

Tensor-Centric Warfare II: Entropic Uncertainty Modeling

Vladimir Ivancevic¹, Darryn Reid¹, Peyam Pourbeik²

¹Joint and Operations Analysis Division, Defence Science & Technology Group, Adelaide, Australia ²Cyber and Electronic Warfare Division, Defence Science & Technology Group, Adelaide, Australia Email: Vladimir.Ivancevic@dsto.defence.gov.au, Darryn.Reid@dsto.defence.gov.au, Peyam.Pourbeik@dsto.defence.gov.au

How to cite this paper: Ivancevic, V., Reid, D. and Pourbeik, P. (2018) Tensor-Centric Warfare II: Entropic Uncertainty Modeling. Intelligent Control and Automation, 9, 30-51. https://doi.org/10.4236/ica.2018.92003

Received: April 17, 2018 Accepted: May 27, 2018

Published: May 30, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

http://creativecommons.org/licenses/by/4.0/ **Open Access**

۲

Abstract

In the first paper of the *tensor-centric warfare* (TCW) series [1], we proposed a tensor model of combat generalizing earlier Lanchester-type systems with a particular emphasis on contemporary military thinking, including the distributed C4ISR system (Command, Control, Communications, Computing, Intelligence, Surveillance and Reconnaissance). In the present paper, we extend this initial tensor combat model with entropic Lie-derivative machinery in order to capture some aspects of this deep uncertainty, while, in the process, formalizing into our model military notion of symmetry and asymmetry in warfare as a commutator, also known as a Lie bracket. In doing so, we have sought to shift the question from the prediction of outcomes of combat, upon which previous combat models such as the Lanchester-type equations have been typically constructed, towards determining the uncertainty outcomes, using a rigorous analytical basis.

Keywords

Tensor-Centric Warfare, Non-Equilibrium Entropy, Uncertainty and Symmetry of Warfare, Lie-Derivative Machinery

1. Introduction

It is generally agreed in military and defence research circles that the older strictly hierarchical manner of organization of military forces and force design decision-making oriented around considering capability in terms of collections of platforms leave something to be desired. In particular, such arrangements make effective and fluid coordination between different force elements difficult at best and impossible at worst, and leave force design susceptible to overlooking both gaps and opportunities afforded by bringing functions housed within different platforms together. Historically, however, the ability to address these limitations of hierarchical, platform-centric structures has been retarded by a variety of factors, including education and training but particularly sharp limitations on the ability to communicate in operational settings. Modern communications, information and autonomous systems technologies seem to provide a basis for finally dissolving rigid crystalline military organizations to yield more flexible forces—better combined effects from heterogeneous capabilities, ability to adapt to rapidly changing operational circumstances, and less prone to cascading failure—by enabling close coordination horizontally across traditional structural boundaries. While the topic has been the subject of much theorizing (for instance, see [2]), explanations have manifested deep methodological flaws with wide-reaching practical implications [3] [4].

These problems with determining how to utilize the emerging technology to best effect, and just with explaining what this effect is and how it comes about, may be perhaps best seen in terms of the distinctions drawn between so-called "tame" problems and "wicked" problems [5]. Military theorizing has largely followed the pattern of long-established classical system engineering-type methods, which have proven successful in dealing with tame problems but which are also known to not generalist adequately to problems that manifest incomplete and contradictory requirements, non-stationarity, complex interwoven interdependencies, unpredictability of future states, and are effectively unique.

Yet these considerations are nothing new: even the founding intellectuals of the Military Enlightenment understood war and battle as manifesting complex, messy phenomena and thus grappled with what this means for rational theory development. The two main figures of the period differed considerably in their subsequent answers. Jomini [6] established the search for a universal theory of war and battle, which has been dominant in military thinking ever since. Jomini's recognition that not everything is predictable and tame and thus not everything will fit into such a theory resulted in his splitting the domain of the complex and uncertain away from rational theory; this split between the "science" of war, dealing in certainties, from the "art" of war, dealing in the ambiguous, messy and uncertain, continues to haunt the field to the present. Clausewitz [7] denied the existence of such a fixed universal theory and emphasized instead problem solving method for dealing with what we now call wicked problems.

Against this backdrop, we developed the TCW model [1], comprising a generalized tensorial union of the classical *Lanchester-Osipov combat equations* [8] [9] [10]) and distributed C4ISR systems, towards a combat model able to capture the basic notion of collaboratively oriented and functionally individuated networked military forces, under the broad aim of capturing some of the features of uncertainty in a formal model that have proven so challenging to model in the past. In this paper, we extend this formalism with using entropic Lie-derivative to advance the representation of uncertainty in our approach, and to capture the notion of symmetry in warfare; opposing forces that are widely different in composition, strategy and operating methods are said to be asymmetrical.

2. Review of the Tensor Combat Model

The basic attrition equations include two forces, Red/Attacker's strength:

 $R = R(t) : \mathbb{R} \to \mathbb{R}$ and Blue/Defender's strength: $B = B(t) : \mathbb{R} \to \mathbb{R}$, governed by two basic types of dynamics:

1) Lanchester square law:

$$\dot{R} = -k_R B, \quad (\text{with } R(0) = R_0, k_R > 0),$$

$$\dot{B} = -k_B R, \quad (\text{with } B(0) = B_0, k_B > 0),$$
(1)

(where k_R and k_B denote individual combat-rate coefficients for the Red and Blue forces, respectively); and

2) Lanchester linear law:

$$R = -k_{BR}BR, \quad (R(0) = R_0, k_{BR} > 0),$$

$$\dot{B} = -k_{RB}RB, \quad (B(0) = B_0, k_{RB} > 0),$$

(2)

(where k_{BR} and k_{RB} denote mixed combat-rate coefficients)

In our first paper [1], we generalized and combined Lanchester Equations (1)-(2) while seeking to capture the broadly agreed theme of contemporary military thinking, which seeks to provide a highly distributed C4ISR system, where information can be readily shared and distributed.

The basic TCW development in [1] was also motivated by the recent paper by McLemore *et al.* [11], so both system inputs and outputs are comparable with inputs and outputs of the Air Campaign Scenario simulation presented in [11]. Our scaled-down 9 aircraft system input is formulated in standard fashion. Our Red forces are given by a Bipartite graph including 4 Fighter aircraft and 5 Sensor aircraft (see **Figure 1** in [1]), and our Blue forces are given by a Tripartite graph including 3 Fighter aircraft, 3 Sensor aircraft and 3 New aircraft (see



Figure 1. Sample simulation of the basic tensor combat Equation (4) for 10 time units with random initial conditions: monotonic dynamics of Blue forces.

Figure 2 in [1]).

The Red and Blue aircraft configurations are defined by the adjacency matrices (with the local feedback-loops):

