

### 動的バイナリーニューラルネットの6足歩行 ロボット制御への応用

Suzuki, Takumi / 鈴木, 匠

---

(出版者 / Publisher)

法政大学大学院理工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学研究科編

(巻 / Volume)

63

(開始ページ / Start Page)

1

(終了ページ / End Page)

4

(発行年 / Year)

2022-03-24

(URL)

<https://doi.org/10.15002/00025339>

# 動的バイナリニューラルネットの 6 足歩行ロボット制御への応用

## APPLICATION OF DYNAMIC BINARY NEURAL NETWORKS TO CONTROL OF HEXAPOD ROBOTS

鈴木 匠

Takumi Suzuki

指導教員 齋藤利通

法政大学大学院理工学研究科電気電子工学専攻修士課程

Dynamic binary neural networks are characterized by binary connection parameters and signum activation function. Depending on the parameters, the networks can generate various binary periodic orbits. Binary connection parameters and signum activation function are suitable for precise analysis and simple hardware implementation. We consider an engineering application of the dynamic binary neural networks. We synthesize networks that guarantees storage and global stability of periodic orbits corresponding to typical walking patterns of hexapod robots. Presenting an FPGA based hardware, switching of the walking patterns is confirmed experimentally.

**Key Words** : Dynamic binary neural networks, Stability, FPGA

### 1. はじめに

動的バイナリニューラルネット(DBNN[1]-[3])は 2 値結合パラメータとシグナム活性化関数によって特徴づけられる。DBNN はパラメータに依存して、さまざまな 2 値周期軌道を生成する。2 値結合パラメータとシグナム活性化関数は、簡素なハードウェア実装と精密な動作解析に適している。工学的応用として、スイッチング電源の制御信号[4][5]、歩行ロボットの制御信号[6][7]などがある。

はじめに、DBNN を導入する。DBNN の動作は 2 値状態ベクトルの差分方程式で記述される。所望の 2 値周期軌道(DBPO)の銘記を保証して、DBPO のエントランスを実現するネットワークの合成方法[2]を示す。エントランスとは DBPO の要素である。ほぼ全ての初期状態はそのエントランスを経由して、DBPO に直接遷移する。エントランスが存在するとき、その DBPO は大域安定である。この合成方法は DBPO が与えられたとき、パラメータが自動的に設定される。そして、1 つの隠れニューロンのドロップアウトがエントランスを作る。

次に、DBNN の工学的応用について考察する。この合成方法を 6 足歩行ロボットの典型的な歩行パターンに対応している 2 つの 6 次元 DBPO に適用する。2 つの DBPO は DBNN に銘記される。そして、この銘記された 2 つの DBPO は大域安定を保証するエントランスを持つ。この DBNN のパラメータを切り替えたとき、2 つの DBPO の

切り替えを実現することができる。この DBNN はロボットの 2 つの歩行パターンの銘記と切り替えを制御することができる。Verilog を用いて、DBNN を FPGA 回路に実装する。そして、歩行パターンの切り替えを実験で確認する。

### 2. 動的バイナリニューラルネットと 2 値周期軌道

動的バイナリニューラルネット(DBNN)は以下の差分方程式で記述される。

$$x_i^{t+1} = f\left(\sum_{j=1}^{N_2} C_{ij}y_j^t + S_i\right), y_j^t = f\left(\sum_{i=1}^{N_1} W_{ji}x_i^t - T_j\right) \quad (1)$$

$$f(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \quad \begin{matrix} i \in \{1, \dots, N_1\} \\ j \in \{1, \dots, N_2\} \end{matrix}$$

