

修士論文の和文要旨

研究科・専攻	大学院 情報理工学研究科 情報・ネットワーク工学専攻 博士前期課程		
	渋谷 峻	学籍番号	1731081
論文題目	仮想物理世界における機械式論理回路を用いた人工生命モデルの開発		
要 旨			
<p>本研究では、仮想物理世界で動作する機械式論理回路を改良し、安定して動作する「歩く論理回路」を実装した。</p> <p>また、論理回路が無数にある環境の中で、回路同士が自然に組み上がる様子を観測し、将来人工生命モデルとして利用できることを提案する。</p> <p>既存の人工生命の研究では、コンピュータ上に仮想環境を構築し、その中で生命モデルを動作させる手法が取られている。環境を複雑に定義したり、生命モデルを動かす人工知能(AI)を実装することで、複雑なシミュレーションを可能にしている。しかし AI 部分は仮想環境の外で予め定義されているため、想定されない環境になった際に適合することは期待できない。しかし生命はそのような進化を実現している。仮に AI そのものを仮想環境内で実装すれば、そのような環境の変化に対応できる人工生命モデルを構築出来る可能性がある。</p> <p>一方で分子ロボティクスの分野では、分子を使って自己組織化や論理回路が作成されている。未来には体内に入る微細なロボットの実現が期待されるが、例えば DNA を使った反応は遅く複雑な回路を実装する上での問題点を議論するには実験環境が不十分である。仮想空間で実験出来れば、ロボット構築へ向けた技術的課題を先に検討出来る可能性がある。</p> <p>「歩く論理回路」の先行研究では、仮想物理世界で機械的に動作する論理回路に足パーツを取り付けることで移動可能な回路を実装した。しかし、動作が不安定であり、センサー等の機能は実現出来ていなかった。本研究で実装した歩く回路は、NAND/AND の出力を持った二種類のゲートのみを複数個組み合わせることで、歩くだけでなくセンサー機能も実現している。障害物に衝突すると後退することが出来る。本研究の歩く回路は先行研究のものより安定しており 200 倍以上高速に動作する。</p> <p>また、磁石のように引き付け合う力を加えることで、環境内で自動的にリングオシレータのような回路が組みあがることを観測した。今後大規模な環境で評価できれば、動き回るような回路が自然に出来上がり、環境内で進化する回路が生まれてくる可能性もある。本提案は、ほぼ同じ二種類のゲートのみで構成されているため、分子ロボティクスのように自己組織化してロボットを構築するという目的に有効であると考えられる。</p>			

平成30年度 修士論文

仮想物理世界における機械式論理回路を用いた
人工生命モデルの開発

電気通信大学 情報理工学研究科
情報・ネットワーク工学専攻
コンピュータサイエンスプログラム

学籍番号 1731081

氏名 渋谷 峻

指導教員 成見 哲

副指導教員 沼尾 雅之

概要

本研究では、仮想物理世界で動作する機械式論理回路を改良し、安定して動作する「歩く論理回路」を実装した。また、論理回路が無数にある環境の中で、回路同士が自然に組み上がる様子を観測し、将来人工生命モデルとして利用できることを提案する。

既存の人工生命の研究では、コンピュータ上に仮想環境を構築し、その中で生命モデルを動作させる手法が取られている。環境を複雑に定義したり、生命モデルを動かす人工知能 (AI) を実装することで、複雑なシミュレーションを可能にしている。しかし AI 部分は仮想環境の外で予め定義されているため、想定されない環境になった際に適合することは期待できない。しかし生命はそのような進化を実現している。仮に AI そのものを仮想環境内で実装すれば、そのような環境の変化に対応できる人工生命モデルを構築出来る可能性がある。

一方で分子ロボティクスの分野では、分子を使って自己組織化や論理回路が作成されている。未来には体内に入る微細なロボットの実現が期待されるが、例えば DNA を使った反応は遅く複雑な回路を実装する上での問題点を議論するには実験環境が不十分である。仮想空間で実験出来れば、ロボット構築へ向けた技術的課題を先に検出出来る可能性がある。

「歩く論理回路」の先行研究では、仮想物理世界で機械的に動作する論理回路に足パーツを取り付けることで移動可能な回路を実装した。しかし、動作が不安定であり、センサー等の機能は実現出来ていなかった。本研究で実装した歩く回路は、NAND/AND の出力を持った二種類のゲートのみを複数個組み合わせることで、歩くだけでなくセンサー機能も実現している。障害物に衝突すると後退することが出来る。本研究の歩く回路は先行研究のものより安定しており 200 倍以上高速に動作する。

また、磁石のように引き付け合う力を加えることで、環境内で自動的にリングオシレーターのような回路が組みあがることを観測した。今後大規模な環境で評価できれば、動き回るような回路が自然に出来上がり、環境内で進化する回路が生まれてくる可能性もある。本提案は、ほぼ同じ二種類のゲートのみで構成されているため、分子ロボティクスのように自己組織化してロボットを構築するという目的に有効であると考えられる。

目次

1	はじめに	5
1.1	背景	5
1.2	目的	5
1.3	本論文の構成	6
2	既存研究	7
2.1	人工生命に関する研究	7
2.1.1	自己複製や新陳代謝に関する研究	7
2.1.2	行動獲得に関する研究	7
2.1.3	人工生命モデルに関する研究	9
2.2	分子ロボティクスに関する研究	10
2.3	機械式論理回路に関する研究	10
3	先行研究の歩く回路	14
3.1	機械式立方体型 NAND/AND ゲート	14
3.1.1	動作の概要	14
3.1.2	故障の防止策	14
3.2	リングオシレータについて	14
3.3	先行研究での足パーツについて	16
3.4	歩く回路について	16
4	本研究で実装した歩く回路	20
4.1	機械式立方体型 NAND/AND ゲート	20
4.1.1	動作の概要	20
4.1.2	Unity 内での階層構造	22
4.1.3	Unity 内でのスクリプト	22
4.1.4	既存の立方体型 NAND/AND ゲートとの差異	23
4.2	歩く回路	24
4.2.1	足	24
4.2.2	リングオシレータとの接続	24
4.2.3	歩行動作の詳細	25
4.3	接触センサ付きの歩く回路	27

4.3.1	センサ機能	27
4.3.2	記憶回路	27
4.3.3	衝突したら停止する回路	29
4.3.4	衝突したら後退する回路	32
5	回路の自動生成	34
5.1	自然に結合と分裂を行う立方体型 NAND/AND ゲート	34
5.1.1	概要	34
5.1.2	入出力面に働く力	35
5.2	回路同士が自然に組み上がる環境	35
5.2.1	回転と移動が自由な回路	36
5.2.2	回転を制限した回路	36
6	評価	40
6.1	歩く回路の安定性について	40
6.1.1	歩く回路の実験結果	40
6.1.2	2つの歩く回路の考察	41
6.2	ロボットとしての機能の実現について	43
6.2.1	コンピュータ	43
6.2.2	センサ	43
6.2.3	アクチュエータ	43
6.2.4	構造	43
6.3	人工生命モデルとしての機能について	44
6.3.1	自己複製機能について	44
6.3.2	新陳代謝について	44
6.3.3	進化できるか	44
6.3.4	人工生命モデルとして提案できるか	44
7	まとめと今後の課題	46
7.1	まとめ	46
7.2	今後の課題	46
7.2.1	論理回路のみで AI を実現	46
7.2.2	新たな移動可能な回路の構造発見	47
7.2.3	回路のパラメータについて	47

7.2.4	人工生命モデルの実現	47
-------	----------------------	----

1 はじめに

本章では研究背景と目的を述べる。また、各章の概要についても紹介する。

1.1 背景

既存の人工生命の研究ではコンピュータ上に構築した仮想環境の中で生命モデルを動作させる手法が取られる。例えば、生命モデルの人工知能 (AI) は仮想環境外からプログラムを与えているため、あらかじめ定義した以上の動作が得られることは少ない。しかし現実の生命現象においては、知能は環境の中にある。仮想環境内で AI 部分も実装できれば、環境の影響を受けて AI が進化するなど想定外の動作が得られる可能性がある。

一方で分子ロボティクスという領域がある。従来の機械によるロボット工学は、加工したパーツから望みのものを作るトップダウンの手法をとるのに対し、分子ロボティクスでは、物質を構成する分子そのものの性質をプログラミングすることで、その物質自身が望みのものに自己組織化し、ボトムアップで微細なロボットやコンピュータを作り上げようという試みである [1]。生体内での計算や診断、治療、人工細胞構築、分子メモリなどへの応用が期待されているが、「分子がどのように反応しているか確認することが難しい」「制御が難しい」「実験に多大な時間が必要」などの問題がある。コンピュータで仮想的に分子を再現することで制御や解析が容易にできれば、研究の発展に繋がる可能性がある。

先行研究では、機械式論理回路¹を仮想物理世界の中で動作させ、動き回る回路が実装された。しかし、人工生命の構成要素として動き回るという 1 つの機能だけが実装されている。また、動作が不安定であるなどの様々な問題があった。

1.2 目的

本研究の目的は、第一に既存研究で実装された機械式論理回路を改良し、安定して動き回る回路を実現することである。先行研究では Unity²を用いて仮想物理世界を構築し、生物のように歩き回る回路を実装したが、動作が不安定であったため、引き続き Unity を用いて改良することを目指す。機械式論理回路は、演算機能に加えて情報や力を伝達する配線としても利用できるため、ロボットの構成要素であるセンサ、アクチュエータ、コンピュータ、構造の機能の全てを論理回路のみで実現することができる。次に、機械式論理回路が多数ある世界を構築し、独立した回路同士が接続したり分裂することで新しい構造を作るなど、生命らしい動作を実現する。今後回路を大規模化して AI を搭載した人

¹機械式論理回路とは電気信号ではなく物理的な接触や力を情報として演算する論理回路である。

²統合開発環境を内蔵するゲームエンジンであり、物理エンジンが組み込まれているため物理シミュレーションを容易に行える [2]

工生命を実現し，この論理回路を基に分子モデルを作って置き換えることで，分子ロボット実現のためのハードルを下げることを目標としている．

1.3 本論文の構成

本論文の構成は以下のようになっている．

1. はじめに

研究背景と先行研究について述べ，本研究の目的を説明する．

2. 既存研究

人工生命や，分子ロボット，先行研究で行われてきた歩く回路について説明し，本研究の新規性を記載する．

3. 先行研究の歩く回路

先行研究で作成された歩く回路について詳細に説明し，課題点について記載する．

4. 本研究で実装した歩く回路

本研究で実装した立方体型 NAND/AND ゲートと歩く回路について記載する．

5. 回路の自動生成

回路が自然に組み上がる環境を実装し，様々な条件で行なった実験について説明する．

6. 評価

実装した歩く回路の評価と人工生命モデルについて考察する．

7. まとめと今後の課題

本研究のまとめと今後の課題を記載する．

2 既存研究

本章では、人工生命のシミュレーションを行う研究や、先行して行われた機械式論理回路の研究を紹介し、本研究との差異と本研究の新規性について説明する。

2.1 人工生命に関する研究

2.1.1 自己複製や新陳代謝に関する研究

自己複製はフォンノイマンやチューリングマシンに始まるコンピュータの自己複製の歴史の中で研究されてきた。自己複製能力は生命の基準の一つと考えられ、人工生命において様々なアプローチで自己複製能力を実現する研究が盛んである [3]。しかし、ロボット工学の分野では、自己複製ロボットの実現は達成できていない課題の一つである。自己複製ロボットを実現するには、材料を調達する機能、論理回路のような演算装置、作成した部品を移動させるパーツなどの特定のパーツが必要とされている。William M. Stevens は仮想世界でこれらの機能を持った立方体状のパーツを作り、図 1 に示す自己複製ロボットのシミュレーションを行った [4][5]。また、図 2 のように、自身を構成する部品が無数にある環境を作り、自由に動き回る部品が、他の部品と衝突すると結合したり、分裂するといったルールを設けることで、自己複製ロボットを実現する物理シミュレーションも行われている [6]。さらに、図 3 のように既に完成している構造から構成パーツがランダムに外れ、外れた箇所を別のパーツが結合することを繰り返す新陳代謝のシミュレーションも行っている [7]。このシミュレーションでは基本の結合法則は自己複製のシミュレーションと同等であるが、初めから 10 個のパーツからなる構造とそれぞれのパーツを無数に配置している。構造を構成するパーツはランダムに分離し、新たなパーツが分離した箇所で結合を繰り返すことで新陳代謝が行われる。

複数種類のパーツや特定の条件下で外部からプログラムを与えることで、自己複製や新陳代謝のシミュレーションを行っている。本研究では外部からプログラムを与えずに、論理回路のみを用いて自己複製や新陳代謝を再現することを目指すため、既存研究とは異なる。

2.1.2 行動獲得に関する研究

近年、人工生命の研究ではコンピュータ上に人工の環境を構築し、その中で生命モデルを動作させる手法が取られている。Moore らは、仮想環境内にカンガルーラットのモデルを配置し、尻尾と両足を跳ねるように使うことを繰り返し学習させることで二足ホッピング動作を獲得させている [8]。尻尾の長さや体の重さ、関節の可動域などを複雑に定義し、様々なシミュレーションを行うことで自然界のカンガルーラットに近い動作や、自