				no s	elf-lo	ops							inclu	ding l	ocal se	elf-loo	ps		
	(0)	0	0	0	1	1	1	1	1)	(1	0	0	0	1	1	1	1	1)	
Red :	0	0	0	0	1	1	1	1	1	0	1	0	0	1	1	1	1	1	
	0	0	0	0	1	1	1	1	1	0	0	1	0	1	1	1	1	1	
	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	
	1	1	1	1	0	0	0	0	0 =	⇒ 1	1	1	1	1	0	0	0	0	$\equiv \mathcal{A}_{b}^{a},$
	1	1	1	1	0	0	0	0	0	1	1	1	1	0	1	0	0	0	
	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	1	0	0	
	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	1	0	
	(1	1	1	1	0	0	0	0	0)	(1	1	1	1	0	0	0	0	1)	
				no	self-lo	ops							inclu	ding	ocal s	elf-loc	ops		
	(0	0	0	no : 1	self-lo 1	ops 1	1	1	1)	(1	0	0	inclu 1	iding 1	ocal s 1	elf-loo 1	ops 1	1))
	$\begin{pmatrix} 0\\ 0 \end{pmatrix}$	0 0	0 0	no : 1 1	self-lo 1 1	ops 1 1	1 1	1 1	1 1	$\begin{pmatrix} 1\\ 0 \end{pmatrix}$	0 1	0 0	inclu 1 1	iding 1 1 1	ocal s 1 1	elf-loo 1 1	^{ops} 1 1	1) 1	
	$ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} $	0 0 0	0 0 0	no : 1 1 1	self-lo 1 1 1	ops 1 1 1	1 1 1	1 1 1	1 1 1	$ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} $	0 1 0	0 0 1	inclu 1 1	iding 1 1 1 1	ocal s 1 1 1	elf-loo 1 1 1	^{ops} 1 1	1 1 1	
	$ \left(\begin{array}{c} 0\\ 0\\ 0\\ 1 \end{array}\right) $	0 0 0 1	0 0 0 1	no : 1 1 1 0	self-lo 1 1 1 0	^{oops} 1 1 1 0	1 1 1 1	1 1 1 1	1 1 1 1	$ \left(\begin{array}{c} 1\\ 0\\ 0\\ 1 \end{array}\right) $	0 1 0 1	0 0 1 1	inclu 1 1 1 1	lding 1 1 1 0	local s 1 1 1 0	elf-loo 1 1 1 1	^{ops} 1 1 1 1	1 1 1 1	
Blue :	$ \left(\begin{array}{c} 0\\ 0\\ 0\\ 1\\ 1 \end{array}\right) $	0 0 0 1	0 0 0 1	no : 1 1 1 0 0	self-lo 1 1 1 0 0	^{rops} 1 1 1 0 0	1 1 1 1	1 1 1 1	1 1 1 1 1 1	$\Rightarrow \begin{vmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}$	0 1 0 1 1	0 0 1 1 1	inclu 1 1 1 1 0	lding 1 1 1 0 1	local s 1 1 1 0 0	elf-loo 1 1 1 1 1	^{ops} 1 1 1 1 1	1 1 1 1 1	$\equiv \mathfrak{A}_b^a$.
Blue :	$ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} $	0 0 1 1	0 0 1 1	no : 1 1 1 0 0 0	self-lo 1 1 1 0 0 0	^{nops} 1 1 1 0 0 0	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1 1 1 1	$\Rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$	0 1 0 1 1 1	0 0 1 1 1 1	inclu 1 1 1 1 0 0	1 1 1 0 1 0	local s 1 1 1 0 0 1	elf-loo 1 1 1 1 1 1	^{ops} 1 1 1 1 1 1	1) 1 1 1 1 1	$\equiv \mathfrak{A}_b^a$.
Blue :	$ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} $	0 0 1 1 1 1	0 0 1 1 1 1	no : 1 1 1 0 0 0 0 1	self-lo 1 1 1 0 0 0 1	nops 1 1 1 0 0 0 1	1 1 1 1 1 1 0	1 1 1 1 1 1 0	$ \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{array} = $	$\Rightarrow \begin{vmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{vmatrix}$	0 1 0 1 1 1 1	0 0 1 1 1 1 1	inclu 1 1 1 0 0 1	1 1 1 0 1 0 1 0	local s 1 1 1 0 0 1 1	elf-loo 1 1 1 1 1 1 1	^{ops} 1 1 1 1 1 1 1 0	1) 1 1 1 1 1 0	$\equiv \mathfrak{A}_b^a$.
Blue :	$ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} $	0 0 1 1 1 1 1	0 0 1 1 1 1 1 1	no ; 1 1 1 0 0 0 0 1 1	self-lo 1 1 1 0 0 0 1 1	^{oops} 1 1 1 0 0 0 0 1 1	1 1 1 1 1 1 0 0	1 1 1 1 1 1 0 0	$ \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{array} = $	$\Rightarrow \begin{vmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{vmatrix}$	0 1 0 1 1 1 1 1 1	0 0 1 1 1 1 1 1 1	inclu 1 1 1 0 0 1 1	ding 1 1 1 1 0 1 0 1 1	ocal s 1 1 1 0 0 1 1 1 1	elf-loc 1 1 1 1 1 1 1 1 1 0	^{ops} 1 1 1 1 1 1 1 0 1	1 1 1 1 1 1 0 0	$\equiv \mathfrak{A}_b^a$.

From the adjacency matrices \mathcal{A}_{b}^{a} and \mathfrak{A}_{b}^{a} the corresponding *combat-tensors*¹ are defined as:

Red:
$$\mathcal{N}_{b}^{a}(\mathbf{x},t) = \mathcal{A}_{b}^{a}T_{c}^{b}(\mathbf{x},t), \quad \text{Blue: } \mathfrak{N}_{b}^{a}(\mathbf{x},t) = \mathfrak{A}_{b}^{a}T_{b}^{a}(\mathbf{x},t).$$

¹A more formal definition of the combat-tensor can be given as follows. Recall that a *metric graph* (possibly with feedback loops) is a graph $\Gamma = (\mathcal{V}, \mathcal{E})$ such that:

- Each edge $e \in \mathcal{E}$ is assigned a finite *length* l_e ;
- The edge *e* is identified with the *interval* $[0, l_e]$; and
- A dynamic coordinate $x_e = x_e(t) \in [0, l_e]$ is associated with each edge $e \in \mathcal{E}$, so that $\mathbf{x}(t) = x_b^a(t) = \{x_e(t), x_{e^2}(t), \cdots\}$.

Any defence sensor, communications and information system network can be defined by the *adaptive connectivity matrix* of a metric graph:

$$C_{b}^{a} = C_{b}^{a}(\mathbf{x}, t) = \begin{cases} w_{b}^{a} x_{b}^{a}(t), & \text{vert. } a, b \text{ are adjacent,} \\ 0, & \text{otherwise.} \end{cases}$$

The weights w_b^a are trainable by Hebbian, backpropagation, or deep learning, thus making $C_b^a(\mathbf{x},t)$ into a *neural network*.

The combat-tensor is now formally defined as a linear combination of a number of connectivity matrices of various operational networks (communications, sensors and information systems, etc.):

combat-tensor:
$$\mathcal{N}_{b}^{a} = \mathcal{N}_{b}^{a} \left(\mathbf{x}, t\right) = \sum_{i=1}^{m} c_{i} \left(C_{b}^{a}\right)^{i} = c_{i} \left(C_{b}^{a}\right)^{i}$$

where the last term introduces *Einstein's summation convention over repeated indices*.



Figure 2. Sample simulation of the basic tensor combat Equation (4) for 10 time units with random initial conditions: monotonic Red-Blue phase plots.

here $T_b^a(\mathbf{x},t)$ and $T_b^a(\mathbf{x},t)$ are *Power tensors* with 9×9 components defined by the bell-shaped spatiotemporal $\operatorname{Sech}(\mathbf{x},t)$ functions, implemented in *Mathematica* as:

$$\begin{aligned} &\text{Table} \Big[\mathcal{N}_{a,b} = 0.1 \mathcal{A} \Big[[a,b] \Big] \text{Sech} \Big[2t-3 \Big] \text{RandomReal} \Big[\Big], \{a,n\}, \{b,n\} \Big], \\ &\text{Table} \Big[\mathfrak{N}_{a,b} = 0.1 \mathfrak{A} \Big[[a,b] \Big] \text{Sech} \Big[3t-4 \Big] \text{RandomReal} \Big[\Big], \{a,n\}, \{b,n\} \Big]. \end{aligned}$$

In this way constructed Red and Blue combat-tensors $\mathcal{N}_{a,b}$ and $\mathfrak{N}_{a,b}$ still retain the configuration of zeros in the adjacency matrices \mathcal{A}_b^a and \mathfrak{A}_b^a , with ones replaced by the random-weighted Sech-functions (which are here, for simplicity, only temporal, not spatial, so we remain within the ODE-framework; see discussion in subsection 4).

3. TCW Stage: Battle-Manifold with Dynamics of Vector and Tensor Fields on It

3.1. The Battle-Manifold

Both in geometrical dynamics of natural complex systems [12] [13] [14] and in modern autonomy of sociotechnical complex systems [15], a set of all active and controllable *degrees-of-freedom* (DOFs), with some additional geometric conditions, of an arbitrary complex system comprises the *configuration manifold* for ²Informally, a smooth manifold M^n is a smooth *nonlinear n*-space which is locally (at each point) equivalent to the *linear* Euclidean *n*-space \mathbb{R}^n . Formally, M^n is defined via compatible Euclidean charts:

$$\varphi_i: M^n \supset U_i \ni x \mapsto \varphi_i(x) \in V_i \subset \mathbb{R}^n,$$

for all points $x \in M^n$ and comprising a smooth atlas (for technical details, see [12] [18] and the references therein).

that system. Following this fundamental manifold prescription, any *battlespace* (see [16] and the references therein) in TCW can be formally defined as the *battle-manifold*, or its subset—the *battle-submanifold*.

Two simple examples of autonomous battle-submanifolds are (see [12]):

1) A 3D configuration manifold of a simple UGV: $M = SO(2) \times \mathbb{R}$, coordinated by $\mathbf{x} = \{x, y, \theta\}$;

2) A 4D configuration manifold of a car: $M = SO(2) \times \mathbb{R}^2$ with $\mathbf{x} = \{x, y, \theta, \phi\}$, where SO(2) is the group of rotations in the Euclidean plane \mathbb{R}^2 , coordinated by $\mathbf{x} = \{x, y\}$.

Two more complex autonomous submanifolds are (see [12]):

3) A 15D configuration manifold for an artificial hand (only flexion/extension in all three joints of all fingers) is defined as the Cartesian product: $M^{15} = \prod_{i}^{15} SO(2)^{i}$;

4) An *n*D configuration manifold for a humanoid robot is the set of all movable joint angles, defined as the Cartesian product: $M^n = \prod_i^n SO(3)^i$ of *n* rotational SO(3)-groups for all joints, where SO(3) is the group of rotations in \mathbb{R}^3 .

We give two examples of more realistic battle-submanifolds.

