元 O としたとき，ロボットの遺伝子は $I \times O$ の行列で表現される．$I = 5, O = 8$ のときの NN の結合重みを表現する遺伝子を Fig.1 に示す．自律制御ロボットの制御系を進化させるために，まず遺伝子を持つロボットを 1 個体として，P 個の個体を生成する．P 個の個体からなる集団を母集団と呼ぶ (lines 7-9 in

$N_M = 8$

$N_s = 5$

	0	1	2	3	4	5	6	7
0	0.1	0.74	0.84	0.58	0.1	0.74	0.84	0.58
1	0.03	-1	0.11	0.91	0.74	0.63	0.75	0.75
2	0.58	1	0.75	0.74	0.84	0.74	0.63	0.63
3	0.91	0.74	0.63	-1	0.11	0.03	-1	0.18
4	-0.9	-0.8	0.11	0.77	1	0.58	1	0.75

Fig. 1: NN の結合重みを表現する遺伝子

Algorithm1)．生成された母集団の個体それぞれを一定時間動かし，発現したロボットの動作を評価関数を用いて評価する (lines 11 in Algorithm1)．評価した後，突然変異による新たな個体の生成を行う．新たな個体の生成は，まず母集団の中で最も評価値の高い個体を選択して，複製を生成する．残りの $P-1$ 個体については，P 個の個体の中からランダムに 2 個体を選択し，2 つの個体のうち評価値の高い個体を複製したものに対して突然変異を行ったものを新たな個体として生成する．突然変異では $I \times O$ の行列のうちランダムに要素を 1 つ選択し，$\mathcal{N}(\mu = w_{ijk}, \sigma = |w_{ijk}|)$ のガウス分布に従う乱数で要素を上書きする (lines 12 in Algorithm1)．以上の過程を世代交代として終了条件を満たすまで繰り返し，最終世代の個体のうち最良個体の遺伝子を最終的な制御系として用いる．

3 持続可能な制御の獲得

進化ロボティクスによる最良個体の獲得は確率的であり，センサが故障して入力が変化した際に影響を受けない個体を生成するには探索領域が膨大になる問題と世代ごとに最良個体のみを獲得しようとするため，センサが故障しても対応可能な個体が残らないという問題があった．そこで任意のセンサが故障した場合にも制御可能な個体を最終世代において保持しておく持続可能な制御獲得手法を提案する．具体的には，最終世代において，通常時に評価値の高い個体，センサ故障時に評価値の高い個体のどちらの個体も残っているように個体を進化させるため，割引率を用いた突然変異である γ-Mutation と遺伝子類似度に基づいた母集団更新手法を提案する．

3.1 γ-Mutation

母集団内の個体をセンサが故障した際にも対応できるような個体を生成する方法として，従来のガウス分布に従う乱数のみによる突然変異に加えて，割引率 γ を導入した突然変異手法である γ-Mutation を提案する．従来手法における突然変異ではトーナメント選択により選択された個体に対して，遺伝子の要素一つをガウス分布に従う乱数による上書きを行っていた．一方で提案する割引率を用いた突然変異では i 番目のセンサ S_i からの入力を受け取る遺伝子全てに対して割引率 γ を掛ける．γ-Mutation のアルゴリズムと通常の突然変異を用いた個体生成のアルゴリズム γ-Mutate Selection を Algorithm2 に示す．

まず，通常の突然変異と同様に P 個から成る母集団について，最良個体の複製体を生成する．次に P 個の個体の中からランダムに 2 個体を選択し，2 つの個体のうち評価値の高い個体に対して，x 番目のセンサ S_x が故障したという前提の下，割引率 γ を遺伝子の x 行目の要素に掛ける．ここで Γ は割引率 γ の集合である．ここまでで，$|\Gamma|+1$ 個の個体が生成される．残りの $P-|\Gamma|-1$ については P 個の個体の中からランダムに 2 個体を選択し，2 つの個体のうち評価値の高い個体に対して，x 番目のセンサ S_x が故障したという前提の下，割引率 γ を遺伝子の x 行目の要素に掛けた後，さらに遺伝子の x 行を除いた要素集合の中から一つをランダムに選択し，$\mathcal{N}(\mu = w_{ijk}, \sigma = |w_{ijk}|)$ のガウス分布に従う乱数で要素を上書きする．Fig.**??**に γ-Mutation により変化する遺伝子の様子を示す．

