A developmental gene regulation network for constructing electronic circuits

Song Zhan, Julian F. Miller, Andy M. Tyrrell
Intelligent Systems Group, Department of Electronics, University of York, UK.
(sz503, jfm, amt@ohm.york.ac.uk)

Abstract. We present a method for constructing electronic circuits that uses analogues of biological multi-cellular development, genetic regulatory networks, and transcription and translation processes to build circuits. We show how small circuits may be evolved and how they may be reused to build larger circuits. We also demonstrate that the artificial 'organisms' are capable of regeneration so that circuit functionality can be recovered after damage.

Keywords: computational development, genetic regulation, digital circuit design

1 Introduction

How can we design systems with intelligent characteristics such as robustness, scalability, adaptivity and self-reconstruction? This is highly desirable for a wide variety of system designers in multiple application areas. Biological organisms that exhibit these characteristics are ubiquitous. Consequently abstracting biological processes and mechanisms has become a popular alternative to traditional design methods. However the exact mechanisms and origins of such characteristics in biology are still far from understood. The perennial question in bio-inspired research is which biology and how much biology? Recent emerging consensus from biological research points to the importance of the dynamic networks of gene activity, these have come to be known as Genetic Regulatory Networks (GRN) [2]. Such network processes are concerned with gene regulation, and protein synthesis. Macroscopically, every complex multi-cellular organism develops from the zygote cell via the mechanisms of gene regulation and cell signaling, this could generally be viewed as a building process from small to large systems. The work presented in this paper is mainly inspired by these two processes: GRNs as the underlying mechanism for a developmental system to construct electronic circuits which exhibit such underlying characteristics (robustness, scalability, adaptivity and self-reconstruction). Section 2 reviews the biological processes that have inspired the work. Section 3 reviews some related work in the application of bio-inspired computational development for circuit designs. Section 4 reviews previous work abstracting biology processes. We build on this work and present a new circuit construction idea in Section 5. Section 6 presents some analysis of initial results. Section 7 summarizes the paper and suggests some possible future work.

2 Biological processes

Biological development is a complex process involving cell division, pattern formation, morphogenesis, cell differentiation and growth. These five processes overlap to allow development to produce the most complex entities on earth [14]. Within this "high level" process genetic regulatory networks play a crucial role in the success of biological development. It is the process of regulating the DNA sequence: transcribing DNA into mRNA and then translating this information into proteins. Regulatory proteins then bind to other gene regulatory sites and regulate DNA sequence once again. These gene and protein activities build up a complex dynamic regulatory network. The components of the GRNs are genes, mRNAs and proteins and the processes include gene expression, protein transcription and translation. These processes are illustrated in figure 1[1].

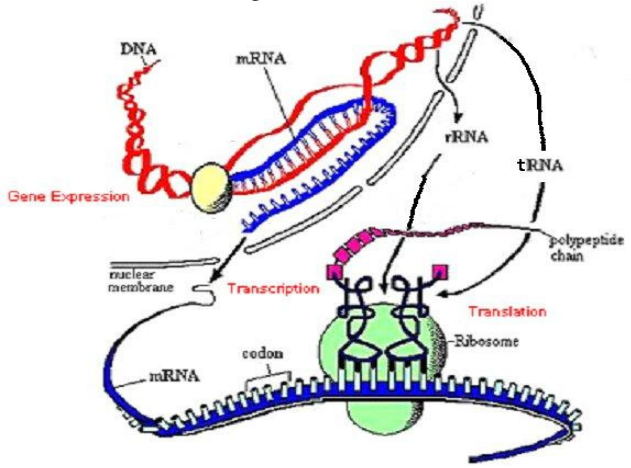


Figure 1 Protein synthesis process in Biology

2.1 Gene Regulation Network

Gene regulations dynamically generate a network structure with genes as network nodes and regulatory proteins as the connections between nodes. It is suggested that many system properties originate from this self-regulation network. In particular, cell differentiation, and homeostasis are generated by positive and negative feedback loops from two distinguished regulation mechanisms: enhance and inhibition [1][12].