• A 3*n*D <u>human</u> socio-political manifold M_{SG}^{3n} , representing a generic social game played among *n* bargaining agents, according to Bueno de Mesquita's socio-political models (see [17] for a recent review), can be formally defined as:

$$M_{\rm SG}^{3n}=\prod_{i=1}^n SO(2)^i.$$

Similar configuration manifolds might be constructed for financial markets, multinational companies, political and military coalitions, and trade consortiums.

• A 6*n*D <u>robotic</u> cyber-physical-cognitive manifold M_{CPC}^{6n} , representing a dynamic configuration of a generic swarm of *cyber-physical-cognitive* robots, is defined as:

$$M_{\rm CPC}^{6n} = \prod_{i=1}^n SE(3)^i,$$

where each robot is mechanically defined as the *Euclidean group* SE(3) of rigid motions in \mathbb{R}^3 (consisting of translations and rotations in \mathbb{R}^3 ; see [15] and the references therein).

The union of a large number of similar submanifolds can form a realistic battle-manifold. In case of a very large battle-manifold M^n , it can be approximated with \mathbb{R}^n , where *n*, the total number of DOFs, can be in millions (using computational framework outlined in the **Appendix**).

Complex warfighting dynamics on such battle-manifolds is naturally defined as an interplay of spatiotemporal vector and tensor fields flowing on them (see [1]).

3.2. Warfighting Actors: Vector and Tensor Fields on the Battle-Manifolds

On any battle-manifold M^n we can observe a dynamic interplay of various

Actors, all defined by various vector and tensor fields, depending on their complexity.

Simpler Actors are formally defined as spatiotemporal vector-fields, $v^a = v^a(\mathbf{x}, t)$, similar to velocities and forces from classical mechanics, or flow-velocities and vortices from fluid mechanics, or Hamiltonian vector-fields from generalized mechanics [12] [18], or Hopfield-Grossberg vector-fields from neurodynamics [19].

The *main Actors* on any battle-manifold M^a are the *Red and Blue vector-fields*, $R^a(\mathbf{x},t)$ and $B^a(\mathbf{x},t)$, respectively, which represent either the Red-Blue populations, or any other *power measure* of the Red-Blue forces.

More complex *supporting Actors* on M^n are defined as various spatiotemporal tensor fields (*tensors*, for short): $T = T(\mathbf{x}, t) = T_{b_1 \cdots b_q}^{a_1 \cdots a_p}(\mathbf{x}, t)$ of different orders (or tensor ranks).

The supporting warfighting tensors may include:

1) 2nd-order tensors, similar to metric/inertia tensors from mechanics, or stress and strain tensors from elasticity, or stress-energy tensors from electrodynamics and gravity; other 2nd-order tensors are usually constructed directly from vector-fields, either as their outer products, or as their *gradients*: $T_b^a(\mathbf{x},t) = \nabla_b v^a(\mathbf{x},t)$.

The main supporting Actor is the combat-tensor $\mathcal{N}_b^a(\mathbf{x},t)$, defined earlier, which belongs to this category; \mathcal{N}_b^a commutes with any other 2nd-order tensor field of the same covariance on the same battle-manifold M^a (e.g.

 $T_b^a(\mathbf{x},t), S_b^a(\mathbf{x},t)$)—they can be added together as linear machines:

 $\mathcal{M}_b^a = \mathcal{N}_b^a \pm T_b^a \pm S_b^a \pm \cdots$

2) More complex, fourth-order tensors, similar to the Riemann curvature tensor, $R_{bcd}^{a} = R_{bcd}^{a}(\mathbf{x},t)$, as well as various tensors from thermo-visco-elasto-plasticity.

All these tensor fields are spatiotemporal dynamical objects. Using time derivatives and Lie derivatives of these tensor fields, we will construct various generalizations of the Lanchester combat equations.

4. Tensor Combat Equations

As already mentioned, we assume the simple 9D battle-manifold $M^9 \approx \mathbb{R}^9$, coordinated by $\mathbf{x} = (x^1, \dots, x^9)$, although all the calculations would equally work for any manifold dimension (up to millions, using the computational framework outlined in the **Appendix**). We start the TCW modeling with the *tensor Lanchester square law*, which is the following vector/tensor generalization of Equation (1):

Red:
$$\dot{R}^a = kA_b^a B^b$$
,
Blue: $\dot{B}^a = \kappa C_b^a R^b$, (3)

where the Red and Blue forces are then defined as vector-fields, $R^a = R^a(\mathbf{x},t)$ and $B^a = B^a(\mathbf{x},t)$, and their effectiveness coefficients are denoted by k and κ . The tensor fields $A_b^a = A_b^a(\mathbf{x},t)$ and $C_b^a = C_b^a(\mathbf{x},t)$ represent the sum of their combat-tensors (\mathcal{N}_b^a and \mathfrak{N}_b^a), their total power (or *stress-energy*) tensors $(S_b^a = S_b^a(\mathbf{x}, t) \text{ and } S_b^a = S_b^a(\mathbf{x}, t))$, and the Red and Blue swarming matrices, $\mathcal{R}_b^a = \mathcal{R}_b^a(\mathbf{x}, t) \text{ and } \mathcal{B}_b^a = \mathcal{B}_b^a(\mathbf{x}, t) \text{ from [11] (provided the swarming matrices have dimension of dim <math>M$):

$$\operatorname{Red} : A_b^a = \mathcal{N}_b^{a} \pm \begin{array}{c} \operatorname{R-Power} & \operatorname{R-McL} \\ S_b^a \pm \mathcal{N}_b^a \pm \begin{array}{c} S_b^a \\ \end{array} + \begin{array}{c} \mathcal{R}_b^a \\ \mathcal{R}_b^a \end{array},$$

Blue : $C_b^a = \mathfrak{N}_b^{a} \pm \begin{array}{c} \mathcal{S}_b^a \\ \mathcal{S}_b^a \\ \end{array} \pm \begin{array}{c} \mathcal{R}_b^{a} \\ \mathcal{S}_b^a \end{array}$.

All tensor calculations in this paper are derived using *Mathematica*^{*} tensor package xTensor [20]³.

Assuming, for simplicity, the coordinate independence (x = const), both sets of expanded Lanchester equations represent sets of coupled nonlinear ODEs, which can be directly numerically solved, for any given Red and Blue initial conditions: $R^a(0) = R_0^a$, $B^a(0) = B_0^a$, using any adaptive Runge-Kutta ODEsolver (e.g. Cash-Karp, Fehlberg and Dormand-Prince integrators), or their corresponding manifold/Lie-group integrators (e.g. Runge-Kutta Munthe-Kaas).

In the general case of explicit coordinate dependence ($\mathbf{x} = \mathbf{x}(t)$), we would be actually dealing with the set of the first-order nonlinear PDEs, which would all require spatial discretization (e.g., using the Method of Lines, as implemented in *Mathematica*), after which the above mentioned ODE-solvers can be used again.

The same computational algorithms will apply, in both cases (ODEs and PDEs), also for the extended tensor Lanchester equations, formulated as follows.

Next, to include the Lanchester linear law Equation (2) into Equation (3), while keeping their covariance (so that each term represents a vector-field), we need to extend them with quadratic terms of the Lanchester unaimed-fire equations (linear law) as:⁴

$$\operatorname{Red}: \dot{R}^{a} = kA_{b}^{a}B^{b} + k_{b}F_{cd}^{ab}B^{c}R^{d},$$

$$\operatorname{Blue}: \dot{B}^{a} = \kappa C_{b}^{a}R^{b} + \kappa_{b}G_{cd}^{ab}B^{c}R^{d},$$
(4)

where the fourth-order tensors F_{cd}^{ab} and G_{cd}^{ab} represent more complex, strategic, tactical and operational, Red and Blue capabilities, which can be defined either as the outer products of various matrices from [11], or composed as triple tensor sums:

$$F_{cd}^{ab} = \operatorname{strat}_{cd}^{ab} \pm \operatorname{tact}_{cd}^{ab} \pm \operatorname{oper}_{cd}^{ab},$$

$$Blue \qquad Blue \qquad B$$

For our readers' convenience, we replicate here the sample simulation (see Figures 1-3) of the tensor Lanchester combat Equation (4) performed in *Mathematica*

³Similar calculations could be performed also with other tensor packages, like *ccgrg* [21].

⁴The Basic Red and Blue tensor combat Equations (3)-(4) are valid for any linear/flat manifold M^{0} . In case of a strongly nonlinear/curved manifold M^{0} , they would need the additional connection coefficients (*i.e.*, Christoffel symbols)—which can be neglected for our purpose, as an unnecessary over-complication. With this view in mind, in the following sections we will introduce more complex dynamics and nonlinear control concepts into Equations (3)-(4), without introducing any geometric connection (e.g., Levi-Civita connection on Riemannian manifolds)—which can always be added to the battlespace system as additional nonlinear complexity (see [13]).



Figure 3. Sample simulation of the basic tensor combat Equation (4) for 10 time units with random initial conditions: monotonic dynamics of Red forces.

for 10 time units (to match the simulations given in [11]) and random initial conditions.