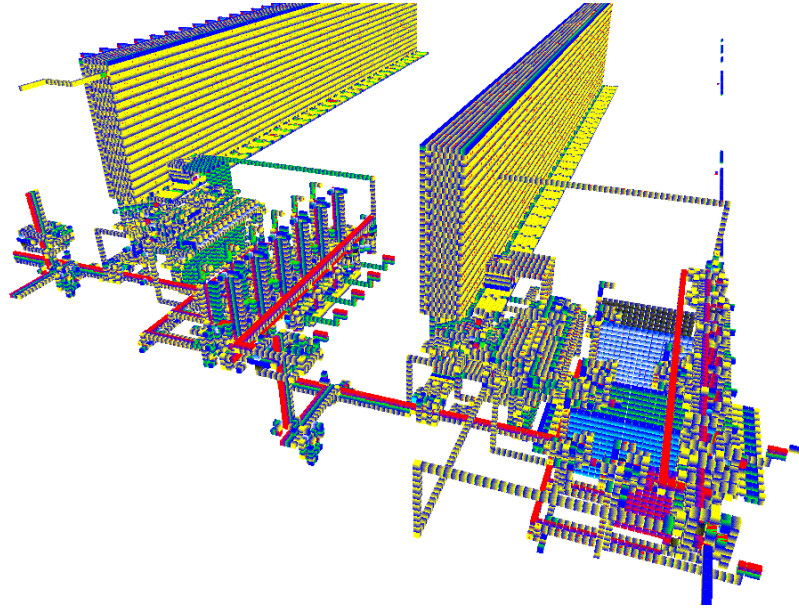


図 1: 自己複製ロボットのシミュレーション (参考文献 [4] より引用)

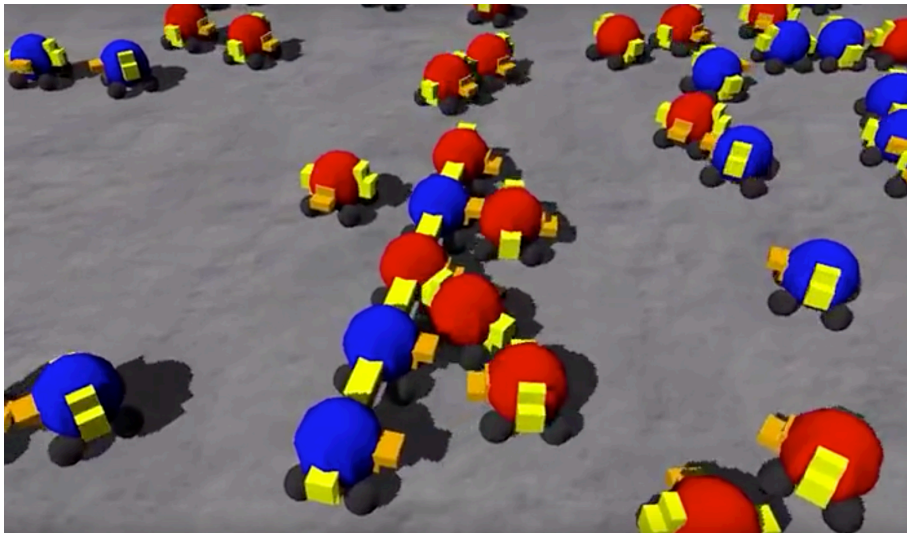


図 2: 赤と青の球体が結合し、自己複製するシミュレーション (参考文献 [6] より引用)

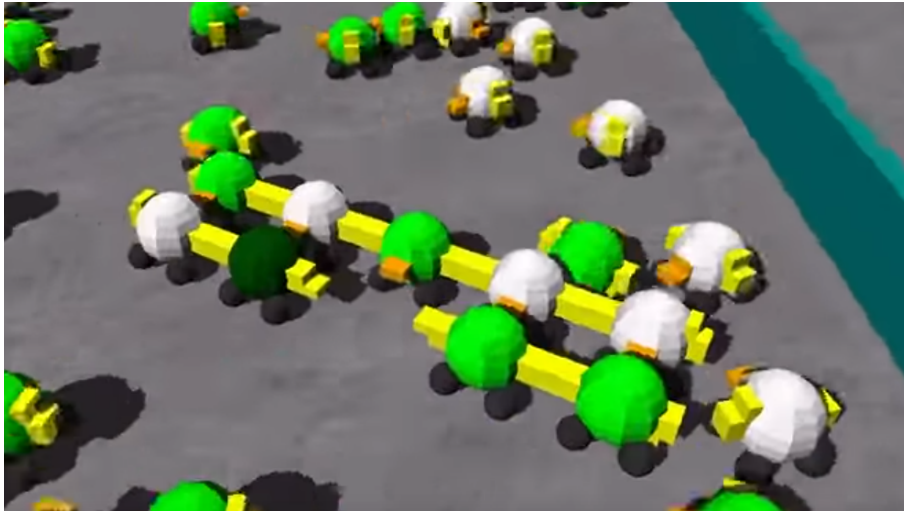


図 3: 緑と白の球体が分離と結合を繰り返し，新陳代謝を行うシミュレーション (参考文献 [7] より引用)

自然界ではみられない新しい発見なども得られている (図 4). 10 秒間で移動できた距離を評価することでカンガルーラットの AI は四肢と尻尾を効率よく動かす移動方法を獲得するが，あらかじめプログラムで定義された評価関数によって学習するので何度繰り返しても最終的には同じ結果が得られると考えられる。

本研究では，現実の生命現象のように知能を環境の中で実装することで，環境の影響を受けて AI が進化するなどの想定外の動作が得られる可能性がある。

2.1.3 人工生命モデルに関する研究

Ingo Althöfer らは，バケツいっぱいのレゴブロック³を洗濯機で洗濯する実験を行った [10]. バラバラであったレゴブロックは洗濯機内で結合し，複雑な構造が形成された。ほとんどのブロックが洗濯機の中で 2 つ以上結合し，中には 6 つのブロックからなる複合体も作られた (図 5). ブロックがランダムで結合する様子から乱数生成器とみなせ，原始的なモンテカルロ法のシミュレーションに利用可能であることや，Miller の実験モデル⁴や，人工生命の実験装置として利用できることを提案している。洗濯機の内部ではブロックは何度も結合と分裂を繰り返し，出来た複合体を再度洗浄を繰り返すことで様々な安定した構造を発見した。これは，生態系における適合種のシミュレーションと言える。さらに洗浄前に人工的に同じ構造の複合体をいくつか洗濯機に入れることでそれらがあたか

³プラスチック製の組み立てブロック玩具 [9]

⁴地球で最初に生命が誕生する様子をフラスコ内で再現する実験 [12]

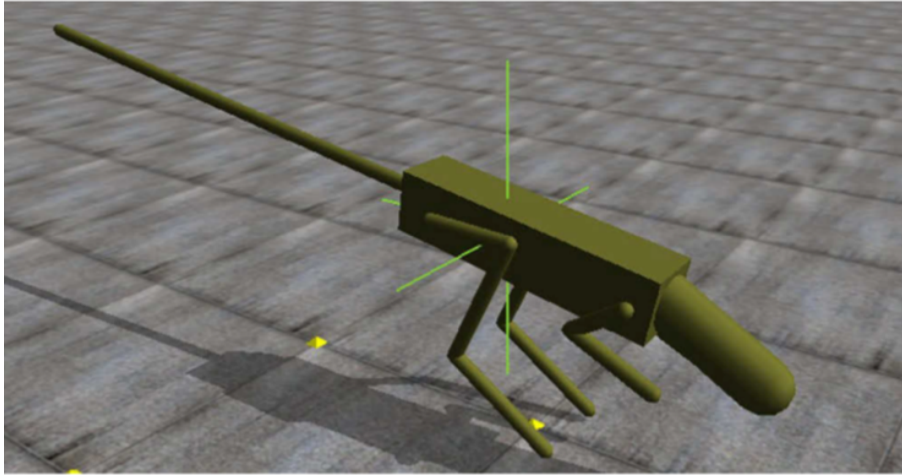


図 4: 2 足ホッピング動作を獲得するカンガルーラットの骨格モデル (参考文献 [8] より引用)

も酵素の役割を果たし、狙った構造の結合体や突然変異のように新しい構造を発見する現象が見られた。

本研究では機械式論理回路をランダムに動かすことで、結合と分裂を繰り返すことによって新たな構造を作るだけでなく、新たな動きを獲得することを目指す。

2.2 分子ロボティクスに関する研究

大原らは人工細胞中で DNA 分子を用いた DNA コンピュータで、論理ゲートの 1 つである AND 演算を行った [13]。DNA 二重螺旋形成に伴う RNA 転写反応には、DNA プロモータ配列を不完全なものから完全なものにすることで、RNA 転写反応を不活性状態から活性状態に切り替えるスイッチ機能ができる。このスイッチ機能を用いて AND ゲートを実現している。2 つの DNA を入力分子として与えた時だけ RNA 転写反応が活性状態になり、出力 RNA の合成が行われることで AND 演算が実行される (図 6)。

分子の制御には多大な時間が必要であるため、本研究では論理回路を基に分子モデルを作って置き換えることにより、分子ロボット実現のためのハードルを下げることを目標としている。

2.3 機械式論理回路に関する研究

仮想物理世界で機械式論理回路を動作させる研究は過去に行われている。瀬戸口らは Unity を用いて力学的に動作する機械式 NAND ゲートを実装し、さらに論理回路同士を

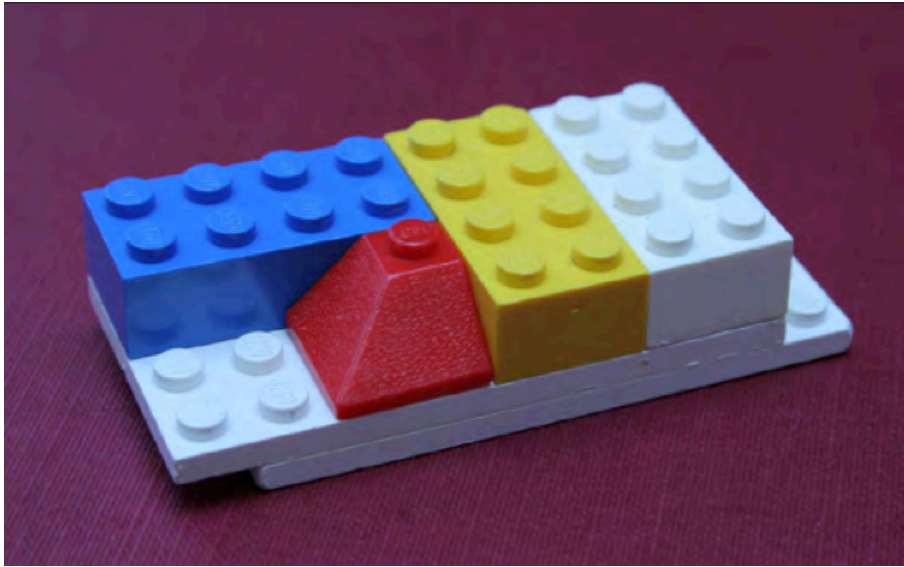


図 5: 洗濯機内部で結合されたレゴブロック (参考文献 [10] より引用)

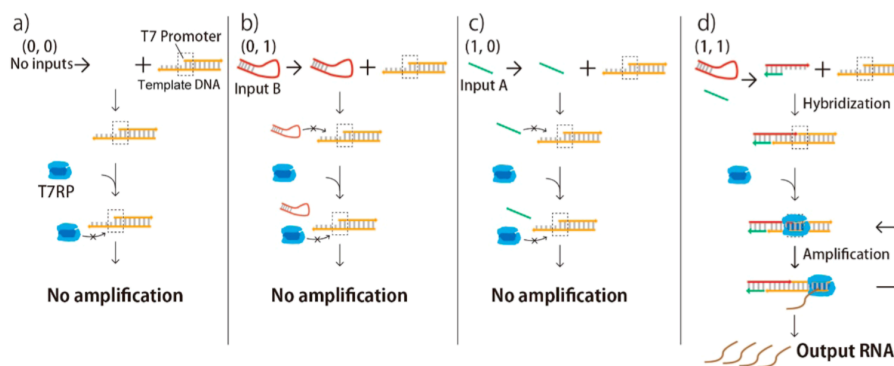


図 6: DNA ロジックゲートの反応の様子 (参考文献 [8] より引用)

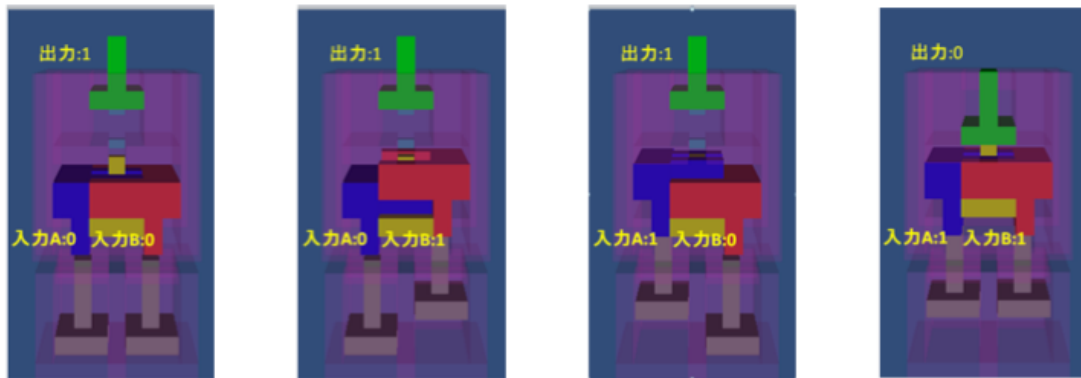


図 7: 箱型 NAND ゲート (参考文献 [14] より引用)

繋ぐ配線パーツとして動作する箱型 NAND ゲートの実装を行った [14](図 7)。故障のしやすさや他のゲートとの干渉を改善するために半透明のカバーを取り付け安定して動作するようになった。さらに立方体型の NAND ゲートを作成し、組み合わせて配置することで論理作用素としてだけでなく配線パーツや駆動部品としても働くような機械式論理回路の提案を行った。しかし、移動するとゲートが壊れて正常に動作しなくなることや、回路によっては構造上配線できないことなどの問題があった。