[1] This diagram is modified from http://www.accessexcellence.org/RC/VL/GG/protein_synthesis.html

2.2 Protein Synthesis

Transcription proceeds in a 5' to 3' sequential carbon direction (terminal phosphate group to the terminal hydroxyl group direction) when RNA polymerase binds to a promoter sequence of a gene. It starts from a promotor site and proceeds through the coding region. Once the termination codon in DNA is met, a message RNA (mRNA) is released. As the name suggested, mRNA contains the information for the translation process. During translation, mRNA passes through the ribsome whereupon proper charged transfer RNAs (tRNAs) will match with the codons in mRNA and the amino acids at the end of tRNAs will be linked together. When the stop codon in mRNA arrives the linked amino acids, as a polypeptide, will be released.

2.3 Development

Eukaryotes develop from a zygote cell to multi cellular organism through multiple division processes. In nature, genes directly or indirectly regulate the cell division process: when a surface receptor encoded by receptor gene is combined with an external protein such as a growth factor or hormone, a signal will be transmitted into the cell cytoplasm and a series of reactions occurs that leads to cell division. When steroid or thyroid hormones are bound with the receptors in cells, the receptors could act as regulatory proteins and regulate cell division. In addition, genes which encode enzymes, and other factors to maintain division activities, could indirectly control cell division [11].

3 Related Work in developmental circuit designs

Using a developmental process for circuit design is a relatively new research field, but the promise from nature is high, and this has motivated research in the last few years to build hardware development platform and mechanisms for circuit designs. Liu et al inspired by Miller's Development CGP model [9, 10] builds up a cellular model for circuits [6]. This model demonstrates high robustness to transient faults by the dynamic cell-cell interaction processes. Gordon applies a gene regulation network mechanism to problems in evolvable hardware and investigates the scalability of developmental mechanisms for hardware design. [3]. Tufte and Haddow use an L-system with restricted cell types for actual phenotype, local knowledge (neighbouring states) and cell death, to achieve cell growth, differentiation and pattern formation on a hardware platform [13]. Koopman [5] introduces hardware-friendly GRNs for POEtic tissue [14] for fault tolerance. Mattiussi [7] inspired by biological GRNs introduces a new approach to the genetic representation and evolution of generic analog networks and presents comparable analog electronic circuit design performances with GP methods.

4. The Embryogenesis model

Our model is built up from an abstraction of the biological development processes that take DNA, to mRNA then to proteins. These are abstracted as layered processes detailed in [16]. The model contains the same genes in each cell, and through the process of gene regulations and cell signaling, the system will develop from a zygote cell to multi cellular "adult".

4.1 The Cell

Figure 2 presents an overview of our model for constructing circuits and their biological counterparts, such as, DNA, regulatory proteins, mRNA, and structure proteins. In addition it presents our abstractions of some fundamental biological processes: gene regulation, protein transcription and translation.

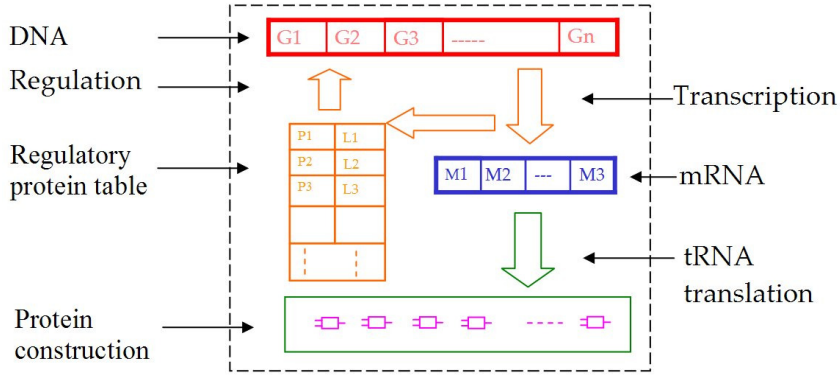


Figure 2 Circuit synthesis process and biological counterparts

A. Gene regulation network

During regulation proteins will be generated and these will regulate other genes as the top loop in figure 2 and illustrated in figure 3. This includes regulatory protein binding and product protein generation. Genes are structured with regulatory sites and product sites. Two distinguished regulatory proteins are abstracted for some potential system aspects: enhancer and inhibitor as described in section 2.1 [15]. Figure 3 is a gene regulation network constructed of three genes. The first two integers are enhancer and inhibitor sites and the last two are expression products. Regulatory proteins function transiently: after regulation they are consumed. Genes can be regulated multiple times (in Fig 3. gene 0 is regulated by gene 1 and 2 and gene 2 is regulated by gene 1 and itself).