Interpretation of this simulation and comparison with the results reported in [11] are given in [1]. Briefly, Red (bipartite) is loosing (**Figure 3**), Blue (tripartite) is winning (**Figure 1**), and all individual conflicts are concentrated in the left-upper corner of the phase-plane (**Figure 2**).

5. Modeling *Uncertainty* and *Symmetry of* Warfare: From Clausewitz to Entropic Lie Derivative

In this section, we propose a Clausewitz-type warfare uncertainty principle at work [7], which is, in spirit, akin to the fundamental Heisenberg uncertainty principle in quantum mechanics. Unlike Heisenberg, Clausewitz did not have the mathematical tools available at the time, or even the language to describe the full warfare complexity at times, but he definitely grappled with uncertainty in war and battle as his most fundamental core intellectual construct. The nature of this uncertainty amounts to the consequences of Gödels incompleteness phenomena in mathematical logic and its real-world applications, meaning that it is due to the presence of (often hidden or disguised) logical paradoxes. Mathematically, the hidden nature takes the form of fixed-point solutions in axiomatic formulations, but it is not difficult to relate them directly with paradoxes long studied in philosophy. It happens whenever we have the potential for self-reference, which is basically any time we don't assume a mathematically "trivial" system. In our view, both Clausewitz- and Heisenbergtype uncertainties are the outcome of a set of Gödelian paradoxes, arising in the theory because of the built-in possibility of self-reference: one effects the result of the observation by the making of the observation because it is set up from the start with the notion that one is observing from within the system, not peering into it from outside. This notion of hidden self-reference in quantum mechanics

is also quite close to the way in which Clausewitz conceived of his uncertainty. The rest of Clausewitz's work is about how to both deal with it on the one hand in terms of avoiding exposure to the worst kind of failures one does not want to happen, and how to take advantage of it through the opportunities it proves, on the other.

Regarding the *warfare symmetry*, we remark that the scalar Lanchester models, (1)-(2), are *symmetric*, *i.e.*, automorphic under a swap of R and B variables⁵. In our general warfare modeling, it is important to properly address the warfare symmetry issue, which we present in this section.

5.1. Entropic Lie-Derivative Machinery

The Lie derivative operator, denoted by \mathfrak{L}_{ν} is the simplest tensor derivative (see **Appendix** for technical details) on any smooth manifold *M*, which does not require connection (or, any other geometric structure) on the manifold (see [12] [18] and the references therein). \mathfrak{L}_{ν} generalizes two familiar cases from mechanics and nonlinear control: the *directional derivative* of a scalar function (*i.e.*, scalar field) along a vector-field and the *Lie-bracket commutator* of two vector-fields⁶.

For an arbitrary geometric/physical object, T, or specifically, any (q, p)-tensor field $T = T_{b_1 \cdots b_q}^{a_1 \cdots a_p} \in M$, its Lie derivative, $\mathcal{L}_v T$, along the $flow^7$ of a certain vector-field $v^a \in M$, measures in what degree the geometric object T moves along with the flow of v^a (*i.e.*, allows for the "Lie dragging" along the flow of v^a), or resists to move with the flow of v^a (*i.e.*, resists the Lie dragging).

This notion of *Lie dragging with/without resistance* represents the *core* concept for our *modeling of complex warfighting under uncertainty*. Physically speaking, it is related to *Prigogine's entropy S* of any open non-equilibrium thermodynamical system, which considers irreversible systems/processes that exchange both energy and matter with their environment. Prigogine's Extended Second Law of Thermodynamics (see [24] [25]) formally reads:

$$\partial_t S \ge 0 \quad \text{for } \partial_t S = \frac{\mathbf{d}_i S}{\mathbf{d}t} + \frac{\mathbf{d}_e S}{\mathbf{d}t},$$

where $d_i S$ is the internal entropy production within the system, and $d_e S$ (=0 for an isolated system) is the external entropy flux (due to the exchanges with ⁵For example, the scalar Lanchester models (1)-(2) can be seen as special cases of the *non-symmetric* Lotka-Volterra predator-prey model (see [22]):

$$R = k_{R}R - k_{BR}RB, \quad (\text{with } R(0) = R_{0}, k_{R}, k_{BR} > 0),$$

$$\dot{B} = -k_{B}B + k_{RB}RB, \quad (\text{with } B(0) = B_{0}, k_{B}, k_{RB} > 0).$$

⁶For a popular visualization of the Lie derivative, see Figs. 14.13-14.16 in [23].

⁷From the fluid-dynamics perspective (where \mathcal{L}_{v} is called the Liouville operator), a vector-field v^{a} is an *infinitesimal generator* of its flow ϕ_{i} and the flow ϕ_{i} is an *integral curve* of its vector-field v^{a} . From the dynamical-systems perspective, the vector-field (e.g., Hamiltonian vector-field) is defined by the set of ordinary differential equations (e.g., Hamiltonian equations of motion), and its flow is the solution for some initial conditions. The existence of the flow for some vector-field is insured by the existence theorems for differential equations. See next footnote for additional technical details.

the environment).

The non-equilibrium thermodynamics has the following Lie-derivative analogy:⁸

Lie dragging with resistance: $\mathfrak{L}_{v}T > 0 \approx \partial_{t}S > 0$: growth of entropy; and

Lie dragging with nonresistance: $\mathcal{L}_{v}T = 0 \approx \partial_{t}S = 0$: conservation of entropy, from which we can infer that the *warfare uncertainty*, as measured by the *non-equilibrium entropy*, grows with resistance; when resistance stops, the warfare uncertainty stays constant.

More precisely, this *Lie-entropy analogy* can be geometrically interpreted as: whenever we have resistance of the opposite side, we are dealing with a positive Lie derivative of the associated geometric object⁹, *i.e.*, $\mathcal{L}_v T > 0$, which means that the non-equilibrium entropy of the underlying open system is growing, therefore the warfare uncertainty is growing. Otherwise, when the Lie derivative is zero, *i.e.*, $\mathcal{L}_v T = 0$, that means the underlying entropy is conserved and the warfare uncertainty remains constant.

As an introductory example, consider an abstract warfare scenario where some sociopolitical force, represented by the vector-field *A*, is *dragging* (either physically, or psychosocially as a "threat") another sociopolitical structure of arbitrary complexity, represented by the general tensor field *B* (which can be another force, or another sociopolitical organization, etc.). This scenario is formally written as the Lie derivative $\mathfrak{L}_A B$. So long as *B* is resisting, we have $\mathfrak{L}_A B > 0$, and correspondingly: $\partial_t S > 0$. When resistance stops, it becomes: $\mathfrak{L}_A B = 0$, and correspondingly: $\partial_t S = 0$.

5.2. Measuring Warfare Symmetry: The [Red, Blue]-Commutator

An important particular case of the general Lie-derivative, $\mathfrak{L}_{v}T$, is the *commutator*, $\mathfrak{L}_{R}B^{a} = [\text{Red}, \text{Blue}]$, also called the Lie-bracket, between the Red vector-field $R^{a}(\mathbf{x},t)$ and the Blue vector-field $B^{a}(\mathbf{x},t)$, which determines the global *warfare symmetry*.

The *Lie-bracket commutator* is defined by the antisymmetric bracket [Red, Blue] = -[Blue, Red], which measures the *degree of non-commutativity* of the Red-Blue *composed flows*:

1) $[\text{Red}, \text{Blue}] \equiv R^a \circ B^a$: flowing first along the Red flow (in which Blue is

⁸Strictly speaking, these relations are formally valid only for positive-definite tensors (*i.e.*, when the tensor expression $\mathcal{L}_v T$ is positive definite). In general, they hold for their corresponding absolute values:

Lie dragging with resistance: $|\mathcal{L}_v T| > 0 \approx \partial_t S > 0$: growth of entropy; and

Lie dragging with nonresistance: $|\mathfrak{L}_{v}T| = 0 \approx \partial_{v}S = 0$: conservation of entropy.

In other words, $\mathfrak{L}_{v}T \neq 0 \Rightarrow \partial_{t}S > 0$; $\mathfrak{L}_{v}T = 0 \Rightarrow \partial_{t}S = 0$. Also, our "Lie dragging with nonresistanc" is in geometrical literature usually called the "Lie transport."