神澤らは仮想物理世界で動作する論理回路を改良し、従来より安定して動作する立方体型の NAND/AND ゲートを作成し、仮想環境で動作する歩く論理回路を実装した [15](図 8)。既存の立方体型 NAND ゲートと比較し、AND の出力も可能になったため、配線パーツとしての利用も容易になった。さらにこれらの立方体型 NAND/AND ゲートを組み合わせたリングオシレータに足パーツを取り付けることで、移動することが可能な論理回路を実装した。足パーツはリングオシレータの出力に合わせて伸び縮みとトルクがかかるようにプログラムされており、4つの足の伸び縮みのタイミングが合うと歩くことができる。

本研究では以下の改定を行なっている点で異なる。

- 足パーツを別途作成していない
- ロボットの構成要素のうちのセンサ機能を実装している
- 動作が安定している

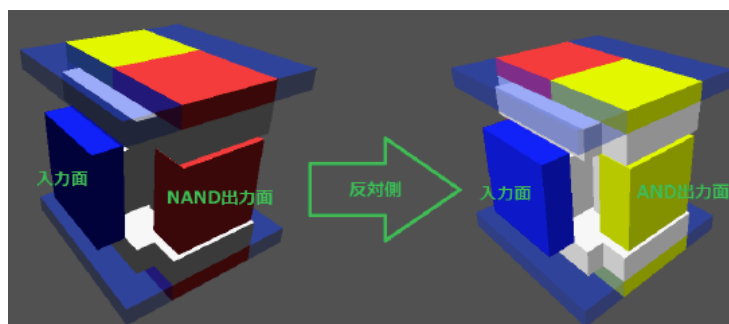


図 8: 立方体型 NAND/AND ゲート (参考文献 [15] より引用)

3 先行研究の歩く回路

先行研究において歩く回路は実現されていたが、様々な課題や実装されていない機能がいくつかある。本章では先行研究で実装された歩く回路について詳しく説明する。

3.1 機械式立方体型 NAND/AND ゲート

3.1.1 動作の概要

既存の立方体型 NAND/AND ゲートの外観は図 8 である。立方体型のゲートには青い入力面と赤と黄の出力面が存在する。入力面に物理的な接触を受けると入力とみなし、入力の結果によって NAND 出力を表す赤い面や、AND 出力を表す黄色い面からブロックが出し入れされる。この機構については 4.1.1 節で詳解する。AND の出力面は入力された値をそのまま伝達する配線パーツとしても利用ができるため、多数の立方体型ゲートの入力面と出力面を組み合わせることで様々な回路を実現することができる。

3.1.2 故障の防止策

論理回路の入力面には力を受けると引っ込み、そうでない時は飛び出るバネのような機構が実装されている。出力面に関しても同様に、出力ブロックが出し入れされる方向に力がかかるように設計されている。出力面は入力面を押し込めるように、入力面のバネより強い力が設定されており、常に出力面のどちらかに引っ張られる力がかかっている。そのためゲート本体から出力ブロックが飛び出してしまわないような様々な工夫がされている。工夫の 1 つとして出力ブロックが本体から飛び出してしまわないように内部にツメが作られている (図 9)。他にもマーカーと呼ばれる小さいオブジェクトを出力面から少し離して設置し、マーカーと出力ブロックとの距離に比例して出力ブロックにかかる力を制御している (図 10)。さらにゲートが軽いと出力ブロックにかかる力でゲート全体が動いてしまうため、ゲート本体の質量を上げることで、接地していれば摩擦力を利用してゲートが動き出してしまうのを防いでいる。

これらの防止策を施しているが、リングオシレータや歩く回路において、多数のゲート同士が強い力や重さで高速に動くと、ゲート同士がめり込んでしまい故障してしまうことがしばしば見られ、動作が不安定な原因となっていた。

3.2 リングオシレータについて

リングオシレータとは、入力された論理値を反転させて入力に戻すことで論理値が反転し続ける回路である。図 11 のように奇数個の NOT ゲートを並べることで実装できる。図 12 のように立方体型ゲート 6 つを用いることでリングオシレータを実現することがで

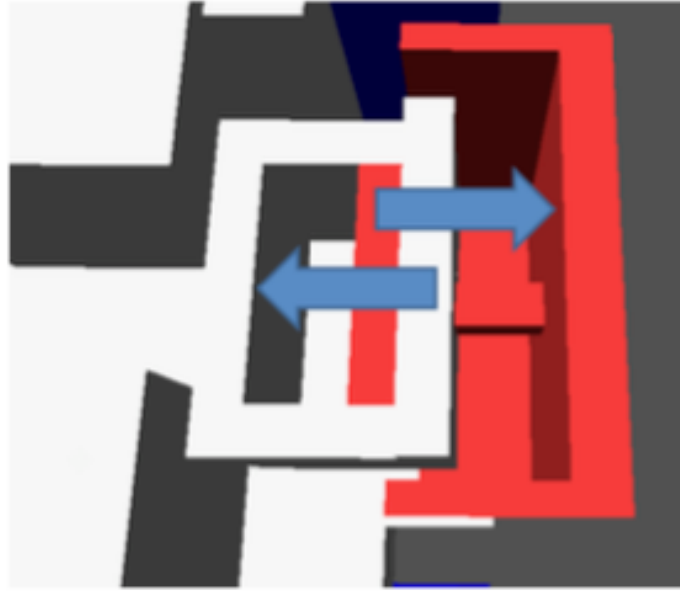


図 9: 出力面が本体から飛び出さないように内部に作られたツメ (参考文献 [15] より引用)

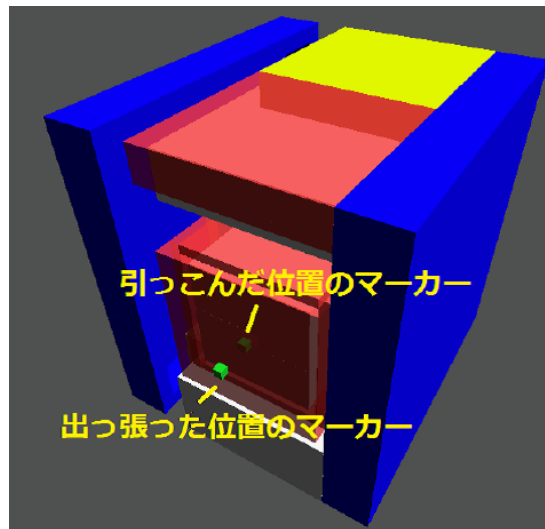


図 10: 出力ブロックに働く力を制御するマーカー (参考文献 [15] より引用)

き、回路図は図 13 となっている。

3.3 先行研究での足パーツについて

先行研究では立方体型ゲートの他に、歩くための足パーツを別途作成している。図 14 は先行研究で作成された足パーツである。入出力面の機能は立方体型ゲートと同等であるが、片方の出力面からブロックが伸びる距離が長くなっている。この足パーツに入力が伝わると足が伸縮し、歩く回路全体を持ち上げることができる。さらに足パーツには、入力面を軸に左右 10 度までの制限で回転を許可されており、足パーツが伸びた時は右、縮んでいる時は左向きにトルクをかけるようになっている。

3.4 歩く回路について

図 12 のリングオシレータと図 14 の足パーツを組み合わせることで歩く回路を実現している。歩く回路の外観は図 15,16 のようになっており、リングオシレータに 4 つの足パーツが取り付けられている。リングオシレータの動作に合わせて 4 つの足が動作するため、1 周期に 1 度それぞれの足が伸縮する。

足パーツとリングオシレータの接続には Unity の Hinge Joint を利用している。Hinge Joint は 2 つのオブジェクトを 1 つの軸で固定し、その軸を中心に回転させることができる。足パーツは伸縮に応じて左右にトルクがかかるようになっているため、伸びている足は左から右にかかるトルクを利用して地面を蹴って左へ進み、縮んでいる足は宙で右から左に足を回転させ、次に地面を蹴る時に備える。

前進するためには対角線上の 2 つの足が同時に動作する必要があり、対角線上の 2 つの足が同時に地面を蹴る時、回路全体が持ち上がり歩行することができる。本リングオシレータでは各足の同時制御ができていないため、足を動かすタイミングがうまく噛みあつた時だけ歩くことができる。

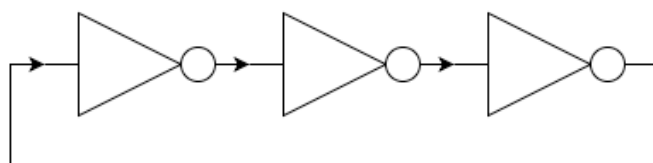


図 11: リングオシレータの回路図

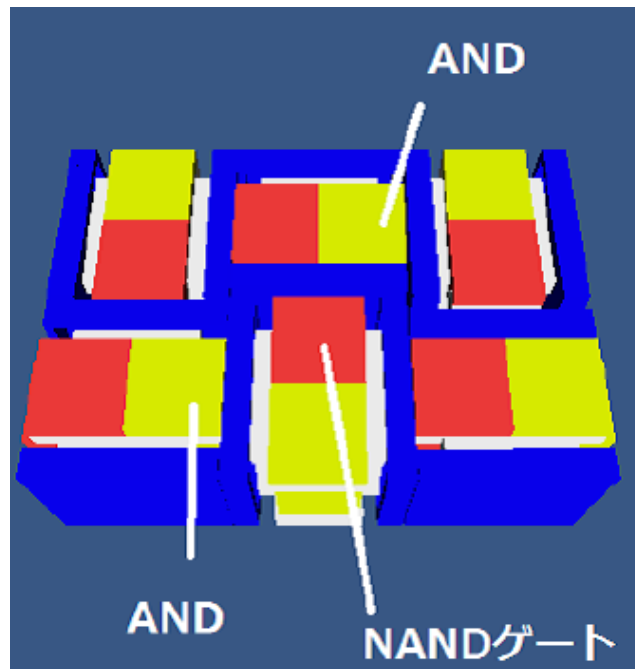


図 12: 機械式論理回路で作成されたリングオシレータ (参考文献 [15] より引用)

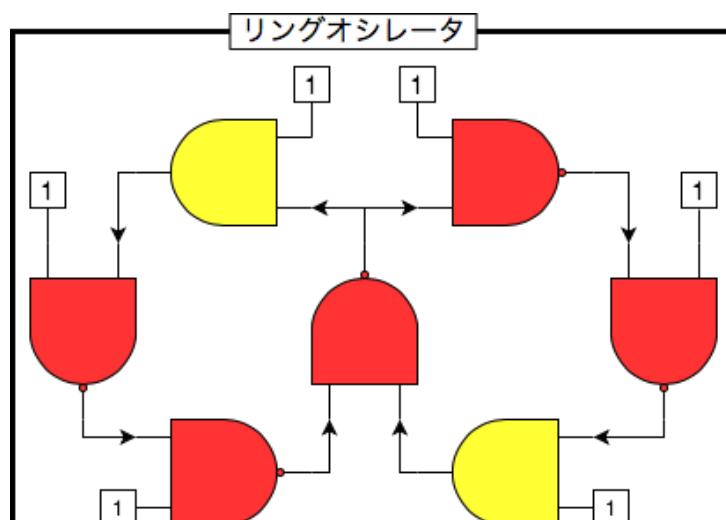


図 13: 機械式論理回路で作成されたリングオシレータの回路図

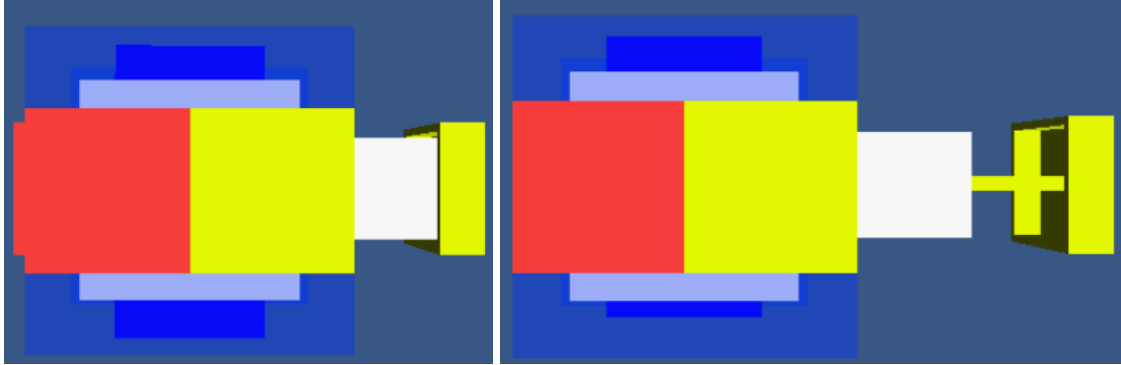


図 14: 足パーツが伸び縮みする様子 (参考文献 [15] より引用)

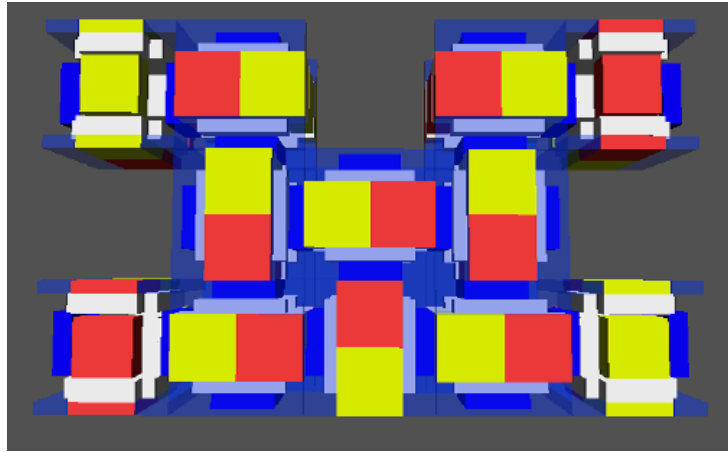


図 15: 歩く回路上から見た様子 (参考文献 [15] より引用)