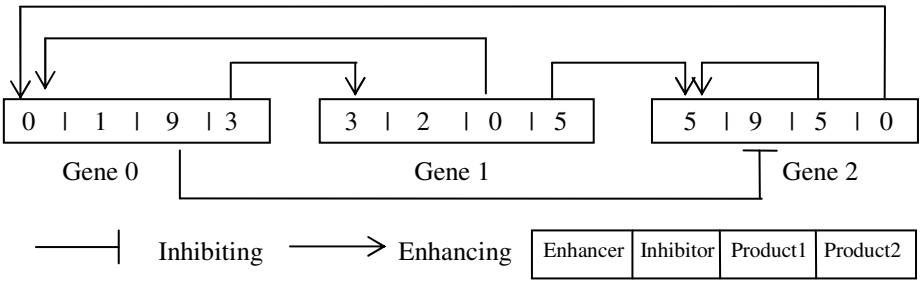


Figure 3. A Three-Gene Regulation Network

B. Transcription: processes during gene regulation cause expressed products to be released as mRNA for later translation, this is detailed in section 5.
C. Translation: processes the mRNA to the final design system detailed in section 5.

4. 2 Cell Signaling

Each cell has two processes associated with signaling: a membrane and a protein pathway. The membrane is modeled as threshold to control the coupling relationship between cells. Each protein in a cell has a cell membrane threshold that determines the minimum diffusion level for diffusion through the membrane to occur. It is implemented as a floating point number as illustrated in figure 11. The pathway is modeled to change the diffused proteins' identifications in its destination neighbor cells. One implementation example is illustrated in figure 11.

4.3 Cell Division

Cell division controls the development process. In this, the proteins in daughter cells are a proportion of proteins from the mother cells but after signaling pathway has taken place. The division process occurs during gene regulation. Some proteins are predefined to determine the division direction so when the active gene's product is the division protein, cell will have the potential to divide. Once a cell is ready to divide there has to be a location to place the daughter cell, in our model division will only occur when the division direction is empty and within a predefined environment boundary [14].

5 Circuit Constructions

We now examine how our model can be used to develop electronic circuit designs. Biological development involves the process of protein synthesis: transcription and translation. It translates the mRNA during gene regulation to the final design system. Therefore mRNA could be viewed as a link between the gene and the phenotype (application). Here we present a way to map mRNA into system structures (in this

case digital circuits) so that the final system will inherit as many of our target underlying characteristics as possible.

To construct a combinatorial circuit, there are three aspects to consider, gate functions, gate interactions and outputs. In biology proteins are constructed in ribosome by linking amino acid translated from mRNA codons. We have been particularly inspired by this process, so gates in circuits are analogous to amino acids from mRNA which themselves arise from gene regulation, through the process of transcription and translation. They are then linked together in a feed forward mechanism described by a netlist that resembles the basic structure of Cartesian Genetic Programming (CGP) [8]

5.1 Construction Processes

A. Transcription

In our model transcription process involves the processes of protein binding, gene regulation and generating regulation products as mRNA. Figure 4 shows an example of the mRNA releasing process where two regulatory proteins: enhancer and inhibitor and two potential products in each gene. In our model cells, either through division or through pathway signaling, cell can get different gene regulations and therefore different regulatory proteins. In Fig 4 if protein 6 is in the top cell and protein 8 is in bottom cell, there will be two different stable self regulations in the two cells. Gene 0, 1 and 2 regulate each other in the top cell and gene 3 and 4 regulate each other in the bottom cell. Consequently, different mRNAs, originating from the different expressed proteins are released in the two cells as seen of the right in figure 4.

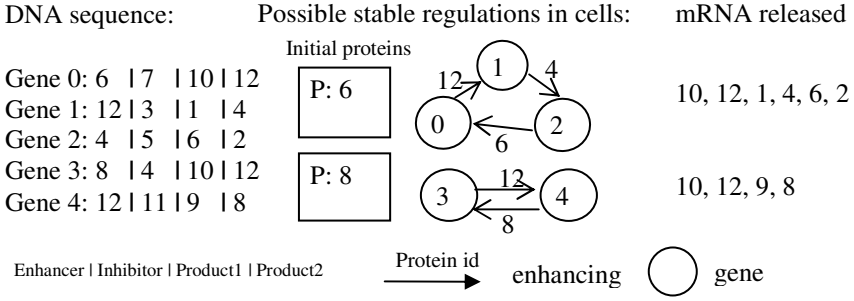


Figure 4: mRNA from DNA releasing example

B. Gate Translation

For this particular application mRNA will be translated into logic gates. To allow the system to have degeneracy within it the mathematical modulo function is chosen as the translation function. Figure 5 illustrates an example of a translation look up table from mRNA to gates where there nineteen system protein and 4 gates are used for circuit construction (hence mod 4 is used in this case). Figure 6 shows how the expressed proteins illustrated in Fig 4 are translated to gate types.

