

Codons and codes

Kalyan Chakraborty¹, Shigeru Kanemitsu², Y. Sun³

¹Sch. of Math., Harish-Chandra Research Institute, Allahabad, India

²Grad. School of Adv. Tech., Kinki Univ., Iizuka, Japan

³Dept. of Electr. Engrg., Kyushu Inst. Techn., Tobata, Japan

Email address:

kalychak@gmail.com (K. Chakraborty)

To cite this article:

Kalyan Chakraborty, Shigeru Kanemitsu, Y. Sun. Codons and Codes. *Pure and Applied Mathematics Journal*. Special Issue: Abridging over Troubled Water---Scientific Foundation of Engineering Subjects. Vol. 4, No. 2-1, 2015, pp. 25-29. doi: 10.11648/j.pamj.s.2015040201.15

Abstract: In this paper we assemble a few ingredients that are remotely connected to each other, but governed by the rule of coding theory ([1], [12]) and formal language theory, i.e. cyclic codes and DNA codes. Our interest arose from the remark that there exist both linear and circular DNAs in higher living organisms. We state the theory of codes in a wide sense due to [1] in order to understand the circular DNAs while we state rudiments of formal language theory as a means to interpret genes. We hope this will be a starter for unifying two approaches depending on the theory of codes and that of formal language.

Keywords: Codes, Codons, Circular Codes, Linear Codes, Formal Language Theory, Regiment

1. Introduction

1.1. Cartesian Product

In Takahashi [14], Cartesian products are used to interpret and unify various phenomena effectively. In this paper, we are concerned with DNA codes as one of manifestations of the Cartesian product. We recall the passage from [3 p.44] which is also cited by Takahashi.

In relation to homeostasis, a chemo-dynamical stability of a cell in a variable environment, a description is made of the equilibrium after replication. The cell is brought out of a stable regime, to become dynamically speaking an attractor, and the stability after replication is regained in a different framework:

the dynamical features of the system reappear in the multiplicity of nearly identical cells represented by the Cartesian product of many copies of the attractor, and ensures the information by perpetuation.

Herein we interpret cells to mean the DNA's. In the remainder of this section, we assemble fundamentals on group theory for readers' convenience.

In [15] and [14], the notion of regiment is touched on, which is a nested PSO---Particle Swarm Optimization, for the purpose of balancing blocks of battery cells to obtain the optimal output power. This type of hierarchical structure may be thought of as arising from the Cartesian product, with each coordinate being a block and each block being the coordinate in (36 in the case of batteries) cells (cf. [2]).

This new concept is taken from the ideas of Schoenheimer on life to the effect that it is a dynamic state of body constituents [13], where a simile is given of a military regime and an adult body.

More elaborated, it leads to:

Life is a flow in dynamic equilibrium.

As with many other constructions, we express the relevant Cartesian product in a more convenient way according to the purposes. Thus the regiment is one for studying life activities.

1.2. *Oceanos Sapientissimus*

To formulate "replicative stability of dynamical systems" a slightly modified Carbone-Gromov suggestion [3, p.44] would be suitable: Different internal time-clocks might use dynamical time of variable fractal dimension taking into account the number of population in the species.

2. Some Plausible Speculations

2.1. DNA-Codes

Example 1. Let \mathcal{A}_4 denote the set of 4 genetic alphabets A, T, G, C (A for adenine, T for thymine, G for guanine, and C for cytosine). Then the Cartesian product \mathcal{A}_4^3 consisting of 64 elements is a genetic scheme for synthesizing amino acids. Each element written as a string, is called in a few different ways according to the disciplines: codon, trinucleotide (molecular biology), word (language theory),

coordinate (math.)

Since there are 3 stop codons, there are 61 codons which determine amino acid synthesis.

What are relevant to synthesis are A, U, G, C (U for uracil) instead of A, T, G, C, but we make an abuse of language to speak of the latter as the genetic code for synthesis. Those speculations made in [11, p. 1077] about the interpretation of RNA as a possible error-correcting code is not very persuasive in the following aspects.