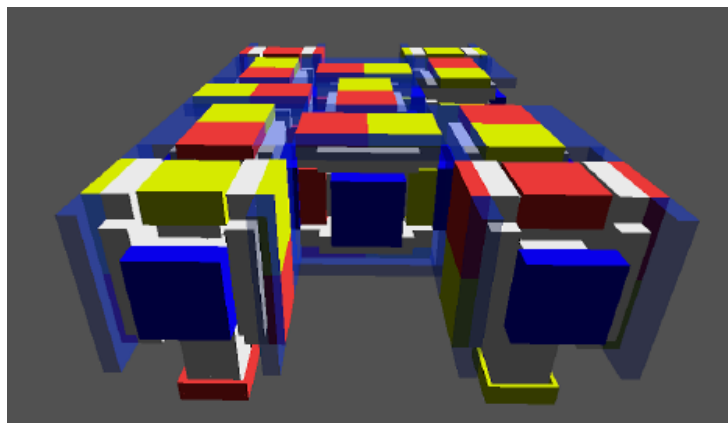


図 16: 歩く回路横から見た様子 (参考文献 [15] より引用)

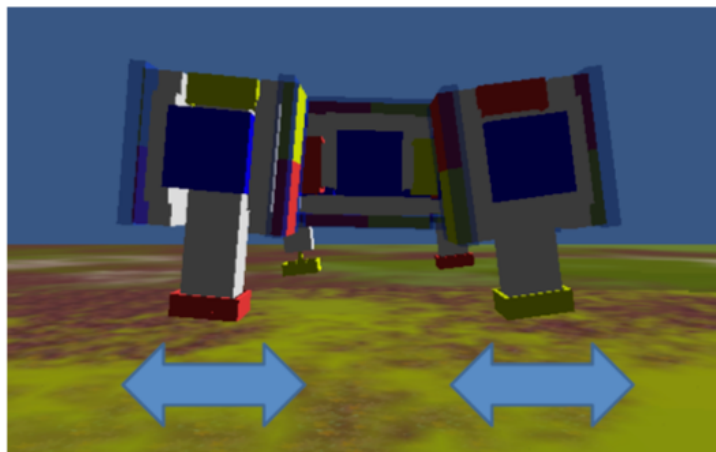


図 17: 先行研究の歩く回路 (参考文献 [15] より引用)

4 本研究で実装した歩く回路

本研究では既存の機械式論理回路を改良し、歩く回路をより安定して動作するように実装した。本章では先行研究での問題点と本研究で改良した点を説明する。

4.1 機械式立方体型 NAND/AND ゲート

4.1.1 動作の概要

本研究では図 18,19 のような立方体型 NAND/AND ゲートを作成した。青い 2 つの入力面と赤い NAND 出力面、黄色い AND 出力面を持っている。

入力面に物理的な接触を受けると入力とみなし、その結果によって出力面が伸縮するようになっている。ゲートの真理値表は表 1 の通りで入力面に接触があるときを 1、接触がないときを 0 とし、出力面が伸びている時を 1、縮んでいる時を 0 とした。図 18(左) は 2 つの青い面どちらにも接触がなく、入力が $\{0,0\}$ であるため、赤い NAND 出力が伸び、黄色い AND 出力が縮んでいる。図 18(右) は右側の 1 つの青い面のみに接触があり、入力が $\{0,1\}$ であるため、赤い NAND 出力が伸び、黄色い AND 出力が縮んでいく。図 19(左) は左側の 1 つの青い面のみに接触があり、入力が $\{1,0\}$ であるため、赤い NAND 出力が伸び、黄色い AND 出力が縮んでいる。図 19(右) は 2 つの青い面どちらにも接触があり、入力 $\{1,1\}$ となるため、黄色い AND 出力が伸び、赤い NAND 出力が縮んでいる。

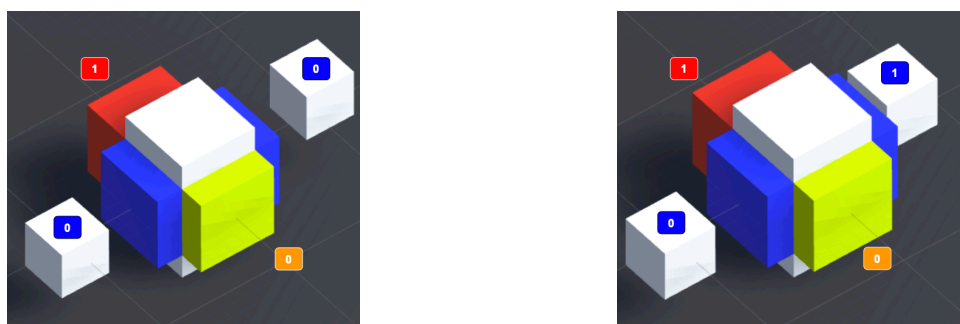


図 18: 本研究で実装した立方体型 NAND/AND ゲート
入力 $\{0,0\}$ の様子 (左), 入力 $\{0,1\}$ の様子 (右)

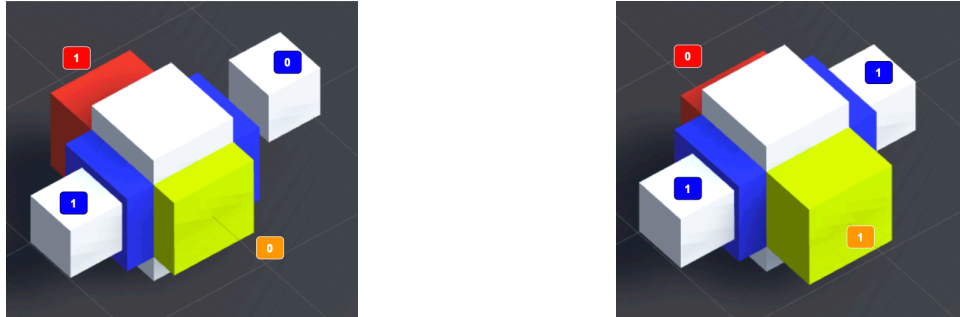


図 19: 本研究で実装した立方体型 NAND/AND ゲート
 入力 {1,0} の様子 (左), 入力 {1,1} の様子 (右)

表 1: NAND/AND ゲートの真理値表

入力 1(青)	入力 2(青)	NAND(赤)	AND(黄)
0(非接触)	0(非接触)	1(飛び出す)	0(引っ込む)
0(非接触)	1(接触)	1(飛び出す)	0(引っ込む)
1(接触)	0(非接触)	1(飛び出す)	0(引っ込む)
1(接触)	1(接触)	0(引っ込む)	1(飛び出す)

4.1.2 Unity 内での階層構造

実装した立方体型ゲートには様々なパラメータが存在する。図 20 に示す通り、立方体型ゲートは 5 つのパーツから構成されている。中央の白い *MainCube* の周りに青い *Input1*, *Input2*, 赤い *NAND*, そして黄色い *AND* パーツが取り付けられている。全てのパーツは Unity の *Cube* オブジェクトから構成されており、それぞれの構成パーツは、自身を動かすスクリプトとともに *GameObject* として *NAblock* という名前で 1 つにまとめられ、下のような親子関係になっている。

NAblock

5 つのブロックとスクリプトをまとめる親オブジェクト

- *blockController.cs*
ゲートを動かすメインスクリプト
- *MainCube*
入出力ブロックを取り付ける中央の白いブロック
- *Input1* 入力を感じ取る青いブロック
 - * *inputBlock.cs*
入力があると呼ばれるイベントが記述してあるスクリプト
- *Input2*
 - * *inputBlock.cs*
- *NandOut*
NAND 出力を表す赤いブロック
- *AndOut*
AND 出力を表す黄色いブロック

4.1.3 Unity 内でのスクリプト

立方体型 NAND/AND ゲートを主に動かすスクリプト *blockController* について説明する。スクリプトにはパブリック変数として *NandMoveDis* と *AndMoveDis* があり、これは Unity のプロジェクト画面から編集することができる。この変数を変更することで、それぞれの出力ブロックが飛び出す距離を変更することができる。ブロックの出力には指定した距離まで伸び縮みするように実装しており、Unity の *Vector3.Lerp* を用いている。こ

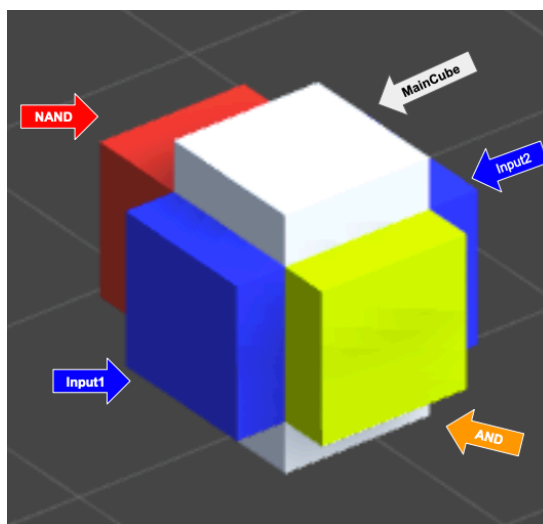


図 20: 立方体型ゲートの構成パーツ

れは出力ブロックの現在の形状と目的の形状を線形補間することができるため、滑らかに出力ブロックを伸縮できる。

入力ブロックでは *inputBlock* というスクリプトが動いている。このスクリプトには他のオブジェクトと接触があったときに呼び出される *OnTriggerEnter* と、接触していたオブジェクトが離れた時に呼び出される *OnTriggerExit* という2つのイベントが定義されている。他のオブジェクトとの接触や分離が起きると、親スクリプトである *blockController* に入力値の変化を伝える。

4.1.4 既存の立方体型 NAND/AND ゲートとの差異

先行研究では強い力でめり込んでしまうなどの原因で不安定であったため、本研究での立方体型ゲートでは、強い力を働かせることなく動作するように設計した。入力面ではバネは採用せず、入力面にオブジェクトが接触する、もしくは出力状態の出力ブロックが接触したか否かで入力されたか判定し、出力面の出し入れは出力ブロックに力をかけるのではなく出力ブロックの形状を線形補間しながら変形させることで実現した。出力面の出し入れに合わせて出力ブロックが伸び縮みするため、力をゲートにかけることなく他のゲートへ情報を伝達することができる。これらの改良によりゲートの内部では力が働かないため本体を軽くしても動き出すことはなく、ゲート同士の入出力のやりとりでもごく小さな衝撃のみで行うことができるため、大規模化してもより安定して動作することができる。

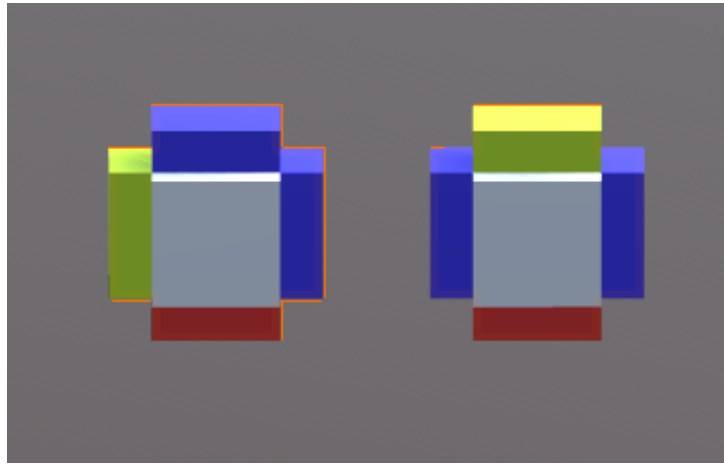


図 21: 立方体型 NAND/AND ゲート L 型 (左) と I 型 (右)

4.2 歩く回路

4.2.1 足

足を作るためにはまず図 21 のような 2 種類の入出力の配置の異なる立方体型ゲートを実装した。図 21(左) を L 型, (右) を I 型と呼び, 2 つのゲートは, 入出力面の配置が異なるだけで機能はどちらも同じである。I 型のゲートは, 入力と出力が互いに側面に並ぶため, 回路は一直線に配線することはできず, 必ず 90 度曲げて配線するという特徴がある。L 型のゲートは, 同一直線上に入力面と出力面が並んでいるため, 図 22 のように一直線上に配線することが可能になる。

伸びていない出力ブロックは入力面に接触しても入力を受け取らないため, 図 22(左) のように NAND のみが伸びた状態になっている。出力と触れていない入力全てに与えると図 22(右) のようになり, 出力距離を 3 倍に伸ばすことができる。

隣接する各ゲートの出力面と入力面は, Unity の Fixed Joint を用いて接続している。Fixed Joint は 2 つのオブジェクトを固定して接続することができ, オブジェクト間の結合の強さや, 結合が解除されて離れてしまう強さなども指定できる。回路が歩くためには足の伸縮で回路本体を持ち上げる必要があるため, 本研究ではこれをリングオシレータに取り付ける事で歩く回路の足とした。

4.2.2 リングオシレータとの接続

本研究で実装した歩く回路を図 23,24 に示す。6 つのゲートからなるリングオシレータに 3 つのゲートからなる足を 4 つ接続することで実装している。リングオシレータは図