Algorithm 1 Basic control

```
1:  procedure BASIC CONTROL
2:      P ← Number of Individuals
3:      G ← Number of Generations
4:      S ← Number of Input
5:      M ← Number of Output
6:      for i to P do
7:          W_i ← Genome(S × M)
8:          Population_i ← W_i
9:      for g to G do
10:         CONTROL()
11:         SELECTION()
12:
13: procedure CONTROL()
14:     for each Individual in the Population do
15:         Fitness_i = 0
16:         Fitness_i ← Fitness_i + SIM(W_i)
17:
18: procedure SELECTION()
19:     Population_0 ← The best Individual
20:     for i = 1 to P do
21:         MUTATION(W_i)
22:         Population_i ← W_i
23:
24: procedure MUTATION(W_i)
25:     j = randint(S)
26:     k = randint(M)
27:     w_jk ← N(μ = w_jk, σ = |w_jk|)
28:
29: procedure SIM(W_i)
30:     p_i = 0(p_i ∼ W_i)                ▷ Performance
31:     for t to T do                     ▷ Time step
32:         p_i = p_i + p_i^t
33:     Return p_i
```

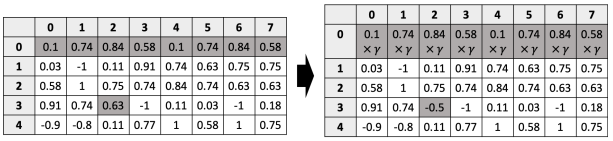


Fig. 2: γ-Mutation の様子

3.2 遺伝子類似度に基づいた母集団更新手法

ロボットのセンサの故障を想定した制御則を事前に獲得するために我々は遺伝子類似度に基づいた母集団更新手法を提案する．センサが故障したときでも制御可能な個体を生成するために，各センサが故障しても問題なく

Algorithm 2 γ Mutation

$\Gamma = \{\gamma_0, \gamma_1, \cdots, \gamma_n\}$
$x \leftarrow$ Index of sensor premised on trouble
procedure γ-MUTATE SELECTION(Γ)
    $Population_0 \leftarrow$ The best Individual
    for $i = 1$ to $|\Gamma|$ **do**
        γ-*MUTATION*(W_i, γ_i)
        $Population_i \leftarrow W_i$
    for $i = |\Gamma|$ to P **do**
        γ-*MUTATION*(W_i, γ_i)
        MUTATION(W_i)
        $Population_i \leftarrow W_i$

procedure MUTATION(W_i))
    $j =$ randint(S\x)
    $k =$ randint(M)
    $w_{jk} \leftarrow \mathcal{N}(\mu = w_{jk}, \sigma = |w_{jk}|)$

procedure γ-MUTATION(W_i, γ))
    for k to M **do**
        $w_{xk} \leftarrow w_{xk} \times \gamma$

制御できるような遺伝子を持つ個体を生成し，遺伝子の類似する個体をグループ化したうえで突然変異を行い，進化させる．センサからの入力次元数 (センサニューロンの数) を N_s，モータへの出力次元数 (モータニューロンの数) を N_m としたとき，個体は $N_s + 1 = |\{G_0, G_1, \cdots, G_{N_s}, G_{N_{s+1}}\}|$ 個のグループに分類される．分類は Fig.1 のように表された遺伝子の類似度に基づき行われる．遺伝子類似度に基づいた母集団更新手法アルゴリズムを Algorithm3 に示す．まず母集団内の個体を評価値に基づいて降順に並べる．個体の遺伝子の各列においてそれぞれ要素が最小となる行のインデックス x を求め，x の個数をそれぞれカウントする．この値を Principal Count(PC) と呼ぶ．遺伝子の中で PC が最も大きくなるインデックス x を求め，その遺伝子を持つ個体を x に対応するグループに仕分けする．PC の求め方の例として，Fig.3 にインデックス 3 が最も PC が大きくなるときの遺伝子を示す.Fig.3 の網掛けの部分が各列において要素が最小となる値であり, 各行それぞれについて網掛けの個数をカウントしたものが PC である. ここで，PC が最も大きくなるインデックス x が複数ある場合は複数の x からランダムでインデックスを選択し，仕分けする．また，任意のグループ G_x に分類される個体数がそのグループの個体許容量 $|G|_{Max}$ に達した場合，それ以降 G_x に分類される個体はすべて $G_{N_{s+1}}$ に仕分けする．これを母集団内すべての個体 P に対して行う．個体の各グループへの分類方法のイメージを Fig.4 に示す.