⁹Such a thermodynamics analogy is not unique for our approach. In the specific framework of Riemannian geometry, there is the so-called *Ruppeiner metric*, defined as a (negative) Hessian of the entropy function [26]. Since we are working in a more abstract, metric-free framework, such a metric is not required. However, if our future work requires specializing to a *Riemannian battle-manifold*, its metric will naturally be the Ruppeiner metric. dragged by Red) and then along the Blue flow (in which Red is dragged by Blue), versus

2) $[Blue, Red] \equiv B^a \circ R^a$: flowing first along the Blue flow (in which Red is dragged by Blue) and then along the Red flow (in which Blue is dragged by Red). The commutator is formally defined as:

$$\begin{bmatrix} \operatorname{Red}, \operatorname{Blue} \end{bmatrix} = \begin{bmatrix} R^a, B^a \end{bmatrix} = \mathfrak{L}_R B^a = -\mathfrak{L}_B R^a = R^a \nabla_a B^b - B^a \nabla_a R^b$$
$$= -\begin{bmatrix} B^a, R^a \end{bmatrix} = -\begin{bmatrix} \operatorname{Blue}, \operatorname{Red} \end{bmatrix},$$
$$\mathfrak{L}_R B^a \neq 0 \Longrightarrow \partial_t S > 0.$$

In case the Red and Blue vector-fields *do commute*, their Lie-bracket commutator is *zero* and the corresponding *entropy* is conserved:

$$\begin{bmatrix} \text{Red}, \text{Blue} \end{bmatrix} = 0 \Leftrightarrow \begin{bmatrix} R^a, B^a \end{bmatrix} = 0, \text{ or}$$
$$R^a \circ B^a = B^a \circ R^a \Leftrightarrow \mathfrak{L}_R B^a = \mathfrak{L}_B R^a = 0,$$
$$\mathfrak{L}_R B^a = 0 \Longrightarrow \partial_t S = 0.$$

In addition, the [Red, Blue] -commutator is associative:

 $\left[\left[\text{Red}, \text{Blue} \right], \text{Red} \right] = \left[\text{Red}, \left[\text{Blue}, \text{Red} \right] \right] \text{ and}$ $\left[\left[\text{Blue}, \text{Red} \right], \text{Blue} \right] = \left[\text{Blue}, \left[\text{Red}, \text{Blue} \right] \right].$

Its straightforward interpretation is that, all other quantities being equal, at each phase of warfighting: *whoever strikes first has the advantage*, and the opponent is left to play the catch-up game. Therefore, the [Red, Blue]-commutator effectively decides who wins and who loses at each phase of warfighting.

More generally, the [Red, Blue]-commutator defines the *Lie algebra* of all possible vector-fields on the battle-manifold *M*, which satisfies the (circular) *Jacobi identity*:

$$\left[R^{a},\left[B^{a},E^{a}\right]\right]+\left[E^{a},\left[R^{a},B^{a}\right]\right]+\left[B^{a},\left[E^{a},R^{a}\right]\right]=0$$

where $E^{a} = E^{a}(\mathbf{x},t)$ is an independent third-party vector-field (like e.g., *United Nations*).

The above linear and quadratic tensor Lanchester Equations (4) are extended with the [Red, Blue]-commutators as:

$$\operatorname{Red}: \dot{R}^{a} = kA_{b}^{a}B^{b} + k_{b}F_{cd}^{ab}B^{c}R^{d} + \left[R^{a}, B^{a}\right],$$

Blue: $\dot{B}^{a} = \kappa C_{b}^{a}R^{b} + \kappa_{b}G_{cd}^{ab}B^{c}R^{d} + \left[B^{a}, R^{a}\right],$
(6)

which address the warfare symmetry. In case of nonzero [Red, Blue]-commutators, $[R^a, B^a]$ and $[B^a, R^a]$, given their potentially decisive role in the warfare, we consider Equation (6) as the first realistic TCW model.

The spatiotemporal Red and Blue vector-fields, $R^{a}(\mathbf{x},t)$ and $B^{a}(\mathbf{x},t)$, respectively, are implemented in *Mathematica* as:

$$X = \text{Table}[x_a[t], \{a, n\}],$$

DOI: 10.4236/ica.2018.92003

$$\begin{aligned} \text{Table}\Big[k\sum_{b=1}^{n}A_{a,b}B_{b}[t]x_{a}[t] + \sum_{b=1}^{n}kk_{b}\sum_{c=1}^{n}\sum_{d=1}^{n}F_{a,b,c,d}B_{c}[t]R_{d}[t]x_{a}[t], \{a,n\}\Big], \\ \text{Table}\Big[\kappa\sum_{b=1}^{n}C_{a,b}R_{b}[t]x_{a}[t] + \sum_{b=1}^{n}\kappa\kappa_{b}\sum_{c=1}^{n}\sum_{d=1}^{n}G_{a,b,c,d}B_{c}[t]R_{d}[t]x_{a}[t], \{a,n\}\Big]. \end{aligned}$$

Their Lie-bracket commutator is defined (from scratch) via Jacobian matrix, using the following functional definitions:

$$\operatorname{JacMat}[v_List, x_List] := \operatorname{Outer}[D, v, x],$$

 $\text{LieBrc}[u_List, v_List, x_List] \coloneqq \text{JacMat}[v, x] \cdot u - \text{JacMat}[u, x] \cdot v.$

In such a way defined commutators are applied to the Red and Blue vector-fields as:

CommutRB = LieBrc[RedVec, BlueVec, X],

CommutBR = LieBrc[BlueVec, RedVec, *X*].

In our specific simulation case, both commutators evaluate to the 9-component *zero vectors*. Obtaining zero commutators means that we are dealing with the *symmetric warfare*. This also implies that the simulations of the basic tensor combat-Lanchester equations would not change.

5.3. Entropic Lie-Dragging of Combat, Strategic, Tactical and Operational Objects

Given this basis, we can explore general Lie-dragging of the opposite side combat-tensors, as well as arbitrary strategic, tactical and operational objects along the Red and Blue vector-fields. We begin with the Red and Blue combat-tensors, \mathcal{N}_b^a and \mathfrak{S}_b^a , respectively. Their Lie-dragging by the opposite side vector-fields is defined as:

$$\begin{split} \mathfrak{L}_{R}\mathcal{N}_{b}^{a} &= R^{c}\nabla_{c}\mathcal{N}_{b}^{a} - \nabla_{c}R^{a}\mathcal{N}_{b}^{c} + \nabla_{b}R^{c}\mathcal{N}_{c}^{a}, \\ \mathfrak{L}_{R}\mathfrak{N}_{b}^{a} &= B^{c}\nabla_{c}\mathfrak{N}_{b}^{a} - \nabla_{c}B^{a}\mathfrak{N}_{b}^{c} + \nabla_{b}B^{c}\mathfrak{N}_{c}^{a} \end{split}$$

and measures the degree of resistance of the opposite side sensor/computer networks. In case of nonresistance, these equations reduce to:

$$\mathfrak{L}_{R}\mathcal{N}_{b}^{a}=0$$
 and $\mathfrak{L}_{B}\mathfrak{N}_{b}^{a}=0$.

Including the Lie-dragging of the opposite side networks into the tensor Lanchester equations with [Red-Blue]-commutators (6), gives the following more realistic warfighting system:

$$\operatorname{Red}: \dot{R}^{a} = kA_{b}^{a}B^{b} + k_{b}F_{cd}^{ab}B^{c}R^{d} + \left[R^{a}, B^{a}\right] + R^{b}\mathfrak{L}_{R}\mathcal{N}_{b}^{a},$$

$$\operatorname{Blue}: \dot{B}^{a} = \kappa C_{b}^{a}R^{b} + \kappa_{b}G_{cd}^{ab}B^{c}R^{d} + \left[B^{a}, R^{a}\right] + B^{b}\mathfrak{L}_{B}\mathfrak{N}_{b}^{a},$$
(7)

where the Lie-dragging terms $(\mathfrak{L}_R \mathcal{N}_b^a)$ and $\mathfrak{L}_B \mathfrak{N}_b^a)$ are multiplied by the Red and Blue vector-fields (as $R^b \mathfrak{L}_R \mathcal{N}_b^a$ and $B^b \mathfrak{L}_B \mathfrak{N}_b^a$ terms) to preserve the *covariance* of the tensor equations.

The Lie-dragged combat Equation (7) represent the realistic TCW model that we have been looking for, the general model which generalizes the Lanchester combat equations under a networked force view of military command arrangements, and also address warfare symmetry and captures a notion of the deep uncertainty of military operational situations. Equation (7) are implemented in *Mathematica* using the following two-step process.

