Subjective Trusts for the Control of Mobile Robots under Uncertainty

Eugene Kagan 1,* and Alexander Rybalov 2

1 Department Industrial Engineering, Ariel University, Ariel 4076414, Israel
2 Laboratory for AI, Machine Learning, Business and Data Analytics, Tel Aviv University, Tel Aviv 6997801, Israel; alexndr_r@yahoo.com
* Correspondence: evganyk@ariel.ac.il

Abstract: The paper deals with the methods of handling imperfect information and the control of mobile robots acting under uncertainty. It is assumed that the robots act in autonomous regime and are controlled internally without referring the externally defined probabilities of the states and actions. To control the activity of the robots, we suggest the novel multi-valued logic techniques based on the recently developed measures, known as the subjective trusts. In addition to specification of the robots’ movements, such a technique allows for direct definition of the robots’ swarming using the methods of artificial neural networks with mobile neurons. The suggested methods are verified by numerical simulations and running examples. The resulting framework forms a basis for processing non-probabilistic uncertainties and making individual decisions with imperfect information.

Keywords: subjective trust; mobile robot; mobile neuron; swarm; control

1. Introduction

Functionality of autonomous systems includes three main processes: observing and measuring the environmental and system’s states, making decisions and prescribing the future actions, and conducting the chosen action aimed to change the state of the environment or of the system. In the case of complete certainty in observations and errorless decision making and acting, the models of control implement the methods originated by Maxwell in 1868 [1]. Following this approach, behavior of the system is defined by a certain dynamical system that prescribes the further actions with respect to the current observations. In the case of uncertainty, the control process is more complicated and requires handling imperfect or incomplete information about the states of the environment and of the system, processing inaccurate and ambiguous decisions and correcting erroneous actions [2,3].

Starting in the later 1950s, operating with uncertainties in the autonomous systems control is based on probabilistic methods [4–6]. It is assumed that the uncertainties in the observation results and the errors in the decisions and actions can be defined in the terms of random variables and their probabilities.

However, the use of probabilistic methods involves presumptions, which follows from the nature of probability, but do not hold in the framework of autonomous systems control. The difficulties in application of the probabilistic methods in the control of the swarm are due to two reasons.

The first is inconsistence of the probabilities to the considered situation. For example, consider a driver in the traffic flow who observes a pedestrian at the roadside. The probability that the pedestrian will cross the road on the red light is close to zero, but since it is possible, each experienced driver is ready to brake near the crosswalk. For the other example, consider observation of unknown or incompletely defined objects in a fog. Since the objects are unknown, the distances to the objects in the fog are also undefined. Hence, the probability that the certain object is observed is undefinable. Most such difficulties are considered in detail in the possibility theory [7,8].
The second reason is more complicated and follows from the nature of decision-making in the swarms, which does not imply direct minimization of the probabilities of outcomes or maximization of the probabilities of rewards (see the classical results by Kahneman and Tversky [9] and their recent validation [10]). Even in the simple case of motion toward the single target, an optimal behavior of the swarm involves a certain level of altruism of the swarm members such that in some steps the agents deny their movements to allow the movements of their neighbors [11]. In each specific task, such steps can be specified or at least approximated by stochastic optimization, however, in the general case, control of the swarm and the reactive activity of its members that leads to near optimal teleological behavior is still unclear.

In the paper we address a novel method for handling uncertainty based on the recently developed subjective trust measures [12,13]. Together with the parameterized version of the uninorm [14] and absorbing norm [15] aggregators, these measures form a formal algebra [16,17], which, in a certain sense, connects probabilistic and possibilistic approaches. Recently these measures were applied for formal description of prospects and demonstrated good correspondence with the experimental results [12]. In this paper, we continue this direction and define control of the swarm that allows to count the altruistic behavior of the agents without considering specific models and optimization of the swarm activity.

Starting from the observed relation between the subjective trusts, possibilities and necessities, we define the extended version of the uninorm and absorbing norm aggregators and the extended version of the algebra with these aggregators. The obtained algebra is considered as the algebra of control variables, which are governed directly by the decisions made by the means of the same algebra. Such an approach allows resolving uncertainties in the decision making and in the control at the same stage of the system’s evolution using the same tools. In addition, since the suggested algebra acts on the bounded interval, the processes are naturally limited, which prevents divergence of the controls even in the cases of erroneous decisions.