4　4 足歩行ロボット

本研究で扱う問題として,4 足歩行ロボットを取り上げる. 実験環境として OpenDynamicsEngine(ODE)[9] 上に構築されたオープンソースの Python ラッパーである Pyrosim[6] を使用し，Fig.5 に示すような仮想上の 4 足歩行ロボットを作成し，提案手法の有効性を検証する．Fig.5 に示されたロボットは円柱状の要素を 2 つ連

Algorithm 3 Grouping

procedure GROUPING()
    MERGE()
    SORT(Population)
    for each Individual in the Population **do**
        $\boldsymbol{X} \leftarrow$ *GET MAX PC()*
        $x =$ randint($|\boldsymbol{X}|$)
        if size(G_x) $> |G|_{Max}$ **then**
            $G_{N_{s+1}} \leftarrow W_i$
        else
            $G_x \leftarrow W_i$

procedure MERGE()
    for gr to $N_s + 1$ **do**
        for i to size(G_{gr}) **do**
            $Population_i \leftarrow W_i$

procedure GET MAX PC()
    for k to M **do**
        if **then**Max($\boldsymbol{w}_k$)
            $\boldsymbol{X} \leftarrow k$
    Return $\boldsymbol{X}$

	0	1	2	3	4	5	6	7	PC
0	0.1	0.74	0.84	0.58	0.1	0.74	0.84	0.58	2
1	0.03	-1	0.11	0.91	0.74	0.63	0.75	0.75	2
2	0.58	1	0.75	0.74	0.84	0.74	0.63	0.63	1
3	0.91	0.74	0.63	-1	0.11	0.03	-1	0.18	3
4	-0.9	-0.8	0.11	0.77	1	0.58	1	0.75	1

Fig. 3: インデックス 3 が PC 最大となるような遺伝子

結させた脚を 4 本と長方形の要素の体で構成されている．円柱状の要素同士は 1 自由度の関節により接続されており，4 本の脚は長方形の要素とそれぞれ 1 自由度の関節で接続されている．Fig.6 に本研究で作成した 4 足歩行ロボットのアーキテクチャを示す. 関節はモータにより制御される．計 8 つの関節の角度の最大値及び最小値は ± 1[rad] であり，その範囲内でモータが駆動する．各脚には触覚センサが内蔵されており，脚が地面と接地した場合は $+1$，地面から離れた場合は -1 の値を返す．また，4 つの触覚センサとは別に光センサが体に内蔵されており，光源との距離に応じた実数値を返す．光センサと光源の距離を d としたとき光センサの返す値 l は $l = 1/d^2$ で表される．4 足歩行ロボットの計 5 つのセンサと計 8 つの関節のモータはそれぞれ外部からの入力により反応するニューロンを有しており，それぞれセンサニューロンと，モータニューロンと呼ばれる．センサニューロンとモータニューロンはそれぞれがシナプスにより結合され，NN を形成している．このときの NN を表現した遺伝子を Fig.7 に示す. また，Fig.7 中のセンサニューロン S 及びモータニューロン M は Fig.6 中の S, M に対応している．NN の動きとしては，センサニューロンが外部から入力を受け取り，モータニューロンへ出力する．モータニューロンがセンサニューロンからの入力値に反応して活性化し，関節のモータに指令を与えることで制御を可能とする．時刻 t における i 番目のモータ $m_i^{(t)}$ に送られ