gates	Id
xor	0
or	1
nand	2
nor	3

Protein Ids	0, 1, 2, 3, ---, 17, 18

Mod (%4)

gates	protein Id
xor	0, 4, 8, 12, 16
or	1, 5, 9, 13, 17
nand	2, 6, 10, 14, 18
nor	3, 7, 11, 15

Figure 5: Gate translation example

10, 12, 1, 4, 6, 2 →2, 0, 1, 0, 2, 2 10, 12, 9, 8 →2, 0, 1, 0

mRNA → Gate id

Figure 6: Gate translation from released mRNA example from figure 4

C. Gate Connections

Currently we have chosen a simple gate construction method. Translated gates are connected together in regulation order. The new translated gate can only be connected to the gate generated before it or alternatively to the cell inputs. Potential connections for each gate are pre-allocated as shown in the example in figure 7 (square brackets) Integers are the connection points as labels in the inputs and gates in figure 8. Figure 8 are the circuits constructed during gene regulations of the example in figure 7. As in the CGP, some gates are not used for the circuits presented in gray but they contain the integer label 4 and 6 in left and 1 and 3 in right in figure 8.

DNA sequence:

Gene 0: 6 | 7 | 10 [0, 1] | 12 [0, 3]
Gene 1: 12 | 3 | 1 [0, 1] | 4 [3, 0]
Gene 2: 4 | 5 | 6 [5, 2] | 2 [7, 3]

Gene 3: 8 | 4 | 10 [0, 1] | 12 [0, 1]
Gene 4: 12 | 11 | 9 [0, 2] | 8 [4, 2]

Enhancer | Inhibitor | Product1 [Connection1] | Product2 [Connection2]

Figure 7: connection allocation for the example in figure 4

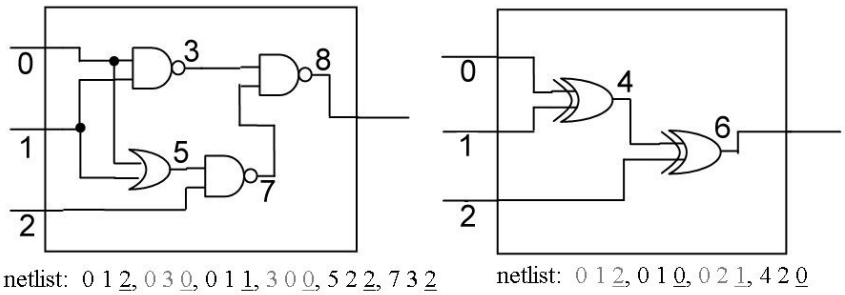


Figure 8 Connected gates in cells for example in figure 7

D. Connection reallocation

Since circuits are constructed from mRNA which is transcribed from expressed genes in DNA, in most of the cases, not all genes are expressed and the connection points for particular genes may be adjusted according to how the genes before it in the sequence regulate. The examples in figures 7 and 9 are two situations for gate connections when the number of inputs for each cell is 3. The connection in figure 7 do not need to be reallocated because genes could regulate one by one in sequence and connection points allocated all exist but in figure 9 gene 1, shown in grey, is not active (because gene 0 does not produce protein 13), the connection in gene 2 will need to be adjusted because the first translated gate from gene 2 can only be connected to either 3 inputs or the 2 gates translated from gene 0. The gate connections can be automatically adjusted so that gate connections will always be valid. An illustration of this is shown in figure 9.