1) Lie-Dragging of the opposite side combat-tensors is computed using the *xTensor* package [20] as follows:

 $<< xAct^{*}xTensor^{*}; \ n = 9; \\ DefManifold[M9, n, \{a, b, c, d\}]; \\ DefTensor[R[a], M9]; DefTensor[B[a], M9]; \\ LrC = LieD[v[a]]@CC/.LieD[v[a]]@t \to 1//Simplify; \\ LbA = LieD[v[a]]@AA/.LieD[v[a]]@t \to 1//Simplify; \\ Table[LR_{a,b} = LrC[[a, b]], \{a, n\}, \{b, n\}]; \\ Table[LB_{a,b} = LbA[[a, b]], \{a, n\}, \{b, n\}]; \\ \end{cases}$

2) The temporal tensor combat Equation (7) are implemented as follows:

$$\begin{split} & \text{EqL=Flatten}[\text{Join}[\\ & \text{Table}[\partial_t R_a[t] ==k\sum_{b=1}^n A_{a,b}B_b[t] + \sum_{b=1}^n kk_b\sum_{c=1}^n \sum_{d=1}^n F_{a,b,c,d}B_c[t]R_d[t] + \sum_{b=1}^n R_b[t]LR_{a,b}, \{a,n\}],\\ & \text{Table}[\partial_t B_a[t] ==\kappa\sum_{b=1}^n C_{a,b}R_b[t] + \sum_{b=1}^n \kappa\kappa_b\sum_{c=1}^n \sum_{d=1}^n G_{a,b,c,d}B_c[t]R_d[t] + \sum_{b=1}^n B_b[t]LB_{a,b}, \{a,n\}],\\ & \text{Table}[R_a[0] ==\text{Ro} \text{ RandomReal}[], \{a,n\}], \text{ Table}[B_a[0] ==\text{Bo} \text{ RandomReal}[], \{a,n\}] \end{split}$$

A sample simulation of this initial-value problem for 10 time units with random initial conditions is given in **Figures 4-6**.

Comparing the Lie-dragged simulation (Figures 4-6) with the basic tensor combat dynamics (Figures 1-3) shows the subtle control action of the Lie-dragging effect on the overall combat Red-Blue dynamics. Specifically, Red (bipartite) forces are still loosing (Figure 4), with double amplitude. Blue (tripartite) forces are still winning (Figure 5), with double amplitude. All individual conflicts transparent in the phase plane are still present, now with double amplitude and spread uniformly along the diagonal of phase-plane (Figure 6), while in the initial tensor combat dynamics they were concentrated in the left-upper corner of the phase plane.

Finally, we consider the Lie-dragging of more complex *strategic, tactical and* operational (STO) objects, similar to F_{cd}^{ab} and G_{cd}^{ab} tensors given by Equation (5), belonging to the Red and Blue sides and defined by the new (3,1) tensors, \mathcal{T}_{bcd}^{a} and \mathcal{T}_{bcd}^{a} , respectively (also composed as triple tensor sums:

 $\operatorname{strat}_{bcd}^{a} + \operatorname{tact}_{bcd}^{a} + \operatorname{oper}_{bcd}^{a}$). Their Lie-dragging by the opposite side vector-fields is defined as:

$$\begin{split} \mathfrak{L}_{R}\mathcal{T}^{a}_{bcd} &= R^{a}\nabla_{a}\mathcal{T}^{a}_{bcd} - \mathcal{T}^{a}_{bcd}\nabla_{a}R^{a} + \mathcal{T}^{a}_{acd}\nabla_{b}R^{a} + \mathcal{T}^{a}_{bad}\nabla_{c}R^{a} + \mathcal{T}^{a}_{bca}\nabla_{d}R^{a}, \\ \mathfrak{L}_{B}\mathfrak{T}^{a}_{bcd} &= B^{a}\nabla_{a}\mathfrak{T}^{a}_{bcd} - \mathfrak{T}^{a}_{bcd}\nabla_{a}B^{a} + \mathfrak{T}^{a}_{acd}\nabla_{b}B^{a} + \mathfrak{T}^{a}_{bad}\nabla_{c}B^{a} + \mathfrak{T}^{a}_{bca}\nabla_{d}B^{a} \end{split}$$

and measures the *degree of resistance* of the opposite side STO objects. Again, in case of nonresistance, these Eqs. reduce to: $\mathfrak{L}_R \mathcal{T}_{bcd}^a = 0$ and $\mathfrak{L}_B \mathfrak{T}_{bcd}^a$

Including the Lie-dragging of the opposite side STO objects into tensor Lie-Lanchester Equation (7) gives the following even more realistic warfighting system:



Figure 4. Sample simulation of the tensor combat Equation (7): time evolution of the Red forces R^{e} for 10 time units and random initial conditions.



Figure 5. Sample simulation of the tensor combat Equation (7): time evolution of the Blue forces B^a .



Figure 6. Sample simulation of the tensor combat Equation (7): Red-Blue phase plot.

$$\operatorname{Red} : \dot{R}^{a} = kA_{b}^{a}B^{b} + k_{b}F_{cd}^{ab}B^{c}R^{d} + \left[R^{a}, B^{a}\right] + R^{b}\mathfrak{L}_{R}\mathcal{N}_{b}^{a} + R^{b}B^{c}E^{d}\mathfrak{L}_{R}\mathcal{T}_{bcd}^{a},$$

$$\operatorname{Blue} : \dot{B}^{a} = \kappa C_{b}^{a}R^{b} + \kappa_{b}G_{cd}^{ab}B^{c}R^{d} + \left[B^{a}, R^{a}\right] + B^{b}\mathfrak{L}_{B}\mathfrak{N}_{b}^{a} + R^{b}B^{c}E^{d}\mathfrak{L}_{B}\mathfrak{T}_{bcd}^{a},$$

where, as before, the Lie-dragging terms ($\mathfrak{L}_{R}\mathcal{T}_{bcd}^{a}$ and $\mathfrak{L}_{B}\mathfrak{T}_{bcd}^{a}$) are multiplied by the Red- R^{a} , Blue- B^{a} and third party- E^{a} vector-fields (as $R^{b}B^{c}E^{d}\mathfrak{L}_{R}\mathcal{T}_{bcd}^{a}$ and $R^{b}B^{c}E^{d}\mathfrak{L}_{B}\mathfrak{T}_{bcd}^{a}$ terms) to preserve the *covariance* of the tensor equations, so that each term still remains the vector-field.

6. Conclusions and Future Work

We have extended the basic tensor-centric warfare, developed in [1] as a generalized tensor union of classical Lanchester equations under the modern notion of utilizing information and communications networks in military operations to enable greater collaborative decision-making across previously acicular hierarchical command arrangements. The extension to an entropic Lie-derivative provides a stronger notion of the uncertainty inherent in battle, and describes formally the notion of symmetry in war.

Future work will examine more complex tensors, including high-dimensional Dirac-delta functions for modeling various delta-strikes and missiles, as well as tensors for learning with adoptions from deep learning/TensorFlow, and tensors of quantum origin, such as tensors from quantum computation and quantum graphs. Our research direction is motivated by the need for more realistic analytical combat models. We feel that the limited applicability of classical Lanchester-type models is less due to the limitations inherent to constructing formal models using a few idealized parameters and more to do with the intention of the model to provide predictions about combat outcomes. In other words, development of new formal combat models of greater utility hinges on the choice of problem that motivates the model in the first place; in place of earlier intentions to analytically predict outcomes, our research direction is oriented around trying to determine the uncertainty of outcomes, and particularly, to make predictions about the uncertainty or variability of outcomes for blue and red, where uncertainty is here taken in an entropic sense. We also intend to develop a richer lexicon of outcomes against which to make such entropic estimates.

The reason this matters is that decision-making under uncertainty is oriented around uncertainty with respect to failure outcomes: blue wants to steer away from uncertainty with respect to failure outcomes for blue, but towards uncertainty around failure outcomes for red, and *vice-versa*. In terms of our modeling, we thus consider decision-making of the opposing forces to be a matter of *control*.

Acknowledgements

The authors are grateful to Dr. Tim McKay and Dr. Brandon Pincombe, Joint and Operations Analysis Division, Defence Science & Technology Group, Australia for their constructive comments which have improved the quality of this paper.