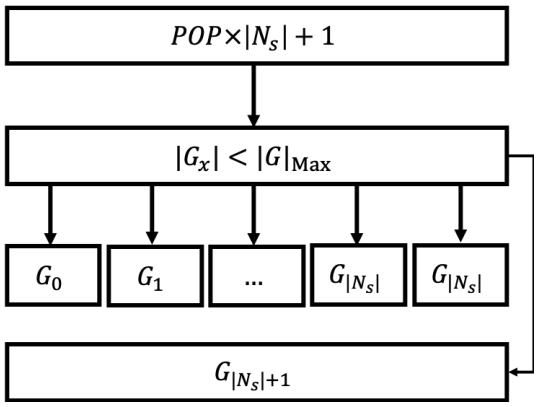


Fig. 4: 個体のグループへの分類方法

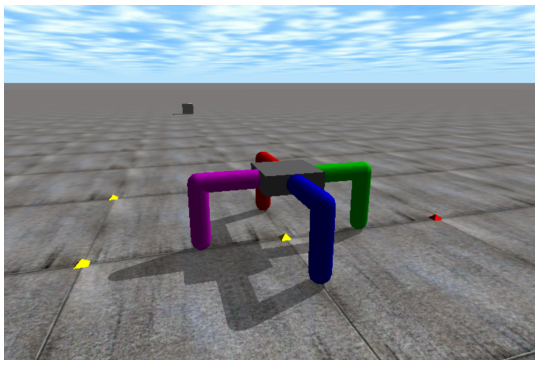
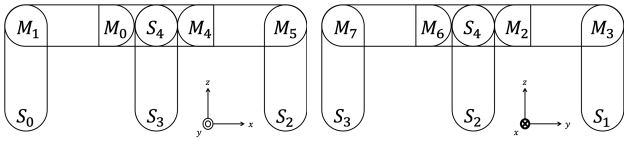


Fig. 5: 仮想上の 4 足歩行ロボット

る指令値は以下の式で表される．ここで 1 試行における終了時刻 (ステップ) を T とする．

$$m_i^{(t)} = tanh\left[m_i^{(t-1)} + \tau_i \sum_{j=1}^{J=5} w_{ji} s_j^{(t)}\right] \qquad (1)$$

ただし τ_i はモータの動きやすさを表す係数，$s_j^{(t)}$ は時刻 t における j 番目のセンサの値，w_{ji} は j 番目のセンサニューロンから i 番目のモータニューロンに伸びるシナプスの結合重みである．結合重みは $[-1, 1]$ の値域を取る．



[1] xz 軸平面　　[2] yz 軸平面

Fig. 6: 4 足歩行ロボットのアーキテクチャ

5 実験

割引率を用いた突然変異 γ-Mutation と遺伝子類似度に基づいた母集団更新手法の有効性を示すため，物理シミュレータ上の仮想ロボットを用いてそれぞれ実験を行う．

γ-Mutation についての実験 (実験 1) では 0 番目のセンサの入力が故障したと仮定して実験を行う．初期生成された個体を進化させ最終世代における最良個体に対して，センサ S_0 への入力をすべて 0 にした際に自律制御が可能かを検証する．γ-Mutation に対する比較手法として，通常の突然変異のみを行って進化する手法，

$N_M = 8$

$N_s = 5$

	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7
S_0	0.1	0.74	0.84	0.58	0.1	0.74	0.84	0.58
S_1	0.03	-1	0.11	0.91	0.74	0.63	0.75	0.75
S_2	0.58	1	0.75	0.74	0.84	0.74	0.63	0.63
S_3	0.91	0.74	0.63	-1	0.11	0.03	-1	0.18
S_4	-0.9	-0.8	0.11	0.77	1	0.58	1	0.75

Fig. 7: 4 足歩行ロボットの遺伝子

すなわち $\gamma = 1$ である手法 (1-mutation) と，予めセンサ S_0 の入力を受け取る遺伝子の値を 0 にして進化する方法，すなわち $\gamma = 0$ である手法 (0-mutation) を用いる．

遺伝子の類似度に基づいた母集団更新手法についての実験 (実験 2) では，別のセンサが故障したときにも同様に自律制御が可能かを検証する．この実験では予め i 番目のセンサ S_i が故障することを仮定せず，どのセンサが故障した場合でも対応できるかどうかを示す．