ただし、 $x_i^t \in \{-1, +1\}$  は離散時刻  $t$  における  $i$  番目の 2 値状態変数、 $y_j^t \in \{-1, +1\}$  は  $j$  番目の 2 値隠れ変数である。結合パラメータ ( $W_{ji} \in \{-1, +1\}$ ,  $C_{ij} \in \{-1, +1\}$ ) は 2 値、しきい値パラメータ  $S_i$  と  $T_j$  は整数値である。シグナム活性化関数  $f$  は  $-1$  または  $+1$  を出力する関数である。出力  $x_i^{t+1}$  は遅延フィードバックにより次の入力となる。よって、DBNN はさまざまな 2 値周期軌道を生成する。以下に式 (1) をベクトル形式にした式を示す。

$$\mathbf{x}^{t+1} = \mathbf{f}(\mathbf{y}^t \mathbf{C} + \mathbf{S}), \quad (\mathbf{y}^t)^\top = \mathbf{f}^\top(\mathbf{W}(\mathbf{x}^t)^\top - \mathbf{T}) \quad (2)$$

ただし,  $\mathbf{C} = (C_{ij})$ ,  $\mathbf{W} = (W_{ji})$ を結合行列,  $\mathbf{S} = (S_i)$ ,  $\mathbf{T} = (T_j)$ をしきい値ベクトルとする. また,  $\mathbf{x}^t \equiv (x_1^t, \dots, x_{N_1}^t) \in \mathbf{B}^{N_1}$ は $N_1$ 次元 2 値状態ベクトル,  $\mathbf{y}^t \equiv (y_1^t, \dots, y_{N_2}^t) \in \mathbf{B}^{N_2}$ は $N_2$ 次元 2 値隠れベクトル,  $\mathbf{f}^\top$ は $\mathbf{f}$ の転置を示す.

周期  $p$  の所望の 2 値周期軌道(DBPO)は以下のように定義される.

$$\begin{aligned} & \mathbf{a}^1, \dots, \mathbf{a}^s, \dots, \mathbf{a}^p \dots; \text{entrance} = \mathbf{a}^2 \\ & \begin{cases} \mathbf{a}^{s_1} = \mathbf{a}^{s_2} & \text{for } |s_1 - s_2| = np \\ \mathbf{a}^{s_2} \neq \mathbf{a}^{s_1} & \text{for } |s_1 - s_2| \neq np \end{cases} \end{aligned} \quad (3)$$

ただし,  $\mathbf{a}^s \equiv (a_1^s, \dots, a_{N_1}^s) \in \mathbf{B}^{N_1}$ ,  $n$ は正の整数を示す.  $\mathbf{a}^2$ を DBPO のエントランスと定義する. 全ての初期状態(DBPO の要素は除く)はエントランスを経由して, DBPO に直接遷移する. このようなエントランスが存在したとき, この DBPO は大域安定である[2]. 便宜上,  $\mathbf{a}^{s+1} \equiv \mathbf{b}^s$ ,  $s \in \{1, \dots, p\}$ と定義する. ただし,  $\mathbf{b}^s \equiv (b_1^s, \dots, b_{N_1}^s) \in \mathbf{B}^{N_1}$ である. 以下の式が成り立つパラメータ( $\mathbf{C}, \mathbf{S}, \mathbf{W}, \mathbf{T}$ )を設定できたとき, DBPO は DBNN に銘記される.

$$\mathbf{b}^s = \mathbf{f}(\mathbf{y}^s \mathbf{C} + \mathbf{S}), \mathbf{y}^s = \mathbf{f}^\top(\mathbf{W}(\mathbf{a}^s)^\top - \mathbf{T}), s \in \{1, \dots, p\} \quad (4)$$

次にネットワークの合成方法[2]を示す.

合成方法: 以下の式のようにネットワークが合成されたとき, 式(3)の DBPO は銘記されて,  $\mathbf{a}^2$ はエントランスになる. ただし, 最初の隠れニューロンはドロップアウトされている.

$$\begin{aligned} \mathbf{x}^{t+1} &= \mathbf{f}(\mathbf{y}^t \mathbf{C} + \mathbf{S}), \mathbf{y}^t = \mathbf{f}^\top(\mathbf{W}(\mathbf{x}^t)^\top - \mathbf{T}) \\ \mathbf{C} &= \begin{pmatrix} \mathbf{b}^2 \\ \vdots \\ \mathbf{b}^p \end{pmatrix} \quad \mathbf{S} = (\mathbf{1}, \dots, \mathbf{1}) \begin{pmatrix} \mathbf{b}^2 \\ \vdots \\ \mathbf{b}^p \end{pmatrix} \\ \mathbf{W} &= \begin{pmatrix} \mathbf{a}^2 \\ \vdots \\ \mathbf{a}^p \end{pmatrix} \quad \mathbf{T} = \begin{pmatrix} N_1 - 1 \\ \vdots \\ N_1 - 1 \end{pmatrix} \end{aligned} \quad (5)$$