- *The third base in a codon seems to have little to do with amino – acid (synthesis)* and so we may speculate that codons are error – correcting codes with the third base a check bit, whencethe codons are thought of as (3,2) – linear codes on the 4 alphabets U, A, T, G (where we changed the thymine T in DNA by uracil U in RNA).
- *The authors assume the complete probability measure, which may not be appropriate in the context of evolution (cf. [6]).*

In living cells the copying operation (transcription from DNA into mRNA) takes place before the construction phase (translation from mRNA into proteins), although in von Neumann's scheme of things these two operations are reversed. The reason is the life's strategy of saving the original design of DNA safely by making spares first. In many circumstances, we find this strategy of life to keep living on to the next generation. The problem occurs with the start codon, AUG, which corresponds to methionine and is *uniquely* works as a reading frame. There are three stop codons, UGA, UAA, UAG.

Invoking the life's strategy of preparing spares first, it is quite natural to conclude that the stop codons are in spares so that the synthesis does not go on and on. There must be one of stop codons which stop the synthesis. How about the start codon? Its uniqueness puffs away the possibility of its being a (3,2)-code. If ever the codons are codes, then they must be (n,3)-code. In their paper [11], the authors consider (12,8)-codes, too.

It is unlikely that the codons have any simple mathematical structures, *a fortiori*, a linear structure. We speculate more plausible structures including comma-free codes and free groups subsequently, Corollary 2 and §4.

2.2. (3,2)-Cyclic Code Over $\text{GF}(4)$

First we recall the field of 4 elements $\text{GF}(4)$. It is obtained as the splitting field of an irreducible polynomial of degree 2 over $\text{GF}(2) = \mathbf{F}_2 = \mathbb{Z}/2\mathbb{Z}$. Since $X^2 - 1$

$= (X - 1)^2$ is reducible over $\text{GF}(2)$, we try $\Phi(X) = X^2 + X + 1$, which is irreducible and separable over $\text{GF}(2)$.

Let $\omega \in \overline{\mathbf{F}}_2$ be a root of this polynomial, which is a primitive cube root of 1. The field

$$\text{GF}(2)(\omega) = \text{GF}(2)[X]/(X^2 + X + 1)$$

$$= \{0, 1, \omega, \omega^2\} \quad (2.1)$$

is $\text{GF}(4)$, the field of 4 elements.

E.g. the Fourier matrix

$$F = \left(\omega^{-(i-1)(j-1)} \right) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega^2 & \omega \\ 1 & \omega & \omega^2 \end{pmatrix} \quad (2.2)$$

is a generating matrix for a (3,3)-cyclic code C_ω over $\text{GF}(4)$. For basics on error-correcting codes, we refer e.g. to [12].

A canonical form of the generating matrix G for a (3,2)-linear code C over $\text{GF}(4)$ is of the form $(a_i \in \text{GF}(4))$.

$$G = (I, A) = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 & a_1 \\ \mathbf{e}_2 & a_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & a_1 \\ 0 & 1 & a_2 \end{pmatrix}. \quad (2.3)$$

Then

$$C = \left\{ \sum_{i=1}^2 \lambda_i \mathbf{c}_i \mid \lambda_i \in \text{GF}(4) \right\}, \quad (2.4)$$

$$\sum_{i=1}^2 \lambda_i \mathbf{c}_i = (\lambda_1, \lambda_2, \lambda_1 a_1 + \lambda_2 a_2).$$

Example 2. In [11] the authors denote A, T, G, C by 0, 1, ω , ω^2 , respectively.

One example is given with $a_1 = 1, a_2 = \omega$. Then for the information bit $(\omega, 0)$, the codeword is $\omega = \omega(1, 0, 1) = (\omega, 0, \omega)$.

We find all 16 codewords for each fixed a_1, a_2 . The rule is that $2a = 0$ for any $a \in \text{GF}(4)$ and $\omega^3 = 1, \omega^2 + \omega + 1 = 0$. There are 64 codes in all.