5.1 実験 1

5.1.1 実験設定

それぞれの検証において 1 試行におけるステップ数は $T = 1000$，母集団サイズ M は 96 とする．また，世代数 500 となるときを終了条件とする．割引率については $\Gamma = \{0.9, 0.8, 0.7, 0.6, 0.5\}$ とした．

5.1.2 評価指標

4 足歩行ロボットと目標となる光源との距離を光センサの値より導出した，以下の式により $fitness$ を評価指標とする．

$$fitness = 1/T * \sum_{t=1}^{T} 1/d_t^2 \qquad (2)$$

経験的に $fitness$ が 0.5 以上になると 4 足歩行ロボットと光源との距離が近くなることから，$fitness > 0.5$ のとき，4 足歩行ロボットは光源に到達したとする．

5.1.3 実験結果

実験結果を以下の Fig.8 と Table1 に示す．ただし Table1 は獲得した最終世代の個体の最良個体が光源に到達した段階で制御を停止したときの $Fitness$ であり，実際に 500 世代の試行回数を行うと，Fig.8 に示されたような，より高い $fitness$ を得られている．Fig.8 から，いずれの手法においても $fitness$ が 0. 5 を超えることが確認でき，センサが故障していない場合，すなわちセンサへの入力を変化させない場合において目的を達成するような個体が獲得できていることが分かる．一方で Table1 を見ると，通常の突然変異のみを用いたもので，500 回世代交代を繰り返したものは，センサ S_1 への入力を変化させないとき最終世代における最大 fitness が最も高いが，センサ S_1 への入力を 0 にさせると $fitness$ が最も低くなっており，故障した際に目的を達成するような個体を獲得できていないことが分かる．

5.2 実験 2

複数のセンサが故障したときにも対応できるように，割引率を用いた突然変異に加えて遺伝子類似度に基づ

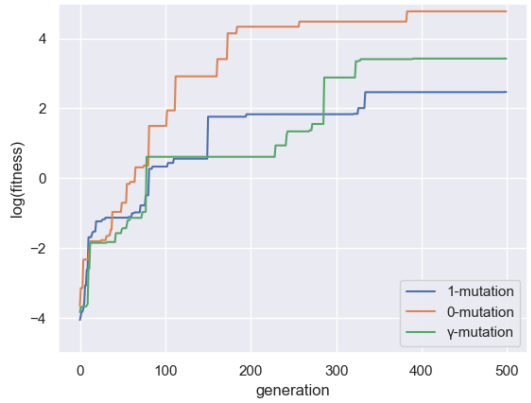


Fig. 8: 通常の制御時の各世代ごとの最大 log(fitness)

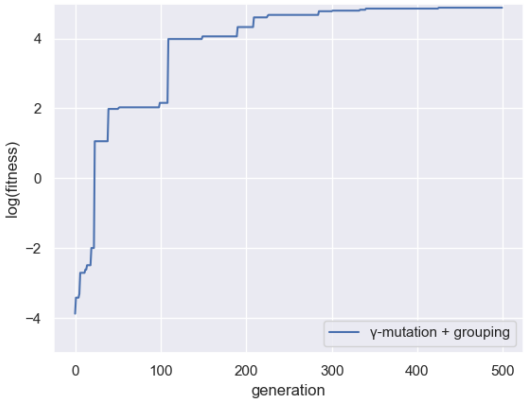


Fig. 9: 通常の制御時の各世代ごとの最大 log(fitness)

Table 1: 最終世代における通常時と S1 への入力を 0 にした時の最大 log(fitness)

	故障前	故障後
1-mutation	1. 084	0. 146
0-mutation	1. 088	1. 088
γ-mutation	1. 034	1. 034

いた母集団更新手法を用いたときに通常の動作が可能かどうかを検証する．

5.2.1 実験設定

この実験においては各グループに 96 個の個体が均等に振り分けられるようにするため，576 の個体を母集団として生成する．個体許容量は 96 とし，終了条件を 500 世代が終了したときとする．また割引率については実験 1 と同様に $\Gamma = \{0.9, 0.8, 0.7, 0.6, 0.5\}$ とした．