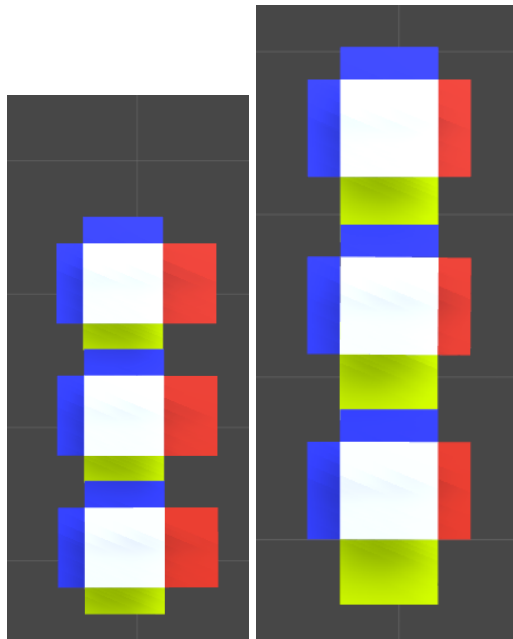


図 22: 歩く回路の足が伸縮する様子

13 と同様の回路であり，歩く回路の該当部分と回路図を合わせたものを図 25 に示す．6 つのゲートからなるリングオシレータのうち 4 つのゲートの片方の出力がリングオシレータの動作には使われていないため，そこに足を取り付けた．これらの出力はリングオシレータの動作には使われていないが，リングオシレータの周期に合わせて伸縮を行う．4 つの足のうちリングオシレータを挟んで向かい合う 2 つの足を前足，隣り合う 2 つの足を後足と呼び，前足が取り付けられている方向に回路は歩行する．

4.2.3 歩行動作の詳細

リングオシレータの論理反転し続ける出力により，4 つの足がそれぞれ伸縮する．前足と後足にはそれぞれ役割があり，次の手順を繰り返すことで回路は歩行する．

1. リングオシレータから出力ブロックが飛び出し後足を押す
2. 設地している後足がリングオシレータを押し返し，リングオシレータは出力と逆向きに力がかかる (図 26)
3. 前足が上がるとリングオシレータにかかる力で全体が前進する
4. 前足が下りて地面と接触し，停止する (図 27)

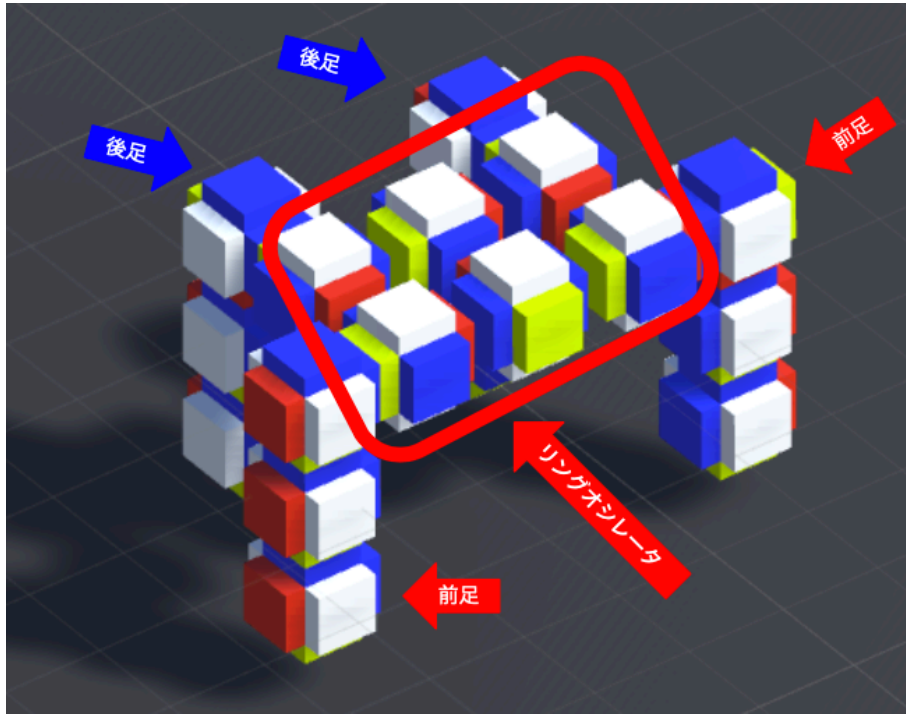


図 23: 本研究で作成した歩く回路

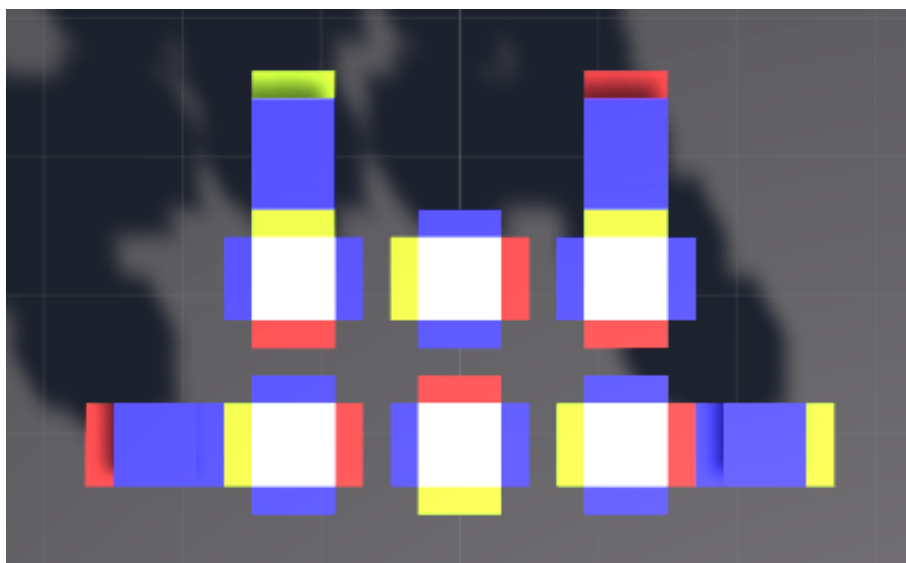


図 24: 歩く回路を上から見た様子

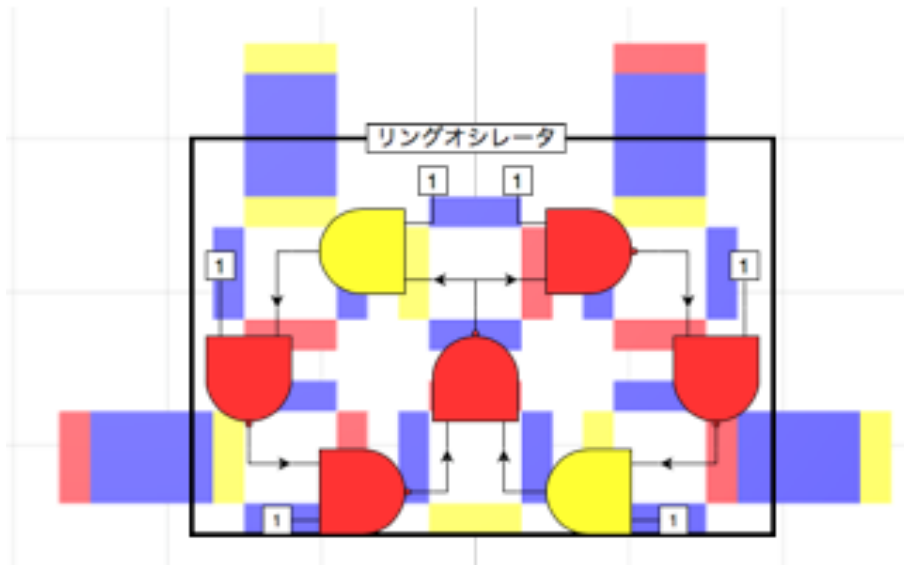


図 25: 歩く回路の上からの様子と回路図

リングオシレータから後足への出力はそのまま後足を押すため、後足が接地していればリングオシレータを押し返し、前進する力がかかる(図 26)。一方リングオシレータは前足も同様に押すため、進行方向右または左に前足から押し返す力が働くが、2つの前足は向かい合っているため押し返す力は釣り合う。そのため回路は進行方向に曲がる事なく進むことができる。前足が縮み、後ろ足が伸びると回路は前進し、前足が伸びると歩行は停止する(図 27)。

4.3 接触センサ付きの歩く回路

4.3.1 センサ機能

リングオシレータの外側についている入力面は常に1が入力されていることになっているが、障害物等などの接触を感知して値を反転させればリングオシレータの動作と停止を制御できる。図 28 の回路を実装できればセンサ機能が実現できる。障害物との接触がなければセンサ相当部からは1が入力されるためリングオシレータは動作し続け、障害物に衝突するとセンサ相当部からの入力反転し、リングオシレータは停止する。

4.3.2 記憶回路

障害物を回避する機構を作るには記憶回路が必要である。図 29 の RS ラッチ回路を立方体型ゲートのみで作成した(図 30)。RS ラッチは1ビットの状態を記録できる回路であ

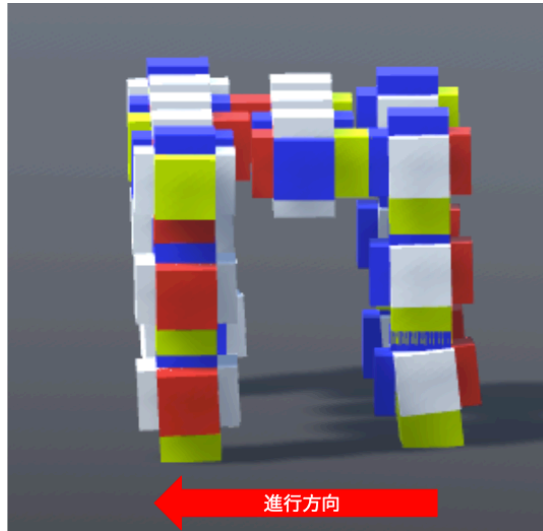


図 26: 後足が伸び進行方向に力がかかっている様子

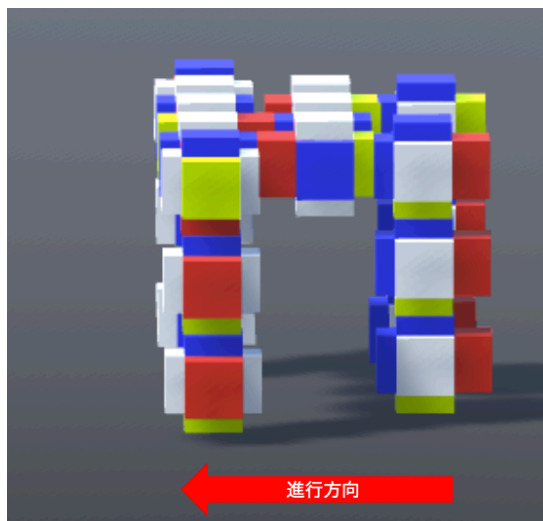


図 27: 後足が縮み歩行が停止している様子

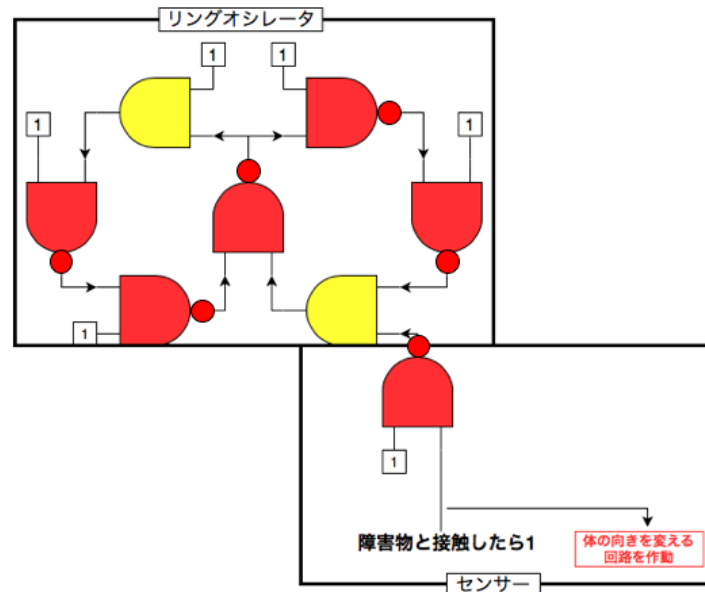


図 28: リングオシレータに NAND/AND ゲートで作られたセンサを取り付けた回路図

り、表 2 に示す動作をする。実際に動作させた様子が図 31,32 であり、S で与えた状態を保持し続けることができる。この RS ラッチの S の入力面を接触センサとして利用する。

4.3.3 衝突したら停止する回路

作成した RS ラッチを用いて、「障害物に衝突したら停止する」歩く回路を実装した (図 33)。RS ラッチのリセット R は今回は使わないため、障害物と接触ないようにブロックの配置を変更してある。RS ラッチは S に入力があるまで Q は 0、 $Q\sim$ は 1 を出力する特性がある。このため $Q\sim$ と歩く回路のリングオシレータの入力面の 1 つと接続することで、障害物と接触するまで前進する。障害物と RS ラッチの入力 S が衝突すると、Q は 1、