The suggested method is illustrated by the example of control of mobile robots. In the example, we start from the known model of a neural network with mobile neurons [18] described using subjective trusts [13] and extend this model to the use of control variables in the defined algebra. Then, swarming of mobile robots [19,20] is defined by direct implementation of the methods widely used in the descriptions of the ensembles of the neurons [21]. However, in contrast to the known methods, mobility of the neurons in the network and, consequently, of the robots in the swarm, is based both on the internal states of the neurons and the robots and on the states of the synapses, which are the values of the interconnections. In addition, we define the process of separation and unification of the synapses and specify the corresponding motion of the neurons and robots.

It should be noticed that the first attempt to use the concept of trusts for control of the mobile robots most likely appeared in the works [22,23]. In this performance centric method, the robot’s behavior is defined by the human-to-robot trust and by the human self-confidence. In other words, the higher the trust to the robot, the higher the quality of its performance and, similarly, the trust to the human is higher the better the human controls the robot. The suggested method, in contrast, can be considered as decision centric, in which the trust is governed by a priori defined algebraic rules, and the performance of the robot is a result of the operations with the trusts.

In addition, in the suggested control method the robot is defined as a dipole consisting of two neurons. However, in contrast to the original dipole activity [24], in the suggested model the dipoles are also governed by operators of the suggested algebra.

Numerical simulations verify the methods and demonstrate their convenience for definition of internal control under uncertainty with imperfect information about the states of the system and its actions. Computational complexity of the algorithms of robots’ control depends on the implemented methods of calculation of the subjective trusts. In the simulations, the trusts were calculated using hyperbolic tangent function, which does not add complexity to the algorithms.
2. Algebra of Control Variables

Control of the mobile robot is defined in the extended version of recently developed algebra that allows consideration in the same framework for both logical and arithmetical operations. We start with general definition of this algebra of multivalued logic and then present its extension on the control variables.

2.1. Algebraic Structure for Multivalued Logic

Let $\oplus_{\theta} : [0, 1] \times [0, 1] \rightarrow [0, 1]$ be a uninorm (or uninorm aggregator) [14] with neutral or identity element $\theta \in [0, 1]$, and $\otimes_{\theta} : [0, 1] \times [0, 1] \rightarrow [0, 1]$ be an absorbing norm (or absorbing norm aggregator also known as null norm) [15] with absorbing element $\theta \in [0, 1]$. The first aggregator, with respect to the value of $\theta$, extends Boolean and and or operators and the second aggregator extends Boolean not xor operator.

Similar to the Boolean binary operators, the uninorm $\oplus_{\theta}$ and absorbing norm $\otimes_{\theta}$ are symmetric and meet the commutative and associative properties. In addition, the uninorm $\oplus_{\theta}$ is transitive. The neutral $\theta$ and absorbing $\theta$ elements play the role of unit and zero for their operators, respectively, such that for any $x \in [0, 1]$ it holds that $\theta \oplus_{\theta} x = x$ and $\theta \otimes_{\theta} x = \theta$. It was proven [25] that for any $x, y \in [0, 1]$ there exist the functions $u : (0, 1) \rightarrow (-\infty, \infty)$ and $v : (0, 1) \rightarrow (-\infty, \infty)$ called generator functions such that

$$x \oplus_{\theta} y = u^{-1}(u(x) + u(y)), \quad (1)$$

$$x \otimes_{\theta} y = v^{-1}(v(x) \times v(y)). \quad (2)$$

Generator functions $u$ and $v$ are continuous, strictly monotonously increasing with the zeroes $u(\theta) = v(\theta) = 0$ and with the limits $\lim_{x \rightarrow 0} u(x) = \lim_{x \rightarrow 0} v(x) = -\infty$ and $\lim_{x \rightarrow 1} u(x) = \lim_{x \rightarrow 1} v(x) = +\infty$. The last property allows definition of the uninorm and absorbing norm on the boundary values 0 and 1 of the interval $[0, 1]$.

It is easy to show [17] that the inverse functions $u^{-1} : (-\infty, \infty) \rightarrow (0, 1)$ and $v^{-1} : (-\infty, \infty) \rightarrow (0, 1)$ have the same properties as probability distributions and, in general, can be defined by any positive sigmoid functions with the values in the interval $[0, 1]$. Then, generator functions $u$ and $v$ are the inverses of the corresponding sigmoid functions and can be considered as quantile functions.