References

- [1] Ivancevic, V., Pourbeik, P. and Reid, D. Tensor-Centric Warfare I: Tensor Lanchester Equations. ICA.
- [2] Alberts, D., Garstka, J. and Stein, F. (1999) Network Centric Warfare: Developing and Leveraging Information Superiority. CCRP.
- [3] Reid, D.J., Goodman, G., Johnson, W. and Giffin, R.E. (2005) All That Glisters: Is Network-Centric Warfare Really Scientific? *Defense and Security Analysis*, 21, 335-367. https://doi.org/10.1080/1475179052000345403
- [4] Reid, D.J. (2017) An Autonomy Interrogative. In: Abbass, H.A., Scholz, J. and Reid, D.J., Eds., *Foundations of Trusted Autonomy*, Springer, New York, Chapter 21, 365-391.
- [5] Rittel, H. (1973) Webber: Dilemmas in a General Theory of Planning. *Policy Sciences*, 4, 155-169.
- [6] Jomini, A.H. (2007) Baron de: The Art of War. Dover Edition, Dover Publications, New York.
- [7] Von Clausewitz, C. (1832) On War. Princeton Univ. Press, Princeton.
- [8] Lanchester, F.W. (1916) Aircraft in Warfare: The Dawn of the Fourth Arm. Constable, London.
- [9] Lanchester, F.W. (2000) Mathematics in Warfare. In: Newman, J., Ed., *The World of Mathematics*, Vol. 4, Simon and Schuster, New York, Dover, 2138-2157.
- [10] Osipov, M. (1995) The Influence of the Numerical Strength of Engaged Forces on Their Casualties. In: Military Operations Research Society, Eds., *Warfare Modeling*, Helmbold, R.L. and Rehm, A.S. Trans., John Wiley & Sons, Hoboken, 290-343.
- [11] McLemore, C., Gaver, D. and Jacobs, P. (2016) Model for Geographically Distributed Combat Interactions of Swarming Naval and Air Forces. *Naval Research Logistics*, 63, 562-576. <u>https://doi.org/10.1002/nav.21720</u>
- [12] Ivancevic, V. and Ivancevic, T. (2006) Geometrical Dynamics of Complex Systems. Springer, Dordrecht. <u>https://doi.org/10.1007/1-4020-4545-X</u>
- [13] Ivancevic, V. and Ivancevic, T. (2008) Complex Nonlinearity: Chaos, Phase Transitions, Topology Change and Path Integrals. Springer, Berlin.
- [14] Ivancevic, V. and Reid, D. (2015) Complexity and Control: Towards a Rigorous Behavioral Theory of Complex Dynamical Systems. World Scientific, Singapore.
- [15] Ivancevic, V., Reid, D. and Pilling, M. (2017) Mathematics of Autonomy: Mathematical Methods for Cyber-Physical-Cognitive Systems. World Scientific, Singapore. <u>https://doi.org/10.1142/10716</u>
- [16] Wikipedia (2017) Battlespace.
- Scholz, J.B., Calbert, G.J. and Smith, G.A. (2011) Unravelling Bueno De Mesquita's Group Decision Model. *Journal of Theoretical Politics*, 23, 510-531. https://doi.org/10.1177/0951629811418142
- [18] Ivancevic, V. and Ivancevic, T. (2007) Applied Differential Geometry: A Modern Introduction. World Scientific, Singapore. https://doi.org/10.1142/6420
- [19] Ivancevic, V. and Ivancevic, T. (2007) Neuro-Fuzzy Associative Machinery for Comprehensive Brain and Cognition Modelling. Springer, Berlin. https://doi.org/10.1007/978-3-540-48396-0
- [20] xTensor (2015) Fast Abstract Tensor Computer Algebra. http://xact.es/xTensor/
- [21] Woszczyna, A., et al. (2014) The Symbolic Tensor Analysis Package, with Tools for

General Relativity. http://library.wolfram.com/infocenter/MathSource/8848/

- [22] Ivancevic, V., Reid, D., Boswell, S. and Pourbeik, P. Evolution of Combat Models: From Linear Lanchester Equations to Nonlinear Schrödinger Equations.
- [23] Penrose, R. (2004) The Road to Reality. Jonathan Cape, London.
- [24] Nicolis, G. and Prigogine, I. (1977) Self-Organization in Nonequilibrium Systems: From Dissipative Structures to Order through Fluctuations. Wiley, Hoboken.
- [25] Ivancevic, V., Reid, D. and Scholz, J. (2014) Action-Amplitude Approach to Controlled Entropic Self-Organization. *Entropy*, 16, 2699-2712. <u>https://doi.org/10.3390/e16052699</u>
- [26] Ruppeiner, G. (1995) Riemannian Geometry in Thermodynamic Fluctuation Theory. *Reviews of Modern Physics*, 67, 605-659. https://doi.org/10.1103/RevModPhys.67.605
- [27] EurekAlert! (2017) Blue Brain Team Discovers a Multi-Dimensional Universe in Brain Networks.
- [28] Reimann, M., et al. (2017) Cliques of Neurons Bound into Cavities Provide a Missing Link between Structure and Function. Frontiers in Computational Neuroscience, 11, 48. <u>https://doi.org/10.3389/fncom.2017.00048</u>
- [29] Bassett, D. and Sporns, O. (2017) Network Neuroscience. Nature Neuroscience, 20, 353-364. <u>https://doi.org/10.1038/nn.4502</u>
- [30] Bauer, U., Kerber, M., Reininghaus, J. and Wagner, H. (2017) PHAT—Persistent Homology Algorithms Toolbox. *Journal of Symbolic Computation*, 78, 76-90. <u>https://doi.org/10.1016/j.jsc.2016.03.008</u>
- [31] Hackage (2017) Ad: Automatic Differentiation. https://hackage.haskell.org/package/ad
- [32] DiffSharp (2017) Differentiable Functional Programming. http://diffsharp.github.io/DiffSharp/

Appendix: Computational Framework

Persistent Homology Algorithms

A network-computational framework, with networks/tensors of up to millions of nodes, can be developed using the publicly available *Matlab*[®] toolbox supporting the cutting-edge topological research of *brain cliques and cavities* from computational neuroscience (the *Blue Brain* project [27] [28] [29]). It is based on the *persistent homology* algorithms on *directed simplices* [30].

General Lie Derivative Calculations

For its definition, the Lie derivative \mathfrak{L}_{v} requires only the vector-field $v^{a} \in M$ along which an arbitrary tensor T would flow or would resist the flow (*i.e.*, would be "dragged" or would resist the "dragging"). For most tensors, \mathfrak{L}_{v} is naturally expressed in terms of the *gradient operator*, $\nabla_{a} = \{\partial_{a} \equiv \partial/\partial x^{a}\}$, for $a = 1, \dots, n = \dim M \ge 4$. Only in case of high-order exterior differential forms, a more natural/elegant would be \mathfrak{L}_{v} expression in terms of Cartan's exterior derivative d (see [12] [18] and the references therein).

The following computations of the Lie derivative $\mathfrak{L}_{\nu}T$ for all possible tensors T up to the order 4 are derived using the fast tensor package *xTensor* [20] for *Mathematica*^{*}:

a) The Lie derivative of a scalar field/function $\phi \in M$ along the vector-field $v^a \in M$ is its directional derivative, *i.e.*, the inner product of the gradient $\nabla_a \phi$ and the vector-field v^a :

$$\mathfrak{L}_{v}\phi = v^{a}\nabla_{a}\phi \equiv v^{a}\partial_{a}\phi = v^{1}\partial_{1}\phi + v^{2}\partial_{2}\phi + \dots + v^{n}\partial_{n}\phi.$$

For a pair of scalar fields/functions $(\phi, \varphi) \in M$, the following two relations hold:

• The Leibniz rule:

$$\mathfrak{L}_{v}(\phi\varphi) = \varphi(\mathfrak{L}_{v}\phi) + \phi(\mathfrak{L}_{v}\varphi), \text{ and }$$

• The *linearity* relation:

$$\mathfrak{L}_{v}(c_{1}\phi+c_{2}\varphi)=c_{1}\mathfrak{L}_{v}\phi+c_{2}\mathfrak{L}_{v}\varphi.$$

These two rules can be extended to higher-order tensors.

b) The Lie derivative (along the vector-field $v^a \in M$) of another vector-field $u^a \in M$ is their Lie bracket $[v^a, u^a] \in M$ (or, commutator), which is yet another vector-field, given by:

$$\mathfrak{L}_{v}u^{a} = \left[v^{a}, u^{a}\right] = \mathfrak{L}_{u}v^{a} = v^{a}\nabla_{a}u^{b} - u^{a}\nabla_{a}v^{b},$$

and measures how non-commutative the composed flow is (*i.e.*, first flowing down the one vector-field and then flowing down the other one), since in general $v^a \circ u^a \neq u^a \circ v^a$.

The space of all vector-fields $\{v^a, u^a, w^a, \cdots\} \in M$ forms a *Lie algebra* with respect to the Lie bracket $[v^a, u^a]$. This Lie algebra satisfies the *Jacobi identity*:

$$\mathfrak{L}_{v}\left[u^{a},w^{a}\right] = \left[\mathfrak{L}_{v}u^{a},w^{a}\right] + \left[u^{a},\mathfrak{L}_{v}w^{a}\right].$$

For any (q, p)-tensor field $T \in M$, the following *Lie-algebra representation* holds:

$$\mathfrak{L}_{[v,u]}T = \mathfrak{L}_v\mathfrak{L}_uT - \mathfrak{L}_u\mathfrak{L}_vT \Longrightarrow \mathfrak{L}_{[v,u]} = \mathfrak{L}_v\mathfrak{L}_u - \mathfrak{L}_u\mathfrak{L}_v = [\mathfrak{L}_v, \mathfrak{L}_u].$$

c) The Lie derivatives of the second-order tensor fields on M (these are the most common tensors, like the stress and strain tensors in elasticity, Maxwell tensor in electrodynamics, etc.) are defined as:

In case of the covariant (0, 2)-tensor field $T_{ab} \in M$ we have:¹⁰

$$\mathfrak{L}_{v}T_{ab} = v^{c}\nabla_{c}T_{ab} + \nabla_{a}v^{c}T_{cb} + \nabla_{a}v^{c}T_{bc};$$

• In case of the mixed (1,1)-tensor field $T_b^a \in M$ along $v^a \in M$ we have: $\mathfrak{L} T^a = v^c \nabla T^a - \nabla v^a T^c + \nabla v^c T^a$:

$$\sim_{v} r_{b} = v \cdot c r_{b} \cdot c \cdot r_{b} + v \cdot b \cdot r_{c},$$