There are $4^2 = 16$ choices for $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ \omega \end{pmatrix}, \begin{pmatrix} 0 \\ \omega^2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ \omega \end{pmatrix}, \begin{pmatrix} 1 \\ \omega^2 \end{pmatrix}, \begin{pmatrix} \omega \\ 0 \end{pmatrix}, \begin{pmatrix} \omega \\ 1 \end{pmatrix}, \begin{pmatrix} \omega \\ \omega \end{pmatrix}, \begin{pmatrix} \omega \\ \omega^2 \end{pmatrix}, \begin{pmatrix} \omega^2 \\ 0 \end{pmatrix}, \begin{pmatrix} \omega^2 \\ 1 \end{pmatrix}, \begin{pmatrix} \omega^2 \\ \omega \end{pmatrix}, \begin{pmatrix} \omega^2 \\ \omega^2 \end{pmatrix}. \quad (2.5)$$

We find C for $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \omega \end{pmatrix}$. Eqn. (2.4) reads

$$(\lambda_1, \lambda_2, \lambda_1 + \lambda_2 \omega^2).$$

Substituting the values from (2.5), we obtain

$$C = \left\{ \begin{array}{l} (0, 0, 0), (0, 1, \omega^2), (0, \omega, \omega^2), (0, \omega^2, \omega), (1, 0, 1), (1, 1, \omega), \\ (1, \omega, 0), (1, \omega^2, \omega^2), (\omega, 0, \omega), (\omega, 1, 1), (\omega, \omega, \omega^2), \\ (\omega, \omega^2, 0), (\omega^2, 0, \omega^2), (\omega^2, 1, 0), (\omega^2, \omega, \omega), (\omega^2, \omega^2, 1) \end{array} \right\}. \quad (2.6)$$

3. DNA Codes

3.1. Linear and Circular DNAs

On [3, p.19], there is a brief description of circular codes. In bacteria there is typically a unique circular DNA. Besides, there is extra genomic information contained in rather short circular DNA present both in eukaryotic (in organelles) and in bacteria--prokaryotic (in plasmids).

Here eukaryotic organisms mean higher than yeast (*S. cerevisiae*) in which the genome is organized into several units called chromosomes which are separate words of DNA.

Thus considering the circular codes is meaningful and in [8]-[10] all circular trinucleotides have been identified.

3.2. Circular and Linear Codes

We confine to providing a candidate for a circular code by following the definitions and argument in [1]. The subsequent argument is a short-cut to Corollary 2 below. Materials can be found in [1].

Definition 1. A non-empty set on which there is defined a binary operation satisfying the associative law is called a semi-group. A semi-group with the identity element is called a monoid.

Let \mathcal{A} be a given non-empty set, called alphabets. We call any finite sequence a_1, a_2, \dots, a_n a word w (or a string): written $w = a_1 a_2 \dots a_n$, where we also call the void sequence a void word, written \emptyset .

Theorem 1. Let \mathcal{A}^* denote the set of all words on \mathcal{A} . On \mathcal{A}^* there is a concatenation operation, i.e. given two words $w = a_1 \dots a_n, w' = a'_1 \dots a'_m$ we concatenate them to get a new word $ww' = a_1 \dots a_n a'_1 \dots a'_m$. Then \mathcal{A}^* forms a monoid with \emptyset the identity, called the free monoid. The set of all non-empty words is denoted \mathcal{A}^+ and is referred to as the free semi-group.

Enough to check that the associative law holds true.

Definition 2. A subset $X \subset \mathcal{A}^*$ is called a code over \mathcal{A} if any word $\in X^+$ can be written uniquely as the product of words $\in X$, or has a unique factorization in words $\in X$. I.e.

$$x_1 \dots x_m = x'_1 \dots x'_n, \quad x_i, x'_i \in X \quad (3.1)$$

implies

$$m = n, \quad x_i = x'_i \quad 1 \leq i \leq m. \quad (3.2)$$

For $k \in \mathbb{N}$, the k -dimensional Cartesian product \mathcal{A}^k is called a uniform code of words of length k .

A subset $X \subset \mathcal{A}^+$ is a circular code if for all $m, n \in \mathbb{N}$, $x_i, x'_j \in X$, $1 \leq i \leq m, 1 \leq j \leq n$, and $p \in \mathcal{A}^*$ and $s \in \mathcal{A}^+$, the equalities

$$sx_2 \dots x_m p = x'_1 \dots x'_n, \quad x_1 = ps \quad (3.3)$$

imply

$$m = n, \quad p = \emptyset \quad \text{and} \quad x_i = x'_i \quad 1 \leq i \leq m. \quad (3.4)$$

By definition, a circular code is a code, but not conversely.