### 3. 6 足歩行ロボットへの応用

式(5)の合成方法を使うと任意の DBPO を銘記することができる. そして, 全ての初期点(DBPO の要素 $\mathbf{a}^2, \dots, \mathbf{a}^p$ を除く)は銘記されている DBPO のエントランスに直接遷移する. この合成方法を以下の 6 周期の DBPO1 と 3 周期の DBPO2 に適用する. DBPO1 と DBPO2 は図 1 の 6 足歩行ロボットの歩行パターンに対応している.

$$\begin{aligned} \mathbf{a}_1^1 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_1^6 \\ \mathbf{a}_1^2 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_1^1 = \text{entrance} \\ \mathbf{a}_1^3 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_1^2 \\ \mathbf{a}_1^4 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_1^3 \\ \mathbf{a}_1^5 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_1^4 \\ \mathbf{a}_1^6 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_1^5 \end{aligned} \quad (6)$$

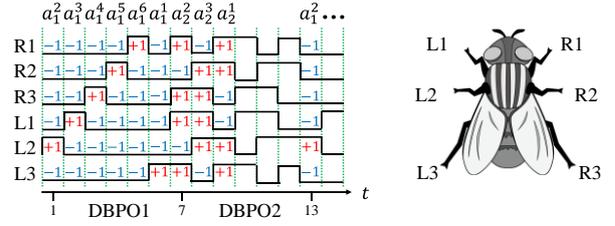


図 1 6 足歩行ロボットの歩行パターンに対応している 2 つの DBPO.

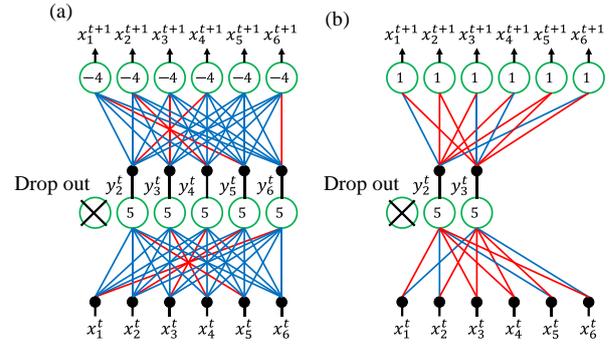


図 2 DBNN のネットワーク図. 赤線は+1, 青線は-1 をそれぞれ示す. (a)DBNN1. (b)DBNN2.

$$\begin{aligned} \mathbf{a}_2^1 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_2^3 \\ \mathbf{a}_2^2 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_2^1 = \text{entrance} \\ \mathbf{a}_2^3 &= (-1, -1, -1, -1, -1, +1) \equiv \mathbf{b}_2^2 \end{aligned} \quad (7)$$

DBPO1 を式(5)に代入すると以下の DBNN1 ができる.

$$\begin{aligned} \mathbf{x}^{t+1} &= \mathbf{f}((y_2^t, \dots, y_6^t) \mathbf{C}_1 + \mathbf{S}_1) \\ (y_2^t, \dots, y_6^t)^\top &= \mathbf{f}^\top(\mathbf{W}_1(\mathbf{x}^t)^\top - \mathbf{T}_1) \\ \mathbf{C}_1 &= \begin{pmatrix} -1 & -1 & -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & +1 \end{pmatrix} \\ \mathbf{S}_1 &= (-1, -1, -1, -1, -1, -1) \\ \mathbf{W}_1 &= \begin{pmatrix} -1 & -1 & -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & -1 & -1 & -1 & +1 \end{pmatrix} \quad \mathbf{T}_1 = \begin{pmatrix} 5 \\ 5 \\ 5 \\ 5 \\ 5 \\ 5 \end{pmatrix} \end{aligned} \quad (8)$$