表 2: RS ラッチの真理値表

S	R	Q	$Q\sim$
0	0	状態保持	
0	1	0	1
1	0	1	0
1	1	禁止	

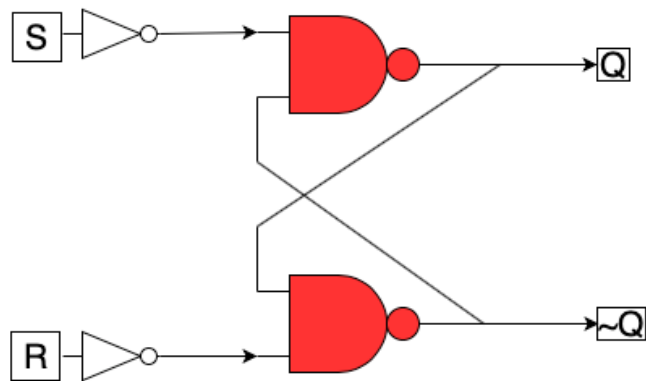


図 29: RS ラッチ回路図

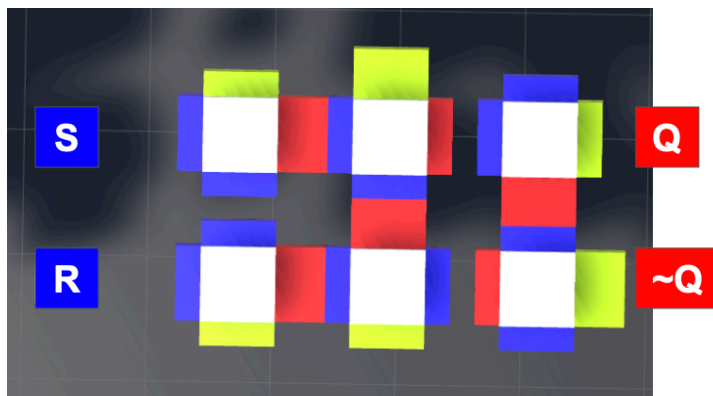


図 30: 立方体型ゲートで実装した RS ラッチ

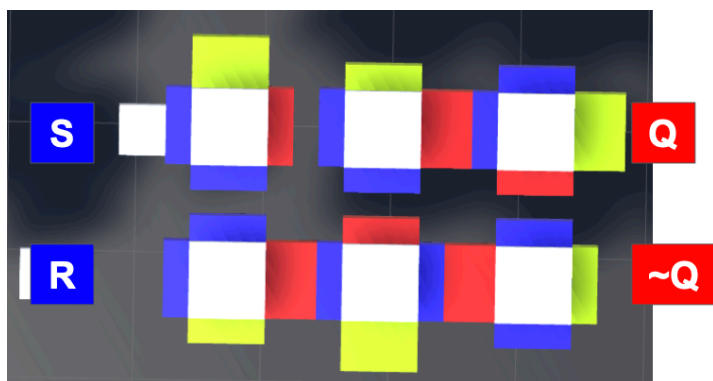


図 31: S に入力 1 を入れた様子

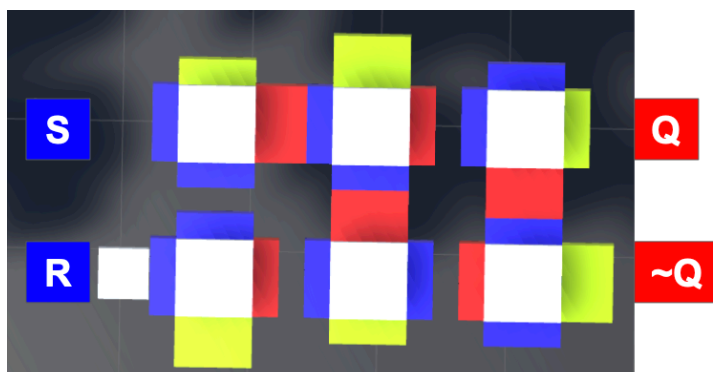


図 32: R に入力 1 を与えた様子

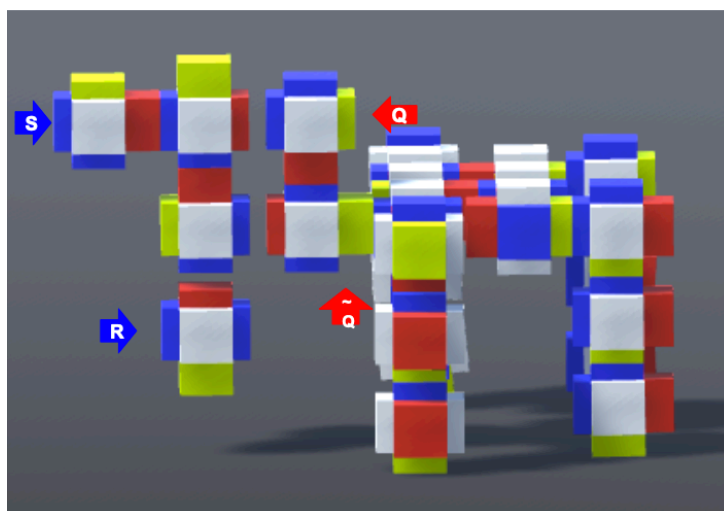


図 33: 接触センサを取り付けた歩く回路

$Q\sim$ からは 0 が出力されるため、リングオシレータは停止し歩行が止まる (図 34)。一度回路が停止し、入力 S と障害物が離れ入力なくなっても、状態を記憶した RS ラッチは Q から 1, $Q\sim$ から 0 を出力し続けているため、回路が再び前進することはない。

この回路では RS ラッチのリセット R を使っていないが、回路の後方に R の入力面を取り付けることで、「停止中の回路の後方から障害物が迫ってきたら、前進して回避する」という動きも可能である。

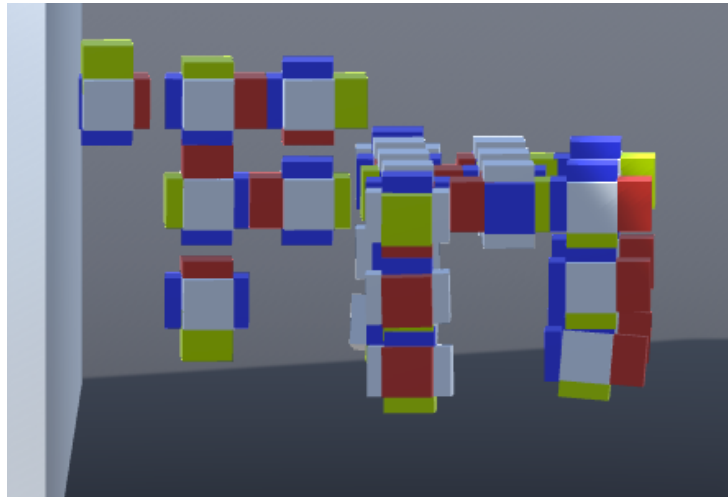


図 34: センサと障害物が衝突し，歩行が停止する様子

4.3.4 衝突したら後退する回路

センサと障害物が衝突したら後退する歩く回路を実装した(図 35,36). センサの Q に新たなリングオシレータを取り付けることで，障害物に衝突すると Q から 1 が出力され，前進用のリングオシレータは停止し，新たなリングオシレータが動作する．前進用のリングオシレータの上に重ねて後退用のリングオシレータを取り付け，後退する方向に力がかかるように 4 つの足を取り付けてある．図 35,36 では区別のために，前進用の足とリングオシレータ相当のゲートは白，後退用の足とリングオシレータ相当のゲートは紫，センサ相当のゲートは緑色で表してある．

最初は，白いゲートからなるリングオシレータが前進用の足を制御し，前進する．緑のセンサの入力 S に障害物が触れると，前進に使われていたリングオシレータが停止し，紫のリングオシレータが動き出す．紫のリングオシレータには，前進時とは逆向きに力がかかるように足が取り付けられているため，回路は後退する．

前節と同様にセンサのリセット R を回路の後方に取り付ければ，「後退中に障害物と接触したら再び前進する」という動きも可能である．

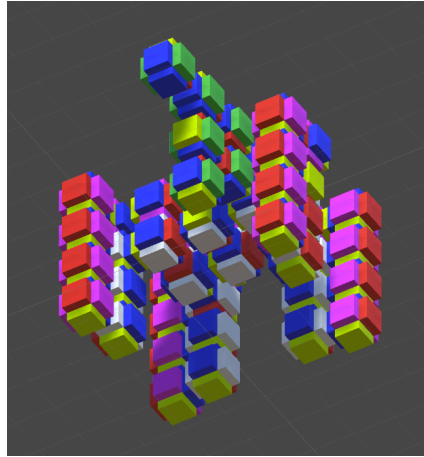
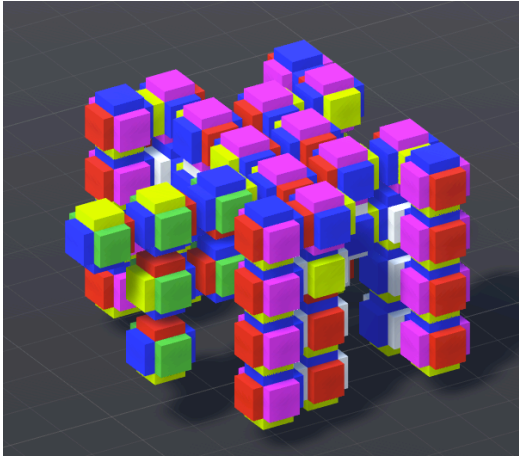


図 35: 障害物と衝突したら後退する回路. 前から見た様子 (左), 下から見た様子 (右)

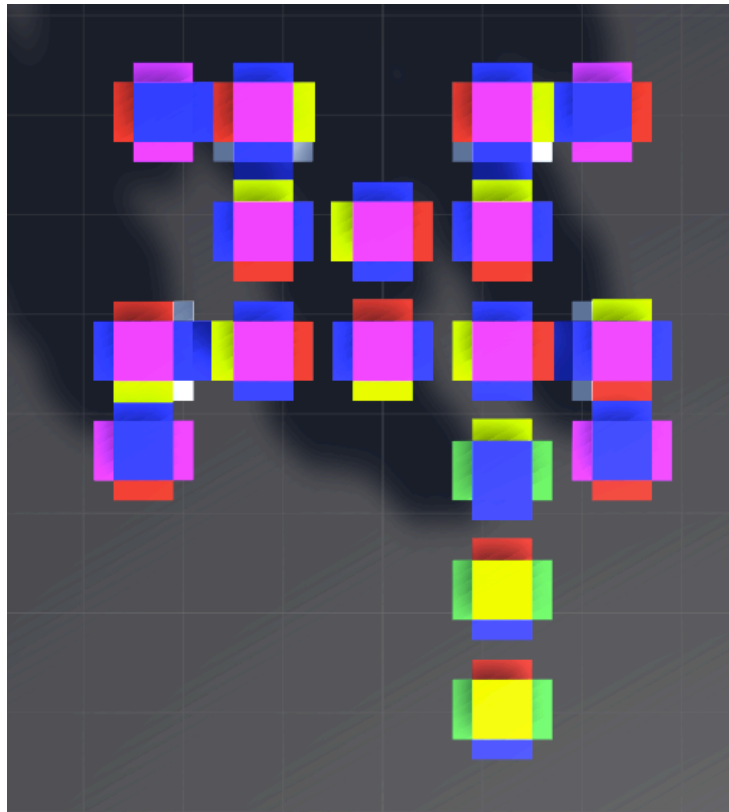


図 36: 後退する回路を上から見た様子

5 回路の自動生成

本章では，人工生命モデルとして提案するために，論理回路が多数ある仮想世界を構築し，回路同士が自然に組み上がることで新しい構造が生まれることを示す．

5.1 自然に結合と分裂を行う立方体型 NAND/AND ゲート

5.1.1 概要

論理回路同士が自然に結合したり，分裂する機能を実現するため，図 37 の立方体型 NAND/AND ゲートを実装した．これまでに実装したゲートの入出力面に，新たに磁石のように引き寄せたり引き離したりするブロックを取り付けた．出力面には赤，入力面には青のブロックが取り付けられており，これらが磁石の磁極のような機能を持っている．論理回路は入力面と出力面が結合することで動作するが，入力面同士，出力面同士では結合しても動作することができない．そのため，取り付けられた磁石は赤と青がそれぞれ N 極，S 極になることで，同極同士では反発し，N 極と S 極は引き付け合う力が発生する．これにより，入力面と出力面のみが引き付け合って結合され

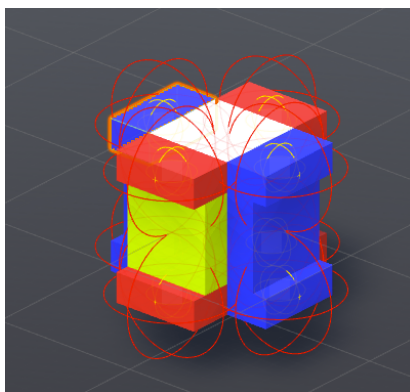


図 37: 磁石を取り付けた立方体型 NAND/AND ゲート

5.1.2 入出力面に働く力

磁力は、Unity が提供する Asset である Simple Physics Toolkit 1.4 をもとに実装した [16]. もともと Asset が提供する機能は複数のオブジェクトに対して引力、もしくは斥力のどちらかのみを与えるというものだったが、オブジェクトに付けられる tag に”N”または”S”の tag を与えることで、オブジェクト間に働く力を適切に判断して与えるように変更した. オブジェクトが増えると磁力の計算が大量になってしまうため、磁力はごく近い距離にあるオブジェクト間のみで働くように設定されている. 図 37 に表示されている赤い球体の線は *outerRadius* と呼ばれるパラメータを可視化したもので、赤い線の内側で磁力が働くことを示している. 一方で *innerRadius* というパラメータがあり、黄色い線で描画される球体の内側は磁力が働かないという距離を示している. 今回はオブジェクトの内部にちょうど収まるように *innerRadius* を設定したため、外観からは見ることはできない.