The values from the interval $[0, 1]$ together with the operations defined by the uninorm and the absorbing norm form an algebra $\mathcal{A} = ([0, 1], \oplus_{\theta}, \otimes_{\theta})$, in which uninorm $\oplus_{\theta}$ acts as a summation operator with zero $\theta$ and absorbing norm $\otimes_{\theta}$ acts as a multiplication operator with the zero $\theta$ [16,17]. For completeness, algebra $\mathcal{A}$ also defined the operations

$$x \oplus_{\theta} y = u^{-1}(u(x) - u(y)), \quad (3)$$

$$x \otimes_{\theta} y = v^{-1}(v(x) / v(y)), \quad (4)$$

where $v(y) \neq 0$.

Algebra $\mathcal{A}$ extends the Boolean algebra $B = \langle \{0, 1\}, \land, \lor \rangle$, which with usual binary conjunction $\land$ and disjunction $\lor$ operators, and its multivalued version $B = \langle [0, 1], \land, \lor \rangle$ defined using $t$-norm and $t$-conorm [26]. In addition, algebra $\mathcal{A}$ can be considered as an arithmetic system on the interval $[0, 1]$ in which uninorm $\oplus_{\theta}$ and absorbing norm $\otimes_{\theta}$ are associated respectively with the weighted arithmetical summation “$+$” and multiplication “$\times$”. Consequently, operations $\oplus_{\theta}$ and $\otimes_{\theta}$ are associated, respectively, with the weighted arithmetical subtraction “$-$” and division “$/$”.

In general, algebra $\mathcal{A}$ is non-distributive, that is

$$(x \oplus_{\theta} y) \otimes_{\theta} z \neq (x \otimes_{\theta} z) \oplus_{\theta} (y \otimes_{\theta} z), \quad (5)$$

while $u \neq v$ or $\theta \neq \theta$; however, if $u = v$ or $\theta = \theta$ then the distributivity property holds [16,17].
Assume that generator functions $u$ and $v$ are equivalent with equivalent neutral and absorbing elements and denote $w = u = v$ and $\eta = \theta = \theta$. For an event $A$ the trusts $\tau(A)$ about $A$ are interpreted as follows:

1. $\tau(A) = 1$ means that $A$ is necessary and $\tau(A) = 0$ means that $A$ is impossible;
2. $\tau(A) = w^{-1}(1)$ means that $A$ is probable and $\tau(A) = w^{-1}(-1)$ means that $A$ is improbable;
3. $\tau(A) = w^{-1}(w^{-1}(1))$ means that $A$ is possible and $\tau(A) = w^{-1}(-w^{-1}(1))$ means that $A$ is unnecessary.

Such interpretation means that if, for example, the trust in the event $A$ is $\tau(A) = w^{-1}(1)$, then the probability of the event $A$ is $p(A) = 1$ and vice versa, and if the trust in the event $A$ is $\tau(A) = w^{-1}(-1)$, then the probability of the event $A$ is $p(A) = 0$ and vice versa. For the possibility and the necessity, the interpretation is similar.

For example, assume that generator function $w$ is

$$w(x) = \ln\left(\frac{x^b}{(1 - x^b)}\right), \quad x \in (0, 1),$$

where $b = -1/\log_2(\eta)$. Then the inverse $w^{-1}$ of the generator function is a sigmoid function

$$w^{-1}(\xi) = \left(\exp(\xi) / (1 + \exp(\xi))\right)^{1/b}, \quad \xi \in (-\infty, \infty).$$

These functions are shown in Figure 1.

![Generator function $w$ with $\eta = 0.5$](image1)

![Inverse generator function $w^{-1}$ with $\eta = 0.5$](image2)

**Figure 1.** Generator function (a) and inverse generator function (b).

The trusts defined by these functions for the pairs of necessity-impossibility, probability-improbability, and possibility-unnecessity are shown in Figure 2.

![Interpretation of the trusts' values](image3)

**Figure 2.** Interpretation of the trusts’ values.

It is seen that on one side the trust $\tau(A) = 0.73$ in probable events is higher than the trust $\tau(A) = 0.67$ in possible events, but is lower than the trust $\tau(A) = 1.00$ in necessary events. On the other side, the trust $\tau(A) = 0.27$ in improbable events is higher than the trust $\tau(A) = 0.00$ in impossible events, but is lower than the trust $\tau(A) = 0.33$ in unnecessary events.

If $A$ is interpreted as a proposition, then from a logical point of view the indicated truth values are interpreted as follows:

1. $\tau(A) = 1$ means that $A$ is objectively true and $\tau(A) = 0$ means that $A$ is objectively false;
2. \(\tau(A) = w^{-1}(1)\) means that \(A\) is subjectively true (true from the observer’s point of view) and \(\tau(A) = w^{-1}(-1)\) means that \(A\) is subjectively false (false from the observer’s point of view);

3. \(\tau(A) = w^{-1}(w^{-1}(1))\) means that \(A\) seems to be true and \(\tau(A) = w^{-1}(-w^{-1}(1))\) means that \(A\) seems to be false.