Indeed, given (3.1), we may view $x_1 = sp$ with $p = \emptyset$,

which is of the form of (3.3), whence (3.4) follows, amounting to (3.2)

Definition 3.

(i) For $0 \leq p, q \in \mathbb{Z}$, condition $C(p, q)$ for a sub

-monoid of \mathcal{A}^* means that for any sequence $\{u_0, u_1, \dots, u_{p+q}\}$ of words in \mathcal{A}^* , the assumptions

$$u_{i-1}u_i \in M \quad (1 \leq i \leq p+q) \quad (3.5)$$

imply $u_p \in M$.

(ii) A code X is said to be limited if there exist $0 \leq p, q \in \mathbb{Z}$ such that X is (p, q) -limited.

Definition 4.

(i) A submonoid M of \mathcal{A}^* is a free monoid if it is an isomorphic image of a free monoid, i.e. if there exists an isomorphism

$$\alpha: \mathcal{B}^* \rightarrow M \quad (3.6)$$

of a free monoid \mathcal{B}^* onto M .

(ii) Let M be a monoid and let N be its submonoid. N is stable (in M) if for all $u, v, w \in M$

$$u, v, uw, wv \in N \Rightarrow w \in N. \quad (3.7)$$

Proposition 1 ([1, Proposition 2.4, p.44]). A submonoid of \mathcal{A}^* is stable if and only if it is free.

Definition 5.

(i) A submonoid M of \mathcal{A}^* is called pure if for all $u \in \mathcal{A}^*$ and $n \in \mathbb{N}$

$$u^n \in M \Rightarrow u \in M \quad (3.8)$$

or equivalently

$$u \in M \Rightarrow u^{1/n} \in M. \quad (3.9)$$

(ii) A submonoid M of \mathcal{A}^* is called very pure if for all $u, v \in \mathcal{A}^*$

$$uv, vu \in M \Rightarrow u, v \in M. \quad (3.10)$$

Proposition 2 ([1, Proposition 2.1, p.329]). Let

$0 \leq p, q \in \mathbb{Z}$ and let M be a submonoid of \mathcal{A}^* . If M satisfies $C(p, q)$ -condition, then M is very pure.

Corollary 1. If M satisfies $C(p, q)$ -condition, then it has a basis X .

Such an X is called a (p, q) -limited code.

Proposition 3 ([1, Proposition 2.2, p.330]). Any limited code is circular.

Definition 6.

(i) We may introduce a partial ordering on \mathcal{A}^* , by the

relation being a left factor of, to the effect that $w \in \mathcal{A}^*$ is left factor of $x \in \mathcal{A}^*$ if w is a prefix of x , i.e. there exists a $u \in \mathcal{A}^*$ such that $x = wu$. The factor is said to be proper if $w \neq x$. This relation is called prefix ordering. We write $w \leq x$ if w is a left factor of x and write $w < x$ if w is a proper left factor of x . In a completely analogous way we may introduce the suffix ordering by reversing the inequality signs above.

(ii) A subset X of \mathcal{A}^* is called a prefix set if no element of X comes before any other words, i.e., if for $x, x' \in X$, $x \leq x'$ implies $x = x'$, i.e. no two elements can be comparable in the prefix ordering.

(iii) A prefix code is a prefix set which is a code different from the singleton consisting of the empty word. By changing the inequality sign we may introduce the suffix code.

(iv) A biprefix code is a code which is both prefix and suffix.

(v) The smallest $s \in \mathbb{N}$ for which

$$x \in X^s, u, v \in \mathcal{A}^*, uxv \in X^* \Rightarrow ux, xv \in X^* \quad (3.11)$$

holds is called the synchronization delay, denoted by $\sigma(X)$.