オブジェクト AB 間の中心座標までの距離を L としたときに働く磁力は、次の式で決まる.

$$\vec{F}_{magnet} = F * \vec{AB} * \frac{L - (outerRadius + innerRadius * 2)}{outerRadius}$$

各磁石オブジェクトは赤い線の内側にあるオブジェクトに対し、距離に反比例して引力、または斥力をかける. 自然界で働く磁力はクーロンの法則より、磁気量や透磁率などの計算が必要であるが、今回は簡易的に定数 F と距離に反比例した力がかかるようになっている. 斥力の場合も同様に計算し、力の方向を逆にすることで実装している.

5.2 回路同士が自然に組み上がる環境

磁石を取り付けた回路を多数用意し、回路同士が自然に組み上がる環境を実装した. 壁に囲まれた環境を作成し、その中に立方体型ゲートを 60 個設置する (図 38).

箱の中にある回路には次の制約を設けた.

- 回路にはランダムな方向から小さな力がかけられ、箱の中を自由に移動する
- もし他の回路の入力面または出力面が接近したら、回路にかけられる力は止まり、磁力だけが働く
- 回路が引き離され磁力が働かなくなると、再び小さな力がかけられ回路は動き回る

これらの制約を設けることで、回路同士が接近したら磁力を利用して結合し、結合体に他の回路からの斥力を受けたり、壁と衝突するなどで分裂すると再び移動を始める.

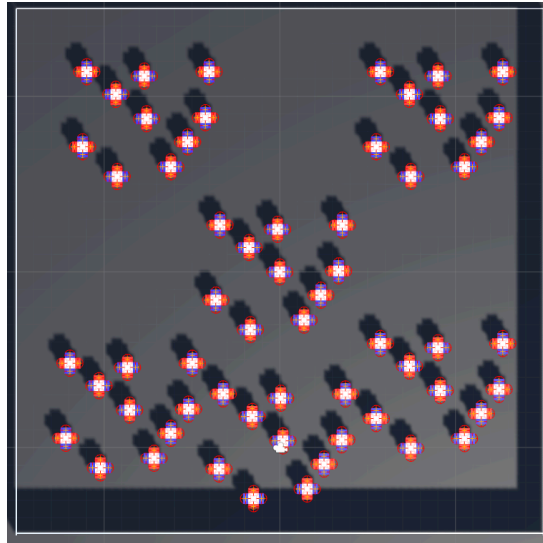


図 38: 箱の中で自由に回路が組み上がる様子

また、箱の広さと回路の配置する個数については、箱の中で多数の回路同士が接触と分裂を繰り返しやすいものを、何度か前実験することで決定した。Ingo Althöfer らによると箱はより狭く、よりたくさんのブロックを入れることで様々な構造が見られたということ参考にした [10].

5.2.1 回転と移動が自由な回路

基本の制約は前節の通りだが、少しずつ環境の条件を変えて様々な実験を行なった。まず、全ての回路は自由に回転と移動ができる条件で回路の結合を試みた。5個から10個程度からなる構造が作られたが、図39のような想定していなかった構造が多数見られた。45度程度に回転した回路同士は、2つ結合している回路の入力面と出力面にうまく結合し、結合の状態も安定しているためなかなか分離することはなく、定常状態となってしまった。回転と移動が自由な今回の条件では、入力面と出力面が正しく結合することが少なく、リングオシレータのように動き出す回路は見られなかった。

5.2.2 回転を制限した回路

回路を自由に回転させてしまうと、回路同士がうまく結合しないことが度々見られたため、回路の回転を禁止して同様の実験を行った。回路は予め0度、90度、180度、270度回転した4種類を同数ずつ用意し(図40)、無作為に配置し同様に箱の中で結合を試みた。前節と同じく5個から10個程度からなる構造が作られたが、図41のような想定し

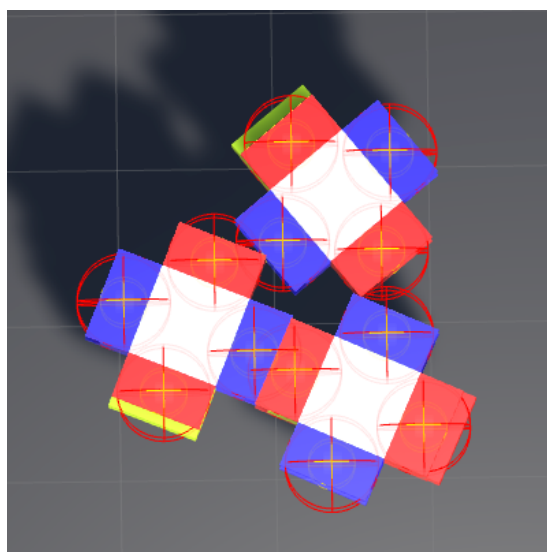


図 39: 想定外の構造で結合した回路

ていなかった構造が多数見られた。入力面と出力面が互いに並行な位置にあり、本来なら結合しない回路同士が、階段状にずれて結合し、定常状態となってしまった。

回転を制限すると、制限しなかった時と比べて入力面と出力面が正しく結合した構造が多数見られた。また図 42 のリングオシレータのような構造も見られ、実際に反転動作を繰り返す様子も観測した。

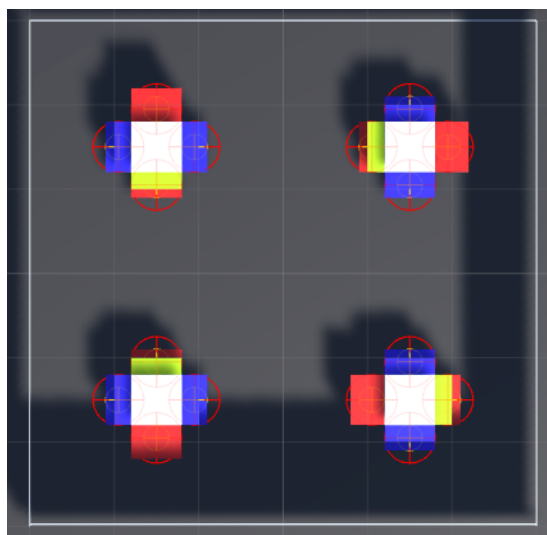


図 40: 0 度, 90 度, 180 度, 270 度回転した 4 つの回路

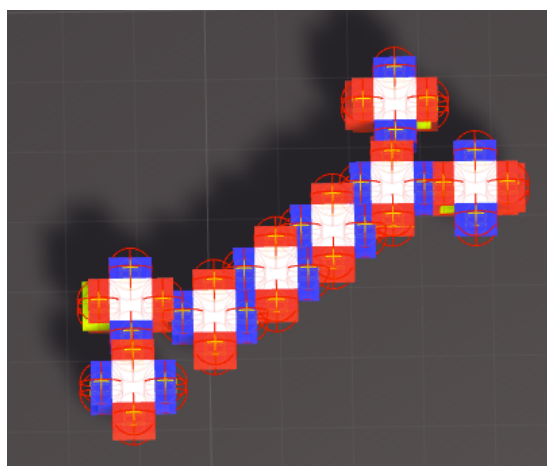


図 41: 回転を制限した環境で想定外の構造で結合した回路

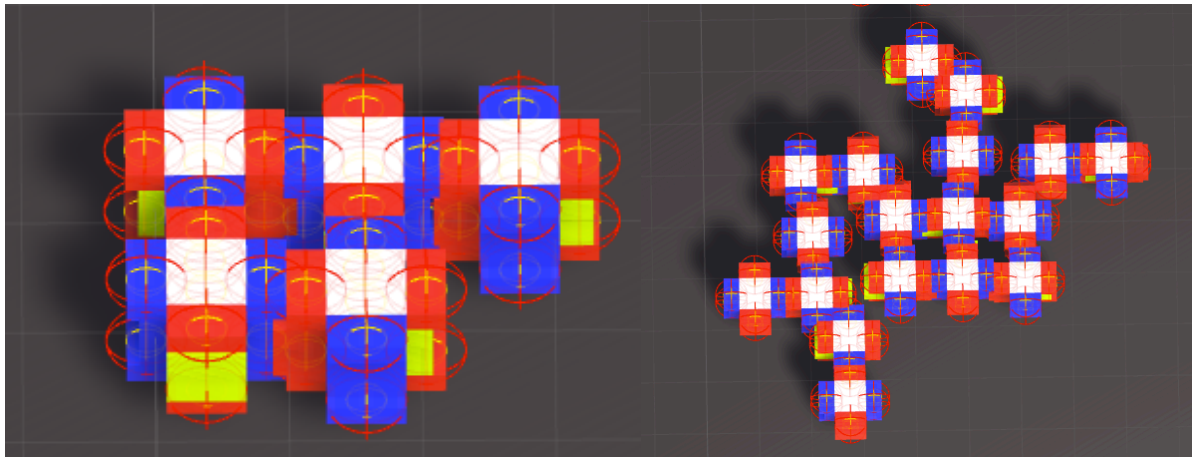


図 42: 自然に組み上がったリングオシレータのような構造

6 評価

本章では本研究で実装した論理回路と歩く回路を先行研究のものと比較し、どの程度研究目的を達成できたかを評価する。また、人工生命モデルとして提案するため開発した環境を評価する。

6.1 歩く回路の安定性について

先行研究で実装された歩く回路と本研究のものでどちらがより安定して動作しているか評価した。評価方法は、2つの歩く回路を動作させ、1歩目を踏み出すまでの時間と、不具合等で回路が停止してしまう頻度を計測した。さらに、歩き出してから自身の全長分の距離を移動するのにかかった時間も計測した。どちらの歩く回路も Unity で実装されており、Unity のバージョンや実行環境は表 3 の通りである。

6.1.1 歩く回路の実験結果

実験結果を表 4 にまとめた。先行研究の歩く回路はゲート同士が強い力でやり取りしているため、回路同士がめり込んで故障してしまったり、リングオシレータが停止してしまうことや、出力がうまく足まで伝わらないことなどが発生し、プログラムの再実行を何度も繰り返す必要があった。そのため、計測方法は Unity のプロジェクトを起動させてから回路が一步を踏み出すまでの時間を計測し、回路に不具合が生じた場合はその回数を記録し、プロジェクトを起動し直した。

先行研究の歩く回路は、本実験では全長分移動することができなかつたため、1歩進んだ距離と時間から全長分の移動時間を推測した。プロジェクト起動から1歩踏み出すのに約3分かかり、初期状態の図 43(左)から足パーツにトルクがかかり図 43(右)のような状態になった。足パーツの回転した角度が約10度でそこから1歩の距離を算出し、全長

表 3: 使用した Unity のバージョンと計算機的环境

Unity	Version 2017.3.1f1
OS	macOS 10.14.2
CPU	2.6 GHz Intel Core i5
RAM	8.0 GB
GPU	Intel Iris
VRAM	1536 MB

表 4: 1 歩の平均時間と全長分移動時間の比較

	本研究	先行研究
1 歩の平均時間 (s)	1.947	178.67
全長分移動時間 (s)	9.3689	2101.528
故障回数	0 回	18 回

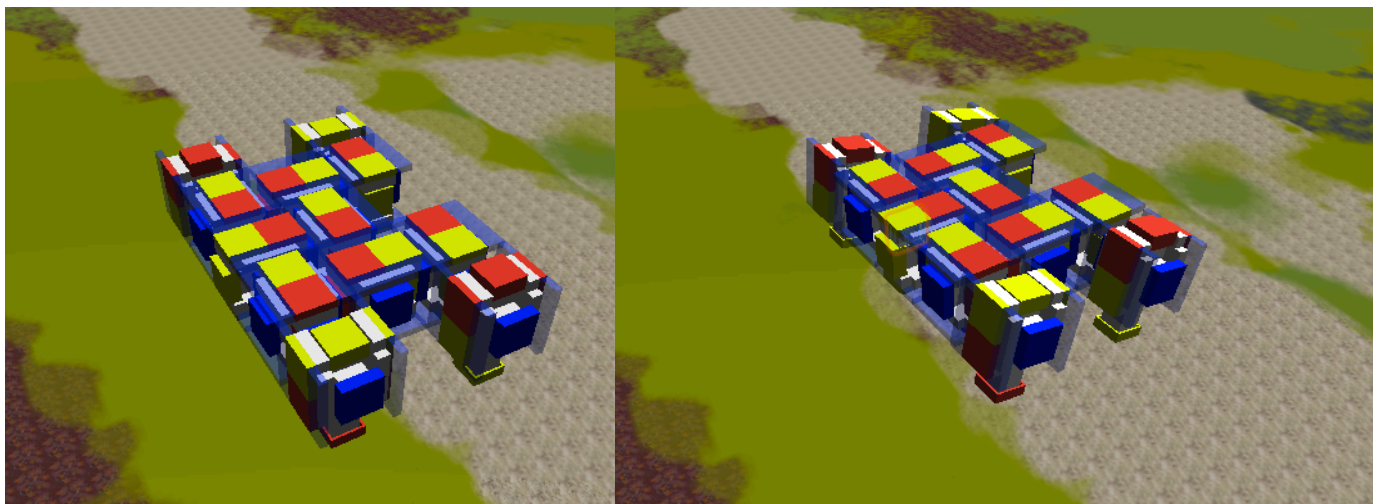


図 43: 先行研究の歩く回路の初期状態 (左) と 1 歩踏み出した様子 (右)

分の長さを 1 歩の距離で割ることで推定の全長分の移動時間を算出した。また、1 歩歩くまでに要したプログラムの再実行回数は 18 回で、うまく 1 歩を踏み出せた時にそのまま推定全長分移動時間の間観察したが、2 歩目を踏み出すことはなかった。