• In case of the contravariant (2, 0)-tensor field $T^{ab} \in M$ we have:

$$\mathfrak{L}_{v}T^{ab} = v^{c}\nabla_{c}T^{ab} - \nabla_{c}v^{a}T^{cb} - \nabla_{c}v^{b}T^{ac}.$$

d) The Lie derivatives of low-dimensional differential forms (*i.e.*, completely asymmetric covariant tensor fields obeying Cartan's exterior calculus) are given by:

In case of the 1-form θ_a ∈ M (e.g., the canonical form of a Hamiltonian system) we have:

$$\mathfrak{L}_{v}\theta_{a} = v^{b}\nabla_{b}\theta_{a} + \theta_{b}\nabla_{a}v^{b}.$$

• In case of the closed 2-form $\omega_{ab} \in M$ (e.g., the symplectic form of a Hamiltonian system) we have:

$$\mathfrak{L}_{v}\omega_{ab} = -2\nabla_{[a}\left(\omega_{b]c}v^{c}\right),$$

where $\omega_{[ab]}$ denotes the asymmetric part of ω_{ab} .

e) The Lie derivative of a general exterior *p*-form $\omega \in M$ (e.g., exterior forms used in Hodge-de Rham theory) along $v^a \in M$ is defined in the coordinate-free form by "Cartan's magic formula:"

$$\mathfrak{L}_{v}\omega = i_{v}d\omega + di_{v}\omega.$$

Here $d \in M$ is the exterior derivative, which commutes with the Lie derivative:

$$\mathfrak{L}_{\xi}g_{ab} = 0 \Leftrightarrow \nabla_{a}\xi_{b} + \nabla_{b}\xi_{a} \equiv \nabla_{(a}\xi_{b)} = 0, \quad (\text{with } \xi_{a} = g_{ab}\xi^{b}).$$

If the flow ϕ_i is a one-parameter group of isometries on M: $\phi_i^* g_{ab} = g_{ab}$, the vector-field ξ^a which generates ϕ_i , is a Killing vector-field. The necessary and sufficient condition for ϕ_i to be a group of isometries is: $\mathcal{L}_{\xi}g_{ab} = 0$. Consequently, the necessary and sufficient condition for ξ^a to be a Killing vector-field is that it satisfies *Killing's equation*: $\nabla^a \xi^b + \nabla^b \xi^a = 0$ or $\nabla_a \xi_b + \nabla_b \xi_a = 0$.

More generally, for any Killing vector-field $\xi^a \in M$ and any conserved, symmetric (0,2)-tensor field $T_{ab} \in M$, we have: $\nabla^a (T_{ab} \xi^b) = T_{ab} \nabla^a \xi^b = 0$.

¹⁰In particular, on a Riemannian manifold *M*, the *Killing vector-field* $\xi^a \in M$ has the property that Lie differentiation with respect to it annihilates the Riemannian metric $g_{ab} \in M$:

 $d\mathfrak{L}_{v}\omega = \mathfrak{L}_{v}(d\omega),$

which in case of scalar functions (or, 0-forms) becomes:

 $\left[\mathfrak{L}_{v}, \mathbf{d}\right] = \mathbf{0}.$

 $i_{\nu}\omega \in M$ is the contraction of the *p*-form ω and the vector-field v^a , which also commutes with \mathfrak{L}_{ν} . In case of a scalar field/function $\phi \in M$, Cartan's formula simplifies as:

$$\mathfrak{L}_v \phi = i_v \mathrm{d} \phi$$

f) The Lie derivatives of the third-order tensor fields on M (which are gradients of the above second-order tensors and the strain gradients in micromorphic elasticity, etc.) are defined as:

• Given a (3,0) tensor-field S^{abc} , its Lie derivative $\mathfrak{L}_{v}S^{abc}$ is given as

$$\mathfrak{L}_{v}S^{abc} = v^{a}\nabla_{a}S^{abc} - S^{abc}\nabla_{a}v^{a} - S^{aac}\nabla_{a}v^{b} - S^{aba}\nabla_{a}v^{c}.$$

- Given a (2,1) tensor-field S_c^{ab} , its Lie derivative $\mathfrak{L}_v S_c^{ab}$ is given as $\mathfrak{L}_v S_c^{ab} = v^a \nabla_a S_c^{ab} - S_c^{ab} \nabla_a v^a + S_a^{ab} \nabla_c v^a - S_c^{aa} \nabla_a v^b$.
- Given a (1,2) tensor-field S_{bc}^{a} , its Lie derivative $\mathfrak{L}_{v}S_{bc}^{a}$ is given as $\mathfrak{L}_{v}S_{bc}^{a} = v^{a}\nabla_{a}S_{bc}^{a} - S_{bc}^{a}\nabla_{a}v^{a} + S_{ac}^{a}\nabla_{b}v^{a} + S_{ba}^{b}\nabla_{c}v^{a}$.
- Given a (0,3) tensor-field S_{abc} , its Lie derivative $\mathcal{L}_{v}S_{abc}$ is given as $\mathcal{L}_{v}S_{abc} = v^{a}\nabla_{a}S_{abc} + S_{abc}\nabla_{a}v^{a} + S_{aac}\nabla_{b}v^{a} + S_{aba}\nabla_{c}v^{a}$.

g) The Lie derivatives of the fourth-order tensor fields on M (e.g., the Riemann and Weyl curvature tensors in gravity and the compliance and stiffness tensors in elasticity) are defined as:

- Given a (4,0) tensor-field R^{abcd} , its Lie derivative $\mathfrak{L}_{v}R^{abcd}$ is given as $\mathfrak{L}_{v}R^{abcd} = v^{a}\nabla_{a}R^{abcd} - R^{abcd}\nabla_{a}v^{a} - R^{aacd}\nabla_{a}v^{b} - R^{abad}\nabla_{a}v^{c} - R^{abca}\nabla_{a}v^{d}$.
- Given a (3,1) tensor-field R_d^{abc} , its Lie derivative $\mathfrak{L}_v R_d^{abc}$ is given as $\mathfrak{L}_v R_d^{abc} = v^a \nabla_a R_d^{abc} - R_d^{abc} \nabla_a v^a + R_a^{abc} \nabla_d v^a - R_d^{aac} \nabla_a v^b - R_d^{aba} \nabla_a v^c$.
- Given a (2,2) tensor-field R_{cd}^{ab} , its Lie derivative $\mathfrak{L}_{v}R_{cd}^{ab}$ is given as $\mathfrak{L}_{v}R_{cd}^{ab} = v^{a}\nabla_{a}R_{cd}^{ab} - R_{cd}^{ab}\nabla_{a}v^{a} + R_{ad}^{ab}\nabla_{c}v^{a} + R_{ca}^{ab}\nabla_{d}v^{a} - R_{cd}^{aa}\nabla_{a}v^{b}$.
- Given a (1,3) tensor-field R^a_{bcd} , its Lie derivative $\mathfrak{L}_v R^a_{bcd}$ is given as $\mathfrak{L}_v R^a_{bcd} = v^a \nabla_a R^a_{bcd} - R^a_{bcd} \nabla_a v^a + R^a_{acd} \nabla_b v^a + R^a_{bad} \nabla_c v^a + R^a_{bca} \nabla_d v^a$.
- Given a (0,4) tensor-field R_{abcd} , its Lie derivative $\mathcal{L}_{v}R_{abcd}$ is given as $\mathcal{L}_{v}R_{abcd} = v^{a}\nabla_{a}R_{abcd} + R_{abcd}\nabla_{a}v^{a} + R_{aacd}\nabla_{b}v^{a} + R_{abcd}\nabla_{c}v^{a} + R_{abca}\nabla_{d}v^{a}$.

h) In general, for a pair of arbitrary tensors $\left(T = T_{b_1 \cdots b_p}^{a_1 \cdots a_q}, S = S_{b_1 \cdots b_k}^{a_1 \cdots a_j}\right) \in M$, the Leibniz rule holds:

$$\mathfrak{L}_{v}(S\otimes T) = (\mathfrak{L}_{v}S)\otimes T + S\otimes (\mathfrak{L}_{v}T).$$

In case that both tensors have the same rank, $(T = T_{b_1 \cdots b_p}^{a_1 \cdots a_q}, S = S_{b_1 \cdots b_k}^{a_1 \cdots a_j}) \in M$, the linearity relation also holds:

$$\mathfrak{L}_{v}(c_{1}T+c_{2}S)=c_{1}\mathfrak{L}_{v}T+c_{2}\mathfrak{L}_{v}S.$$

All possible Lie-derivative expressions can be derived using the tensor package *xTensor* [20] for *Mathematica*.

After *Mathematica* derivation, all tensor expressions can be completely evaluated (for manifolds of any dimension) using quasi-symbolic gradients computed by *automatic differentiation*, implemented either in C++, or in functional languages like Haskell (*AD* library) [31] and/or F-sharp (*DiffSharp* library) [32].