(vi) A code $X \subset \mathcal{A}^+$ is called a comma-free code if (i) X is biprefix and (ii) $s(X) = \emptyset$, i.e. X is a comma-free code if and only if

$$uxv \in X^* \Rightarrow u, v \in X^*. \quad (3.12)$$

Proposition 4 ([1, Proposition 2.9, p.336]). A code $X \subset \mathcal{A}^+$ is comma-free if and only if it is (p, q) -limited for all p, q with $p + q = 3$ and $\mathcal{A}^+ X \mathcal{A}^+ \cap X = \emptyset$.

Corollary 2. A comma-free code is circular.

Proof follows from Propositions 3 and 4.

Comma-free codes are those which are the easiest for deciphering. Indeed, if in a word, some factor can be identified in X , then this term is one factor of the unique X -factorization of that word.

In the case of codons, we have $\mathcal{A}_4 = \{A, T, G, C\}$ and \mathcal{A}_4^* is the set of all (single-stranded) DNA's. We refer e.g. to [5], where the difference between circular and linear DNA's is remarked and also that the present language theory deal with linear strings. Therefore, the codons are treated in pairs as in the following subsection.

3.3. An Example from a Language Theory

We recall the setting in §3.1, cf. Example 1. \mathcal{A}^* is a finite set, called an alphabet. In English there are $26 \approx 25 = 5^2$, in Japanese there $49 = 7^2$, and in the genetics, there are $4 = 2^2$ alphabets $\mathcal{A}_4 = \{A, T, G, C\}$. Any sequence that can be constructed using alphabets in \mathcal{A} is called a word or a string and their totality is denoted by \mathcal{A}^* . The number of alphabets contained in a word is called its length. We assume

that the length 0 string (the null string) is included in \mathcal{A}^* , denoted by \emptyset . Any subset of \mathcal{A}^* is called a language over the alphabet \mathcal{A} . In \mathcal{A}^* , the concatenation is defined: For two words x, y of length m, n we denote by xy the new word of length $m + n$ which is obtained by concatenating the left end of y to the right end of x , calling it the concatenation of x, y . For the set \mathcal{A}_4 of nucleotides, \mathcal{A}_4^* indicates the set of all possible nucleotides---primary structure of DNA.

The involution on \mathcal{A}^* is a function $f: \mathcal{A}^* \rightarrow \mathcal{A}^*$ satisfying

$$f(f(x)) = x, \quad f(xy) = f(y)f(x). \quad (3.13)$$

An involution must be a one-to-one mapping. Hence in particular, if $a \in \mathcal{A}$, then $f(a) \in \mathcal{A}$. Since $f(a_1 a_2 \cdots a_n) = f(a_n) \cdots f(a_2) f(a_1)$, it follows that the values of f are determined uniquely by those on \mathcal{A} . We also write $f(a) = a'$. Under this notation, (3.13) reads

$$x'' = x, \quad (xy)' = y'x'. \quad (3.14)$$

Since DNA is double-stranded, \mathcal{A}_4^* contains a word of the form

$$\begin{array}{l} 5' \cdots N G A A T T C N \cdots 3' \\ 4' \cdots N C T T A A G N \cdots 5' \end{array} \quad (3.15)$$

In view of the limitation of the developing state of language theory, we are to interpret this as a word in a single string. Denoting

$$\begin{array}{c} A G C T \\ T C A G \end{array}$$

by

$$[A/T], [G/C], [T/A], [C/G], \quad (3.16)$$

we see that the DNA in (3.15) reads

$$\cdots, [N/N], [G/C], [A/T], [A/T], [T/A], [T/A], [C/G], [N/N] \cdots \quad (3.17)$$

and we may apply language theory.

The DNA alphabet

$$D := \{[A/T], [G/C], [T/A], [C/G]\} \quad (3.18)$$

has the natural involution

$$\begin{array}{l} [A/T]' = [T/A], [G/C]' = [C/G], \\ [T/A]' = [A/T], [C/G]' = [G/C] \end{array} \quad (3.19)$$