本研究の歩く回路は、リングオシレータは停止することなく安定して動作したため、計測は 100 歩歩くのに経過した時間から 1 歩歩く平均時間を算出した。結果は 1 歩に平均 1.947 秒となったが、100 回の計測のうち、1 回の伸縮時間が長い時は約 3 秒で、短い時は約 1.5 秒であった。また、全長分移動時間はそのまま歩く回路が自身全長分移動するのにかかる時間を測定し、結果は 9.3689 秒であった。

6.1.2 2 つの歩く回路の考察

先行研究の歩く回路は、回路が故障することや、1 歩踏み出すのに時間がかかるなどの問題があり不安定であった。回路が故障する原因は、ゲート間に働く強い力により回路がめり込んでしまうためであった。1 歩踏み出すのに大きく時間がかかるのは、ゲートが隣

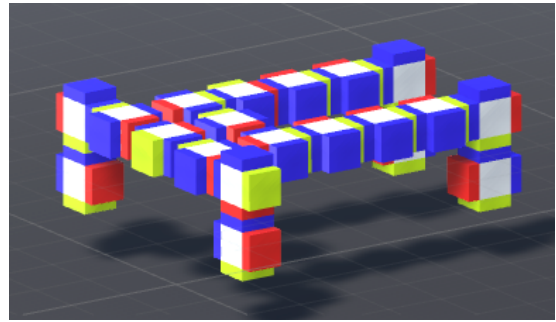
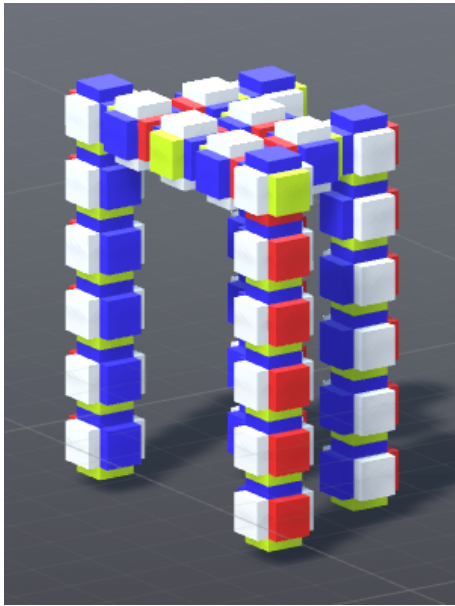


図 44: 跳ねながら移動するのっぽ型歩く回路 (左) と 1 歩の移動距離が長い早足型歩く回路 (右)

のゲートの入力面をうまく押し込むことができているからと考えられる。もし、ゲート間の伝達がスムーズに行われれば、各足はリングオシレータが 1 周期するごとに 1 回動作するはずである。先行研究ではリングオシレータの動作実験を行っており、立方体型 NAND/AND ゲートで作られたリングオシレータの平均周期は 4.55110(s) という結果が出ているため [15]、リングオシレータから足までの伝達が理想的に行われれば、もっと短い時間で足を踏み出すことができると思われる。

本研究の歩く回路は 4 つの足が停止することなく常に動作し、各足の同期が取れなくても前進することができる機構になっているため、安定して歩行することができた。ただし、本研究の歩く回路は各ゲートの質量や、Joint の強さの調整について検討する必要がある。3 段に積み重ねた足は入力を受けて伸びた際に地面を強く押すため、ゲートが軽いと足が浮かび上がってしまう。そのため、ある程度各ゲートの質量を上げる必要があるが、上げすぎてしまうとゲート間同士の Fixed Joint が壊れて分離、もしくは一時的に離れて再度結合するということが起きる。今回はいくつかのパラメータを設定し、一番うまく動作しているものを主観的に選択した。図 44 左は、同じ質量のゲートと同じ Joint の結合の強さで作った歩く回路であるが、足の長さを延長している。動作させると地面を蹴る力がとても強いため、ふわふわと月面歩行のように跳ねるように移動する。図 44 右は、後ろ足を延長しているため、1 歩の移動距離がとても長くなり、素早く移動できる。

6.2 ロボットしての機能の実現について

本研究では新たにセンサ機能を取り付けた歩く回路を実装し、前進中に障害物と衝突したら後退する回路を実現した。そこで、歩く回路が「同一の論理回路のみを使って安定して歩き回るロボット」という本研究の目的を実現できたのか考察する。動き回ることにはできているので、ロボットの構成要素であるコンピュータ、センサ、アクチュエータ、構造の4つを先行研究と比較して再現できたのか考える。

6.2.1 コンピュータ

コンピュータとは、ロボットが演算や動きの制御を行う要素で、生物の脳にあたる。リングオシレータによって、4つの足に出力を与え、順番に伸縮させることで歩くことができたので、制御ができたと言える。先行研究の歩く回路でもコンピュータ機能は実現できているが、評価結果より本研究のものの方がより安定して動作している。AIが動くような大規模なコンピュータはできていないが、最低限の再現はできたと考えられる。また、1つの構成要素はNAND/ANDゲートであるため、一般のCPUのような機能を実装することは原理的には可能である。

6.2.2 センサ

センサはロボットが外部からの情報を受け取ることができる要素で、生物の五感に相当する。先行研究の歩く回路ではセンサの要素は実現しなかったが、本研究では同一のパーツだけで状態記憶装置とセンサ機能を再現した。

6.2.3 アクチュエータ

アクチュエータはロボットが動くための要素で、生物の筋肉に相当する。先行研究では足パーツにトルクをかける事で地面を蹴って歩いたため、アクチュエータは論理回路では実現できていなかった。本研究では、論理演算によって伸縮する出力で足が動作し、地面を蹴って回路が歩くことができたので、論理回路だけでアクチュエータを実現できたと考えられる。

6.2.4 構造

ロボットが自らを支えるための要素で、生物の骨格に相当する。先行研究の歩く回路では足パーツと論理回路で実現していたが、本研究では全て単一の論理回路で歩く回路の構造を実装したため、構造の要素を単一のパーツのみで実現できたと考えられる。

6.3 人工生命モデルとしての機能について

人工生命モデルとして利用できることを提案するため、回路が自由に組み上がる環境を開発した。この環境での実験による結果から生命の定義である「自己複製」「新陳代謝」「進化できる」をどの程度満たしているのかを考察する。

6.3.1 自己複製機能について

生命は細胞の複製を繰り返すことで維持している。実装した環境では、回路は結合と分裂を繰り返し様々な構造を観測できたが、自己複製機能については観測できなかった。Ingo Althöfer らの実験のように酵素となって他や自身を狙った形に結合させる構造や、結合や分離の起こりにくい安定した構造を発見できたら自己複製に近い機能は実現できる可能性がある [10]。

6.3.2 新陳代謝について

生命は古くなった組織を次々に入れ替える。実装した環境では、組み合わせさせた回路同士の衝突や、反発する向きでの回路の接近などによって回路は分裂する。回路が分裂した箇所に新たに別の回路が結合することで、新陳代謝に近い機能は実現できたと考えられる。実際に2, 3個からなる構造が分裂し、再び同じ構造になるように結合する様子は観測できている。しかし、それ以上の個数からなる回路が分裂した箇所に、別の回路でできた同じ構造のものが再び結合する様子は観測できなかった。そのため、新陳代謝については小さな構造に関しては再現できたが、大規模なものでは再現できていない。

6.3.3 進化できるか

生物は次第に変化して、より複雑な構造を手に入れる。実装した環境では、組み合わせさせた回路からリングオシレータに相当する構造を発見できた。これは、単独で何もできなかった論理回路が組み合わせあって進化し、論理演算が可能になったと考えられる。さらに回路が組み合わせり、歩く回路のような構造が構成することができれば、ロボットの構成要素を自然に満たし、生命のように移動して活動できる。しかし、自然に組み上がった回路が移動可能な構造に組み上がることは観測できなかったため、進化については満足にはできていない。

6.3.4 人工生命モデルとして提案できるか

ここまで上げてきた。生命の定義のうち新陳代謝のみがある程度実現でき、自己複製や進化については実現できていないのが現状である。論理回路を大規模化し、進化アルゴリズムなどを実装できれば、自身で移動可能な構造を考えてパーツを構成すること

や，故障箇所を切り離し新たなパーツを取り付けることで自己複製や新陳代謝を行うことができる．また，移動可能な回路がたくさん存在し，それらが衝突した時に欠損などを受けると，環境の適合種として自身の構造を進化させることができるのではないかと考えられるが，現時点では設計できていない．

7 まとめと今後の課題

本章では本研究で実現したことをまとめ、今後解決すべき課題を述べる。

7.1 まとめ

本研究では仮想物理世界で論理回路を用いて人工生命の AI を実現するため、以下の事を行なった。

- より安定して動作する歩く回路の実現
先行研究の歩く回路はリングオシレータの出力が足に伝わらず停止するなど不安定であった。本研究では、より安定して動作する歩く回路を実現し、評価を行なった。
- センサ機能の実現
先行研究ではセンサ機能を実現していなかったが、状態記憶回路である RS ラッチを用いる事でセンサを実装した。実装したセンサを歩く回路に取り付け、前進中に障害物と接触したら後退する歩く回路を実装した。
- 同一のパーツでロボットの 4 要素全て満たした
同一の論理回路のみで、ロボットの 4 要素であるコンピュータ、センサ、アクチュエータ、構造を全て満たすことができた。
- 人工生命モデルとしての提案
磁石を取り付けた論理回路を無数に配置し、自然に結合と分裂を繰り返す環境を実装した。この環境を人工生命モデルとして利用可能であるか検討するため、生命の定義である自己複製、新陳代謝、進化できるかという 3 点から評価した。しかし、どの機能も完全に再現できていないため、人工生命モデルとして利用することはできない。

7.2 今後の課題

7.2.1 論理回路のみで AI を実現

本研究で実装した立方体型ゲートはロボットの 4 要素全て満たすことができたため、大規模化できれば、仮想環境内で AI を実装したロボットを実現できるのではないかと考えられる。

7.2.2 新たな移動可能な回路の構造発見

リングオシレータに伸縮する足を取り付けることで、歩く回路を実現したが、歩く以外にも「飛ぶ」「転がる」「這う」などの移動方法が実現できるのではないかと考えられる。

7.2.3 回路のパラメータについて

本研究で実装した機械式論理回路には、出力面の伸びる距離や速さ、回路同士の結合の強さや質量など様々なパラメータが存在する。本研究では、パラメータを様々な値に変更して実験することで、安定して歩く回路を実現したが、このパラメータが最適であったのか検討していない。機械学習などを用いて最適なパラメータを発見できれば、より安定して動作する歩く回路が実装できると考えられる。

7.2.4 人工生命モデルの実現

人工生命モデルとして利用するには、生命の基準を満たす必要がある。自己複製や進化などに必要な AI を仮想環境内で実現できれば、環境内で移動可能な構造が多数生まれ、環境内の影響を受けて様々な構造に進化していくモデルが開発できると考えられる。

参考文献

- [1] 瀧ノ上 正浩, 生体内で働く分子ロボットの実現へ: 情報媒体としての DNA 分子と DNA コンピューティング, 情報管理 60 巻 9 号 p.629-640, 2017
- [2] Unity, URL:<https://unity3d.com> (参照日 2019 年 1 月 1 日)
- [3] 赤間世紀, 『人工生命入門』, 工学社, 2010
- [4] William M. Stevens, Self-Replicating Machines, URL: <http://www.srm.org.uk/home.html> (参照日 2019 年 1 月 1 日)
- [5] Will Stevens, Self Replication # 3, YouTube. https://www.youtube.com/watch?v=PBXO_6Jn1fs (参照日 2019 年 1 月 13 日)
- [6] ロボットは自己複製できるか?/物理エンジン, YouTube. https://www.youtube.com/watch?v=ML_Wb8TFaPM&t=117s (参照日 2019 年 1 月 13 日)
- [7] ロボットは自身の構成部品を全て入れ替えることができるか?/物理エンジン, YouTube. <https://www.youtube.com/watch?v=Z8jLGhkD9nA&t=5s> (参照日 2019 年 1 月 16 日)
- [8] Jared M. Moore, Anne K. Gutmann, Craig P. McGowan, Philip K. McKinley, Exploring the Role of the Tail in Bipedal Hopping through Computational Evolution, Proceedings of the 12th European Conference on Artificial Life, 978-0-262-31050-5, pp11-18, 2013
- [9] LEGO, URL: <https://www.lego.com> (参照日 2019 年 1 月 1 日)
- [10] Ingo Althöfer, Evolution of Complexes from LEGO Bricks in a Washing Machine, Annals of Improbable Research vol. 20, no. 2, 2014
- [11] Ingo Althöfer, Random Structures from Lego Bricks and Analog Monte Carlo Procedures, Althfer2013RandomSF, 2013
- [12] Stanley L. Miller, A Production of Amino Acids Under Possible Primitive Earth Conditions, Vol. 117, Issue 3046, pp. 528-529, 1953
- [13] Ohara Masayuki, Takinoue Masahiro, Kawano Ryuji, Nanopore Logic Operation with DNA to RNA Transcription in a Droplet System, ACS Synthetic Biology 2017, vol.6, no.7, p1427-1432

- [14] 瀬戸口幸寿, 成見哲, 仮想物理世界で動く論理回路の実装, 2014 年度人工知能学会全国大会 (第 28 回), 3O1-11in, 2014
- [15] 神澤俊, 成見哲, 仮想物理世界で歩く論理回路の実現, 電気通信大学 情報・通信工学専攻 修士論文, 2015
- [16] Simple Physics Toolkit, バージョン 1.4, URL:https://assetstore.unity.com/packages/tools/physics/simple-physics-toolkit-magnet-water-wind-vr-ready-53912?aid=1100137E9&utm_source=aff