図 2(a)は DBNN1 を示している. DBNN1 は DBPO1 の銘記を保証している. 最初の隠れニューロンをドロップアウトすることで,  $\mathbf{a}_1^2$ はエントランスになる. DBPO2 を式(5)に代入すると以下の DBNN2 ができる.

$$\begin{aligned} \mathbf{x}^{t+1} &= \mathbf{f}((y_2^t, y_3^t) \mathbf{C}_2 + \mathbf{S}_2) \\ (y_2^t, y_3^t)^\top &= \mathbf{f}^\top(\mathbf{W}_2(\mathbf{x}^t)^\top - \mathbf{T}_2) \\ \mathbf{C}_2 &= \begin{pmatrix} -1 & +1 & +1 & +1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 \end{pmatrix} \\ \mathbf{S}_2 &= (\mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}) \\ \mathbf{W}_2 &= \begin{pmatrix} +1 & -1 & +1 & +1 & -1 & +1 \\ -1 & +1 & +1 & +1 & +1 & -1 \end{pmatrix} \quad \mathbf{T}_2 = \begin{pmatrix} 5 \\ 5 \end{pmatrix} \end{aligned} \quad (9)$$

図 2(b) DBNN2 を示している. DBNN2 は DBPO2 の銘記

を保証している。最初の隠れニューロンをドロップアウトすることで、 $a_2^2$ はエンタランスになる。よって、DBNN1とDBNN2のパラメータを切り替えることで、図3の2つの歩行パターンの切り替えに対応しているDBPO1とDBPO2の切り替えを実現することができる。

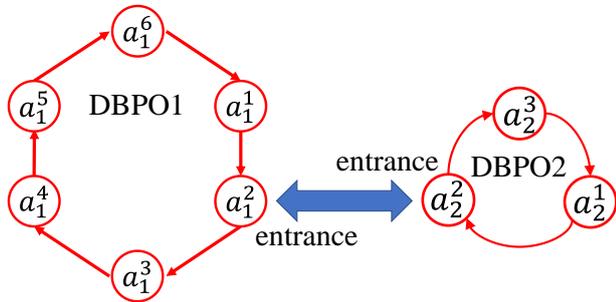


図3 DBPO1とDBPO2の切り替え。

#### 4. FPGA回路に基づくハードウェア実装

Verilogを用いて回路を設計した。図4はバイナリニューロンを実現する。バイナリニューロンはXNORゲート、加算器、比較器を用いた。図5はDBNNを実現する。ネットワークの遅延フィードバックはD型フリップフロップを用いた。また、カウンタを用いることでパラメータの切り替えを実現した。

DBPO1とDBPO2の切り替えを実験で確認した。図6(a)はDBNNが生成した信号である。 $t=7$ でDBPO1からDBPO2のエンタランスに直接遷移した。そして、 $t=13$ でDBPO2からDBPO1のエンタランスに直接遷移した。また、図6(b)はDBNNが生成した信号から変換したパルス幅変調(PWM)制御信号である。DBPOの+1と-1をパルス幅1.7msと1.5msにそれぞれ変換している。図7は6足歩行ロボットの歩行のスナップショットを示す。以下に実験で使用したツールを示す。

- Verilog version: Vivado 2018.2 platform (Xilinx).
- FPGA: BASYS3(Xilinx Artix-7 XC7A35T-ICPG236C).
- 周波数: 50 [Hz](通常の周波数 100 [MHz]を分周).
- 測定器: ANALOG DISCOVERY2.
- 測定で使用するソフトウェア: Waveforms 2015.