4. Free Groups

We referred to codes and formal language theory for possible candidates for codons. In this section as a third

candidate, we briefly mention the notion of free groups. The case of free semi-groups goes parallel with the following. Given a family of groups $\{G_\lambda\}_{\lambda \in \Lambda}$, \mathcal{A} is the disjoint union of G_λ 's and \mathcal{A}^* is the set of all words on \mathcal{A} . \mathcal{A}^* is a monoid as above. To introduce the group structure, we define the relation $w \rightarrow w'$ if either (i) the word has successive members a, b in the same group G_λ and w' is obtained from w by replacing a, b by their product, or (ii) some members of w is an identity and w' is obtained by deleting them. For two words w, w' we write $w \equiv w'$ if there is a finite sequence $w = w_0, \dots, w_n = w'$ such that for each $j, 1 \leq j \leq n$, either $w_j \rightarrow w_{j-1}$ or $w_{j-1} \rightarrow w_j$ holds. Then we may prove that this relation is an equivalence relation and so we may construct the quotient set $G = \mathcal{A}^* / \equiv$ on which we may define the multiplication and G becomes a group, the free product of G_λ 's.

Thus, as stated in [7, p. 13], in order to multiply the word w by another word w' , we write them down in juxtaposition and carry out the necessary cancellations (multiplications in a group) and contractions (deleting identities).

On [1, p. 20, p. 56 etc.], one finds some interesting arguments on the single-stranded DNAs as words in the free group F_2 generated by two alphabets A and G with $T = A^{-1}, C = G^{-1}$. The abelianized group $F_2/[F_2, F_2]$, where the modulus is the commutator group, is isomorphic to \mathbb{Z}^2 , an infinite cyclic group and would result in excessive cancellation (hybridization). In addition to these 4 natural alphabets, there are synthesized ones including X, Y . It would be an interesting problem to find the reason why creatures use only 4 alphabets. We may need to use formal language theory developed so that it can treat both circular and linear strings to consider such a problem and we hope to return to this at another occasion.

References

- [1] J. Berstel and D. Perrin, Theory of codes, Pure Appl. Math. Vol. 117, Academic Press, London 1985.
- [2] C. Bonfiglio and W. Roessler, A cost optimized battery management system with active cell balancing for Lithium ion battery stacks, IEEE 2009.
- [3] A. Carbone and M. Gromov, A mathematical slices of molecular biology, Supplement to volume 88 of Gazette des Mathématiciens, French Math. Soc. (SMF), Paris 2001.
- [4] P. J. Davis and R. Hersh, Descartes' dream---The world according to mathematics, Harcourt Brace Jovanovich Publ., San Diego etc. 1986. Theo Hahn, International tables for crystallography, Reidel Publ. Co. 1983.
- [5] T. Head, Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors, Bull. Math. Biology 49 (1987), 737-75.
- [6] H. Kitajima and S. Kanemitsu, Math-Phys-Chem approaches to life, Intern. J. Math. Math. Sci., Volume 2012, Article ID 371825, 29 pages (doi:10.1155/2012/371825), published May 13, 2012.
- [7] A. G. Kurosh, The theory of groups I, II, Chelsea, New York 1960.
- [8] C. J. Michel and G. Pirillo, Identification of all trinucleotide circular codes coding the 20 amino acids in variant nuclear codes, Comput. Bio. Chem. 33 (2010), 122-125.
- [9] C. J. Michel, G. Pirillo and M. A. Pirillo, A classification of 20\$-trinucleotide circular codes, Inf. Comput. 212 (2012), 55-63.
- [10] C. J. Michel and G. Pirillo, A permuted set of a trinucleotide circular code coding the 20 amino acids in variant nuclear codes, J. Theor. Bio. 319 (2013), 116-121.
- [11] M. Ohya and S. Matsunaga, Coding and genes, J. Electr. Inf. Comm. Soc. J74-A (1991), 1075-1084 (in Japanese).
- [12] V. Pless, Introduction to the Theory of Error-Correcting Codes, 2nd ed., Wiley, New York etc. 1989.
- [13] R. Shoenheimer, The dynamic state of body constituents, Harvard Univ. Press. Massachusetts, 1942.
- [14] K. Takahashi, Descartes' dream: Cartesian product, Special issue "Abridging over troubled water", Pure Appl. Math. J. 2015.
- [15] Z.-Y. Zou, PSO optimization for cell-balancing charge of Lithium ion batteries, in preparation.