DNA sequence:	DNA sequence:
0: 6 | 7 | 10 [0, 1] | 12 [0, 3]	0: 6 | 7 | 10 [0, 1] | 12 [0, 3]
1: 13 | 3 | 1 [0, 1] | 4 [3, 0]	1: 13 | 3 | 1 [0, 1] | 4 [3, 0]
2: 12 | 5 | 8 [5, 2] | 2 [7, 3]	→ 2: 12 | 5 | 8 [3, 2] | 2 [5, 3]
3: 8 | 4 | 10 [0, 1] | 12 [0, 1]	3: 8 | 4 | 10 [0, 1] | 12 [0, 1]
4: 12 | 11 | 9 [0, 2] | 8 [4, 2]	4: 12 | 11 | 9 [0, 2] | 8 [4, 2]

Enhancer | Inhibitor | Product1 [Connection1] | Product2 [Connection2]

Figure 9: Adjusting connections when genes are inactive

E. Cell Connections

Like multi-cellular biological system, our circuits are built up from cells (circuits). Each cell circuit arises through its GRN. The mechanism that we use to connect the cells is through another CGP-like netlist. Figure 10 (left) gives an example of a 1-bit adder circuit that arises from the genotype of figures 4-7. The outputs of cells are labeled using the same scheme as that employed in CGP to label gates in columns. The first two groups of three integers show where the circuits within the cells take their inputs from. The last two integers indicate where the circuit outputs are taken from. At present we assume that cell circuits only have one output. Introducing multiple outputs is a topic for future research. Many larger circuits are built up by reusing sub modules e.g., full adder circuit from half adders and simple gates. Figure 10 (right) illustrates a larger circuit built up by reusing the gene regulations in four cells. The netlist used to connect the cells and obtain the circuit outputs is given in the figure. The first four groups of three integers indicate where cells circuits connect and the last three integers indicate where the circuit outputs are taken from.

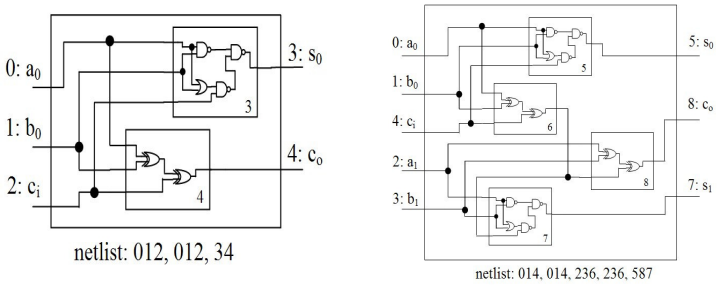


Figure 10: Example of connected cell circuits (left) and larger circuit made by connecting four single output cell circuits

6. Analysis and Results

In the example that we have discussed cell replication plays a significant role in the final circuit, but to achieve this cell differentiation is required. Here we investigate the potential of the system for producing cell differentiation and consequent reusability in the underlying mechanism of our system and show the initial results for small circuits.

6. 1 Cell Differentiation

In previous work we found positive feedback loops in gene regulations contributed to differential gene regulation in cells, in addition, signaling pathways altered the regulations in cells produced cell differentiation as well [14]. Figure 11 shows an evolved individual with 5 genes, and a total of 19 proteins. A gene is activated when the amount of enhancing protein is larger than the amount of inhibiting protein. In Figure 13 we show single columns of cells. Initially we have one zygote (at the bottom) with one protein in it which develops into a two cell organism with stabilized genetic regulation (Figure 13 A). The different shades in Figure 13 represent the presence or absence of different gene product proteins irrespective of protein amount. Gene 0, 1 and 2 are active in the top cell and gene 3 and 4 are active in bottom cell in figure 13 A.

```
/----------DNA------------/
Gene 0 : 1  | 0  | 1  | 7
Gene 1 : 14 | 12 | 11 | 5
Gene 2 : 3  | 15 | 2  | 2
Gene 3 : 0  | 11 | 0  | 15
Gene 4 : 15 | 14 | 10 | 8

/------- Zygote Protein ---------/
Protein 0 : Concentration : 40.0
```

Gene Id: Enhancer | Inhibitor | Product1 | Product2
Pathway: protein id | transferred protein id
Membrane: protein id | threshold

```
/----------Signalling----- ----------/
Pathway : 0 | 11 , 1 | 1 , 2 | 0 ,
3 | 13 , 4 | 0 , 5 | 6 , 6 | 1 ,
7 | 16 , 8 | 1 , 9 | 12 , 10 | 3 ,
11 | 4 , 12 | 3 , 13 | 11 , 14 | 7 ,
15 | 6 , 16 | 14 , 17 | 12 , 18 | 5

Membrane : 0 | 112.94 , 1 | 26.69 ,
2 | 99.57 , 3 | 27.64 , 4 | 38.10 ,
5 | 103.94 , 6 | 18.93 , 7 | 85.16,
8 | 64.53 , 9 | 7.79 , 10 | 46.67 ,
11 | 88.49 , 12 | 22.32 , 13 | 53.22 ,
14 | 18.97 , 15 | 64.24 , 16 | 106.43 ,
17 | 1.81 , 18 | 40.76 , 18 | 40.76
```