5.2.2 評価指標

評価指標には 5.1.2 節と同様のものを用いる．

5.2.3 実験結果

実験結果を以下の Fig.9 と Table2 に示す．ただし Table2 は Table1 と同様に獲得した最終世代の個体の最良個体が光源に到達した段階で制御を停止したときの $fitness$ であり，実際に 500 世代の試行回数を行うと，Fig.9 に示されたような，より高い $fitness$ を得られている．Fig.9 から，各センサがいずれも故障していない状態で $fitness$ が 0. 5 以上となっていることから，遺伝子類似度に基づいて母集団をグループ分けし，そのグループ内で割引率を用いた突然変異により進化を行った場合に目的の制御即を獲得する進化が行われることが分かる．さらに Table2 からいずれかのセンサ S への入力を 0 にしても $fitness$ の値が 0. 5 を超えることから，センサが故障する前から各センサの故障を事前に想定した制御則を獲得することができたと分かる．

6 考察

6.1 γ 値の使用頻度

今回実験 2 で獲得した最良個体がどのような γ 値により更新されたかを示す表を Fig.10 に示す．今回得られた個体は $\gamma = 0.9$ を採用することで効率よく進化していったと考えられる．一方で，進化の過程で，世代数

Table 2: 最終世代における各センサへの入力を 0 にしたときの最大 log(fitness)

	故障前	故障後
S0	1. 315	1. 042
S1	1. 315	1. 038
S2	1. 315	1. 070
S3	1. 315	1. 070
S4	1. 315	1. 059

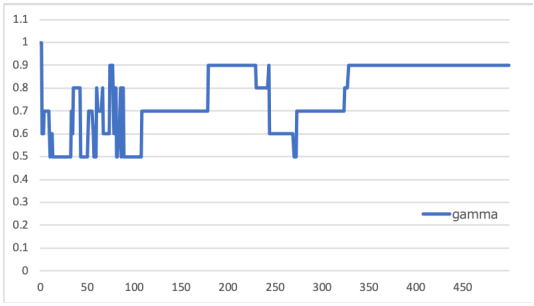
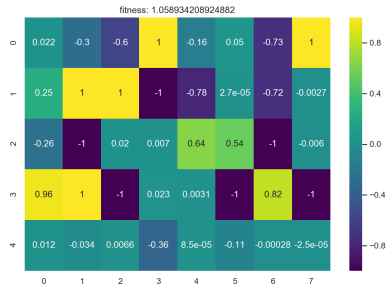


Fig. 10: 最良個体が使用した γ 値

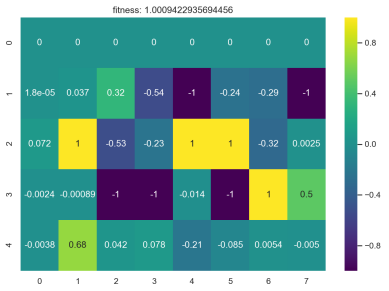
の若いものは様々な γ 値を取りながら進化していくと考えられ，世代を経るごとに γ 値が一定の値に定まるような変化をしていくことが Fig.10 から見て取れる．

6.2 最終世代に獲得された遺伝子

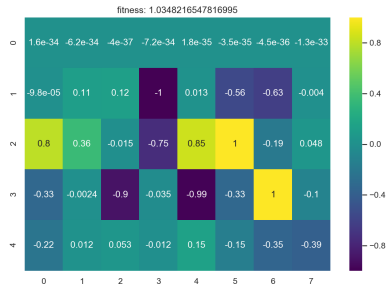
Fig.11 に最終世代における最良個体の遺伝子を示す．[1] ではそれぞれのセンサに対応した遺伝子の行で，すべての行が 0 に限りなく近い値は存在しないため，センサからの入力が変わった際に制御則が変化したことにより，結果として $fitness$ が小さくなったと考えられる．一方で Fig.11 下段の γ-Mutation を用いた手法 [3], [4] は，それぞれ，センサ S_0 に対応した遺伝子が 0 に限りなく近い値になっているものと，センサ S_1 に対応した遺伝子が 0 に限りなく近い値になっているものが獲得されていることが分かる．γ-Mutation により，センサが故障したという前提をおいた場合に対応する遺伝子が 0 ではないものの，限りなく 0 に近い値をとるように多様な個体を選択することが出来ている．これにより，[1] とは対称的にセンサからの入力が変化しても制御則が大きく変化することがなく，$fitness$ に変化がなかったと考えられる．