The truth values of the subjective truth and false are denoted by \(I_w\) and \(O_w\), respectively, and these values hold the following true [16,18]:

\[
0 < O_w < \eta < I_w < 1,
\]

where 0 and 1 represent Boolean false and true values that are limiting values for subjective false and subjective true, respectively. The truth value \(\eta\) indicates that it is undecided whether the proposition \(A\) is true or false.

The suggested algebra with uninorm and absorbing norm operations allows direct definition of the control of mobile robots and their swarming. In the next section, we define such a system using the extension of the considered algebra based on the model of a neural network with mobile neurons.

2.2. Algebra of Control Variables

Let us adopt the presented above algebra \(A\) to direct the handling of the control variables. For convenience, assume that the control variables obtain their values from the interval \([-1, 1]\). Such an assumption is certainly not necessary, but it meets most cases of the control systems.

Formally it is required to extend algebra \(A\) to the interval \([-1, 1]\). The first option is to apply direct and inverse linear transformations

\[
c = 2x - 1, \quad x \in [0, 1],
\]

\[
x = (c + 1)/2, \quad c \in [-1, 1].
\]

Then, the values \(c\) of control variables are converted to the trusts \(x\) and then after appropriate handling the resulting trusts \(x\) are converted to the controlled values \(c\).

The second option is to define the extended algebra \(A^*\), which inherits the properties of the algebra \(A\), but acts on the interval \([-1, 1]\). Let \(u^* : (-1, 1) \rightarrow (-\infty, \infty)\) and \(v^* : (-1, 1) \rightarrow (-\infty, \infty)\) be continuous, strictly monotonically increasing functions with the zeroes \(u^*(0^*) = v^*(0^*) = 0\) and with the limits \(\lim_{x \rightarrow 1^-} u^*(x) = \lim_{x \rightarrow 1^-} v^*(x) = -\infty\) and \(\lim_{x \rightarrow 1^+} u^*(x) = \lim_{x \rightarrow 1^+} v^*(x) = +\infty\).

Using these functions, the extended uninorm \(\oplus^*_\theta : [-1, 1] \times [-1, 1] \rightarrow [-1, 1]\) and absorbing norm \(\otimes^*_\theta : [-1, 1] \times [-1, 1] \rightarrow [-1, 1]\) are defined similarly to the Equations (1) and (2):

\[
x \oplus^*_\theta y = u^{s^{-1}}(u^s(x) + u^s(y)),
\]

\[
x \otimes^*_\theta y = v^{s^{-1}}(v^s(x) \times v^s(y)).
\]

The inverse operations then are

\[
x \oplus^*_\theta y = u^{s^{-1}}(u^s(x) - u^s(y)),
\]

\[
x \otimes^*_\theta y = v^{s^{-1}}(v^s(x)/v^s(y)),
\]

where \(v^*(y) \neq 0\).

Consequently, an extension of the algebra \(A\) to the interval \([-1, 1]\) is a triple \(A^* = ([-1, 1], \oplus^*_\theta, \otimes^*_\theta)\), in which extended uninorm \(\oplus^*_\theta\) and absorbing norm \(\otimes^*_\theta\) specify algebraic operations with zeroes \(\theta^s\) and \(\theta^s\), respectively.

Algebra \(A^*\) inherits the properties of the algebra \(A\), but since it is defined on the interval \([-1, 1]\) the trusts \(x \in [-1, 1]\) can be directly considered as control variables \(c\) and
vice versa. This algebra can be considered as a continuous version of both the balanced ternary numerical system and the ternary logic.

The defined algebra $\mathcal{A}^*$ can be used for direct definition of the control and learning system. Below we consider the extension of previously defined neural network with mobile neurons. This network is a basis for specification of navigation and swarming of mobile robots.

3. Neural Network with Mobile Neurons in Algebra $\mathcal{A}^*$

Control of the mobile robots is defined using the extended algebra presented above and uninorm and absorbing norm aggregators. Such an approach allows direct application of formal operations for specification of the robots’ actions. The swarming process is then defined using the model of neural network with mobile Tsetlin neurons.

3.1. States of the Neurons and of the Synapses

In general, in a neural network with mobile neurons it is assumed that the connectivity between the neurons depends on the distance between the neurons, and then the training process leads to restructuring the network and forming ensembles of the neurons. On the other hand, the distance between the neurons is defined with respect to the connectivity between the neurons; thus, the learning process is governed by the neurons’ motion [18].