Figure 11. Evolved genotype for two cells differentiated organism

6. 2 Evolving a basic circuit

A hill climbing like search algorithm (figure 12 left) is used to search for solutions that construct a 1-bit adder in a 3*3 development environment. Figure 12 right shows the fitness versus generation for 10 different runs of 400 generations using a fixed netlist for connecting cells. In this case evolution must find GRNs that define gates, gate connections, and the signaling (membrane and pathway) to achieve the circuit. We can see some runs successfully construct a one-bit adder.

1. *Random create some individuals,*
2. *Choose the best individual*
3. *Mutate every gene one by one in sequence in each generation.*
4. *Accept the better ones immediately*
5. *Else if it gets the same fitness accept it with a certain probability*
6. *Else If it becomes worse mutate it more times with a certain probability*

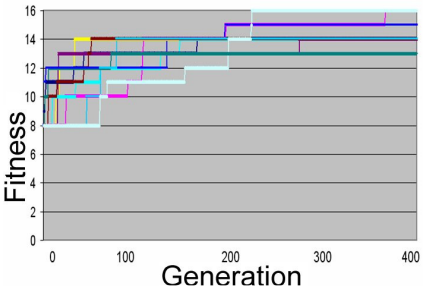


Figure 12 Evolution of 1 bit adder in 3*3 cells and the searching algorithm

6. 3 Modularity and Reusability should lead to Scalability

Modularity and reusability are two of the important aspects in allowing larger evolved systems[4]. In our system, we use gene regulations to construct circuits, therefore, the modularity of gene regulations in a small system and the ability to reuse this modularity in larger systems become important to capture system's scalability. Figure 13 shows one solution to the reuse issue. This method is an example of reuse that is applicable for many different individuals (not just this one), although not generic to every individual. We allow the cell a kind of movement behavior by allowing the daughter cells to have more area to be placed in, but only in the division direction. For example, without movement, the individual when placed in a larger environment (4 cell environment) will develop as shown in figure 13B because of the condition that cell divisions are accepted only when the division direction is empty. However, if more cell locations are given so that the daughter cells are allowed to be placed in the additional north directions, we see the original 2 cell regulation pattern (A) is repeated (figure 13 C, D & E).

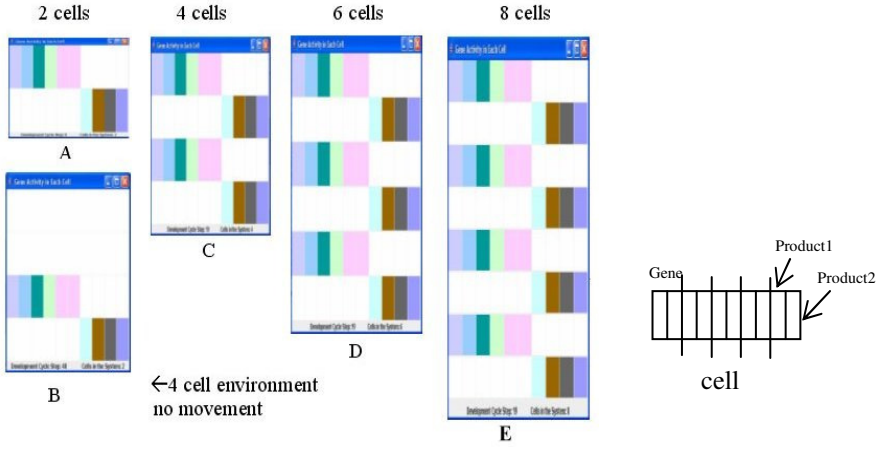
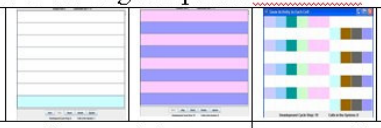


Figure 13 Modular gene regulation patterns and its reusability

6.4 Self-regulating systems can lead to self-reconstruction

A self-regulative system is generally thought of as a system that can self-construct and reconstruct after damage. This mechanism could serve as the recovery mechanism for circuits. Table 1 shows some damage response of the example individual in figure 13E with two kinds of damage: (i) changed protein and (ii) removal of cells after they develop to stabilized regulation states. It develops from a zygote cell state to its stabilized states as in the first row. The figures in the first two columns in each case use the largest protein as the cell types so that we can easily see the damage. The concentrations are the regulation patterns in the last column. Table 1 shows two kind of damage response, after cell removal and protein alteration. We find that the individual will either recover back to the original gene regulation pattern, or change to other regulation states. To achieve stable circuit construction, we suggest that degeneracy should be introduced so that some different gene regulation patterns can still represent the same circuit function.