#### 5. むすび

本論文ではDBNNを導入して、6足歩行ロボットへの応用について考察した。任意のDBPOの銘記の保証とエンタランスを作るネットワークの合成方法を示した。この合成方法を用いて、6足歩行ロボットの歩行パターンに対応する2つのDBPOの切り替えを実現した。FPGA回路にハードウェア実装して、DBPOの切り替えを実験で確認した。また、DBPOをPWM制御信号に変換して、歩行ロボットを実現した。今後の課題は、DBPOの切り替え方法、ハードウェアの最適化、工学的応用などが挙げられる。

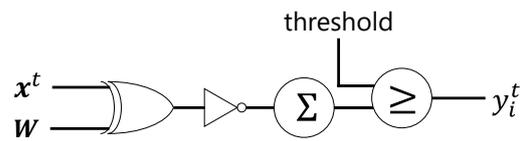


図4 バイナリニューロンの回路図。

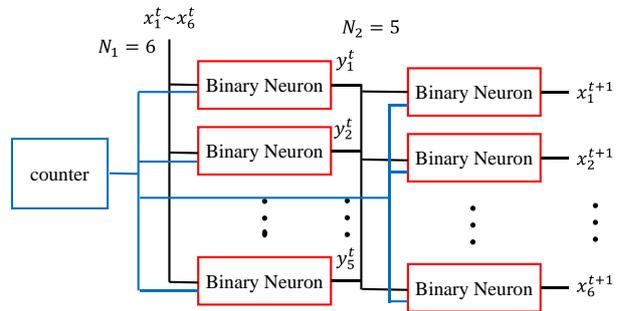


図5 DBNNの回路図。

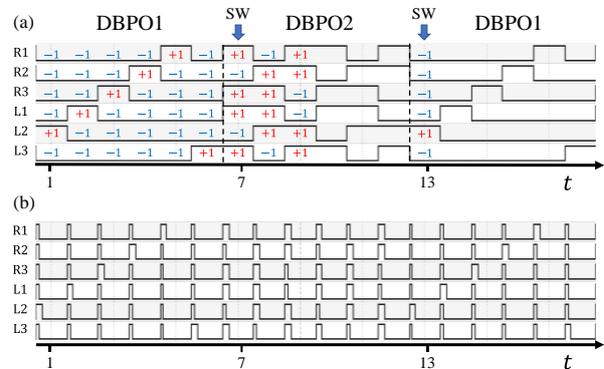


図6 測定結果。(a)DBPO1とDBPO2の切り替え。(b)PWM制御信号。

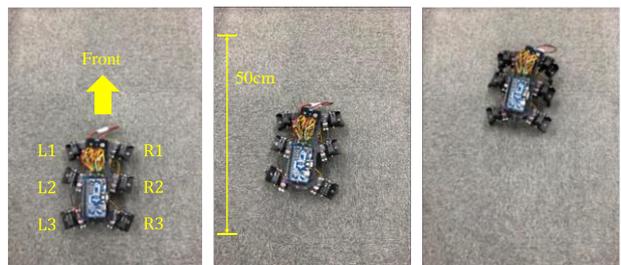


図7 6足歩行ロボット。

#### 参考文献

- 1) S. Koyama and T. Saito, Guaranteed storage and stabilization of desired binary periodic orbits in three-layer dynamic binary neural networks, *Neurocomputing*, 416, pp. 12-18, 2020.
- 2) S. Anzai, T. Suzuki and T. Saito, Dynamic binary neural networks with time-variant parameters and switching of desired periodic orbits, *Neurocomputing*, 457, pp. 357-364, 2021.
- 3) T. Suzuki and T. Saito, Synthesis of three-layer dynamic binary neural networks for control of hexapod walking robots, *Proc. IEEE/CNNA*, 2021.

- 4) B. K. Bose, Neural network applications in power electronics and motor drives - an introduction and perspective, *IEEE Trans. Ind. Electron.* 54(1), pp. 14-33, 2007.
- 5) W. Holderbaum, Application of neural network to hybrid systems with binary inputs, *IEEE Trans. Neural Netw.* 18(4), pp. 1254-1261, 2007.
- 6) P. Ramdya, R. Thandiackal, R. Cherney, T. Asselborn, A. J. Ijspeert, R. Benton, and D. Floreano, Climbing favours the tripod gait over alternative faster insect gaits, *Nat. Commun.* 8, 14494, 2017.
- 7) M. Lodi, A. L. Shilnikov and M. Storace, Design principles for central pattern generators with preset rhythms, *IEEE Trans. Neural Netw. Learn. Syst.* 31(9), pp. 3658-3669, 2020.