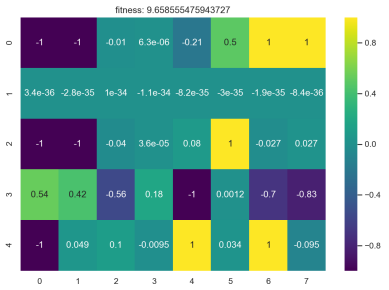
[1] 1-mutation



[2] 0-mutation



[3] γ-mutation($PC = 0$)



[4] γ-mutation($PC = 1$)

Fig. 11: 最終世代における最良個体の遺伝子

7 まとめ

本研究ではニューロ進化を用いたロボット制御を行うため進化ロボティクスという観点から，物理シミュレータ上の 4 足歩行ロボットを用いた実験により，故障によりセンサの入力が変化した場合でも持続可能な制御が可能となるロボットの多様な制御則が獲得できることを示した．具体的には，センサが故障することを想定してニューロンの重みを減却しながら突然変異をさせる γ-Mutation と，制御則の多様性を維持しつつ，センサ別の故障に対応した進化を可能にする遺伝子の類似度に基づいた母集団更新手法により個体の多様性を維持し，センサが故障した際にも目的の制御を達成可能な個体を事前に獲得することが出来た．しかしながら獲得された個体は通常時でもセンサの故障を想定した制御則を活用するという問題があり，センサが故障していない場合においては局所的に良い個体である可能性が考えられる．また，割引率が遺伝子の進化にどのような影響を及ぼしているかという点についてや，センサが故障した際に遺伝子の値がどのような値になっていることが望ましいかについて議論の余地がある．今後の展望として，センサが故障していない通常時において最適な制御則を持つ個体の獲得のために発展的な遺伝子類似度に基づいた母集団更新手法を提案すると共に，Fig.10，Fig.11 のような分析を通して，より効率的な割引率を選択する方法や，割引率の遺伝子に対する影響や遺伝子の値の制御への影響についての考察を行うことでより効果的な進化の方法について研究を行う．

参考文献

1) 淺間 一: “災害時に活用可能なロボット技術の研究開発と運用システムの構築”, ロボット学会誌, VOL.32, NO.1, pp.37-41 (2014)

2) J. Bongard, Anton Bernatskiy, Ken Livingston, Nicholas Livingston, John Long, Marc Smith: “Evolving Robot Morphology Facilitates the Evolution of Neural Modularity and Evolvability”, *In Proceedings of The 2015 Annual Conference on Genetic and Evolutionary Computation*, pp.129-136 (2015).

3) J. Bongard, Victor Zykov, Hod Lipson, “Resilient Machines Through Continuous Self-Modeling”, *SCIENCE*, VOL.314, pp.1118-1121 (2006).

4) Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. and John J. Leonard: “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age”, *IEEE TRANSACTIONS ON ROBOTICS*, VOL.32, NO.6 (2016).

5) J. Gauci, K. O. Stanley, “A Case Study on the Critical Role of Geometric Regularity in Machine Learning”, *Twenty-Third AAAI Conference on Artificial Intelligence*, pp.628-633 (2008).

6) S. Kriegman, C. Cappelle, F. Corucci, A. Bernatskiy, N. Cheney, and J. Bongard, “Simulating the evolution of soft and rigid-bodyrobots”, *In Proceedings of the 2017 Annual Conference on Genetic and Evolutionary Computation*, pp.1117-1120 (2017).

7) R. Pfeifer, J. Bongard, 細田 耕, 石黒 章夫 訳: “知能の原理: 身体性に基づく構成論的アプローチ”, 共立出版 (2010).

8) Nolfi, S., and Floreano, D., “Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines”, *Cambridge, MA: MIT Press* (2004).

9) R. Smith, “Open dynamics engine v0.5 user guide”, URL http://www. ode.org/ode-docs (2008).

10) K. O. Stanley and Risto Miikkulainen, “Evolving Neural Networks through Augmenting Topologies”, *Cambridge, MA: MIT Press* (2002).