Table 1: Damage response behaviour

Original development						
	Changing proteins			**Removing cells**		
	Change proteins	Response		Remove cells	Response	
Recovery						
Changing to other patterns of regulation						

7. Summary and Future work

A novel bio-inspired electronic circuit construction mechanism model is presented that aims to capture desirable biological characteristics such as robustness, scalability, adaptivity and self-reconstruction. Circuits are constructed using the biological development process from one cell to multi cellular by the process of gene regulation, gate translation and connection. This mimics the protein synthesis process in biology: transcription, translation and linking. We demonstrate the systems' ability to build

small circuits and investigate some emergent characteristics that lead to scalable and self-reconstruction circuits. Future research will concentrate on achieving larger circuits from evolved small circuits through re-use of modular gene regulations in many cells, and applying the system to the control of a mobile agent in a changing environment.

Acknowledgments. The first author would like to thank R. Thomson for checking the programming code.

References

1. Brenner, S., Dove, W., Herskowitz, I. & Thomas, R. (1990). Genetics 126, pp. 479-486
2. Davidson, E. (2001). Genomic Regulatory Systems: In Development and Evolution. Academic Press Inc., London, UK.
3. Gordon, T. G. W and Bentley, P. J., (2005), Development Brings Scalability to Hardware Evolution. In Proc. NASA/DoD Conference on Evolvable Hardware, pp. 272-279, IEEE Computer Society
4. Hornby, G. (2005), Measuring, Enabling and Comparing Modularity, Regularity and Hierarchy in Evolutionary Design, GECCO 2005.
5. Koopman A. (2004) Hardware-Friendly Genetic Regulatory Networks in POEtic Tissue, Master thesis.
6. Liu, H., Miller, J. F., and Tyrrell, A.M. (2004), An Intrinsic Robust Transient Fault-Tolerant Developmental Model for Digital Systems, Workshop on Regeneration and Learning in Developmental Systems, GECCO 2004.
7. Mattiussi, C. and Floreano, D. (2007) Analog Genetic Encoding for the Evolution of Circuits and Networks. IEEE Transactions on Evolutionary Computation, 11(5) pp. 596--607.
8. Miller, J. F, Thomson, P. (2000), Cartesian Genetic Programming, Third European Conference on Genetic Programming, Proceedings published as Lecture Notes in Computer Science, Vol. 1802, pp. 121-132
9. Miller, J. F. (2004), Evolving a self-repairing, self-regulating, French flag organism, Proceedings of GECCO, Part I. LNCS, 3102 Springer 2004 pp. 129-139.
10. Miller, J. F. (2003), Evolving developmental programs for adaptation, morphogenesis, and self-repair, Seventh European Conference on Artificial Life, LNAI Vol. 2801, pp. 256-265
11. Stephen L. Wolfe (1993), Molecular and Cellular Biology, Chapter 22, Wadsworth.
12. Thomas, R. (1991), Regulatory networks seen as asynchronous automata: A logical description. J. Theor. Biol., 153, 1991
13. Tufte. G. and Haddow, P. C. (2003), Building knowledge into Developmental Rules for Circuit Design, in ICES 2003, Springer-Verlag Berlin Heidelberg, P69-80.
14. Tyrrell, A.M., Sanchez, E. Floreano, D. Tempesti, G. Mange, D. Moreno, J-M. Rosenberg, J & Villa, A. (2003), POEtic Tissue: An integrated architecture for bio-inspired hardware. In Evolvable Systems: From biology to Hardware. Proceedings of ICES 2003, pp. 129-140. Springer-Verlag.
15. Wolpert, L. (1998). Principles of Development. Oxford University Press.
16. Zhan, S, Miller, F. J., Tyrrell, M. A. An Evolutionary System using Development and Artificial Genetic Regulatory Networks, IEEE Congress on Evolutionary Computation (CEC), 2008. (in press).