Let us consider a neural network defined in the algebra $\mathcal{A}^*$. Formally this network follows the previously defined Tsetlin network acting in algebra $\mathcal{A}$ [13], but the use of the interval $[-1, 1]$ instead of the interval $[0, 1]$ results in more convenient specification of the neurons’ motion.

Assume that the $i$th neuron obtains the inputs via $l$ synapses and that at each time $t = 0, 1, 2, \ldots$ the input value $x_k(t)$ appears at the $k$th synapse, $k = 1, 2, \ldots, l$, which is characterized by the weight $\omega^k(t)$. The weight can be considered as a transition possibility, which is the ability of the synapse to transfer the input value or as a transition trust that is the trust that the synapse transmits the input value. In the considered framework, transition possibility and transition trust have the same meaning and specify the level of connectivity between the neurons. At the same time, as a logical value, the weight $\omega^k(t)$ is an operand of multi-valued logical operations.

In the Tsetlin network defined in $\mathcal{A}^*$, each neuron acts as follows. The input value $x_k(t)$ is aggregated with the weight $\omega^k(t)$ by the uninorm $\oplus^*_\beta$ or by the absorbing norm $\otimes^*_\beta$ and the resulting value is transmitted to the neuron. The neuron compares the obtained value with its internal state $s_i(t)$ using the absorbing norm $\otimes^*_\beta$. Finally, using the uninorm $\oplus^*_\beta$ the neuron aggregates the results of such a comparison for all $l$ inputs and specifies the obtained result as an output $z_i(t)$, which can also be (or can be not) used as the next internal state $s_i(t + 1)$.

Formally, such activity is defined as follows. Let $(x_1(t), x_2(t), \ldots, x_l(t))$ be the values appearing at the synapses of the neuron at time $t$, $s_i(t)$ be the neuron’s internal state and $(\omega^1(t), \omega^2(t), \ldots, \omega^l(t))$ be the weights of the neuron’s synapses. The output values of synapses $k = 1, 2, \ldots, l$ are defined using the uninorm $\oplus^*_\beta$ ($u$-synapses)

$$y_k(t) = x_k(t) \oplus^*_\beta \omega^k(t)$$  \hspace{1cm} (15)

or using the absorbing norm $\otimes^*_\beta$ ($a$-synapses)

$$y_k(t) = x_k(t) \otimes^*_\beta \omega^k(t).$$  \hspace{1cm} (16)

Then each of these values is compared with the current state of the neuron using the absorbing norm, that is

$$c_k(t) = y_k(t) \otimes^*_\beta s_i(t),$$  \hspace{1cm} (17)
and, finally, the results of comparisons are aggregated using the uninorm, and the output of the neuron is specified by
\[ z_i(t) = \bigoplus_{k=1}^{t} c_k(t). \] (18)

The next state \( s_i(t + 1) \) of the neuron is specified by a certain state function, which in the simplest case is
\[ s_i(t + 1) = z_i(t). \] (19)

Let us consider the weights \( \omega(t) \in [-1, 1] \) of the synapses, which specify the strength of the connection between the neurons. The changes of these values are associated with the learning process. The weights can be defined either externally by a certain training process or internally using the states of the neurons. In the first case, the network converges to a definite configuration, which leads to the desired result on the output neurons, while in the second case the network demonstrates certain self-organization [27] and can change its configuration with respect to the states of the neurons and of the environment. In the suggested model, we follow the second approach and define the transition possibilities with respect to the states of the interconnecting neurons.

Let \( N_i \) and \( N_j \) be two neurons connected by the synapse \( S_{ij} \). The states of the neurons at time \( t \) are denoted by \( s_i(t) \) and \( s_j(t) \) and the weight of the synapse is denoted by \( \omega_{ij}(t) \). Notice that in the notation of the weight \( \omega^k \) the upper index stands for \( k \)th input of the neuron and in the notation \( \omega_{ij} \) the bottom pair of indices represents the neurons interconnected by the considered synapse. Using the absorbing norm \( \otimes_a \), the value \( \omega_{ij}(t) \) is defined as
\[ \omega_{ij}(t) = s_i(t) \otimes_a s_j(t), \] (20)
which, in essence, is a result of the comparison between the neurons’ states \( s_i(t) \) and \( s_j(t) \).

In the network with mobile neurons, the weights \( \omega_{ij}(t) \) are applied for two goals. The first is a conventional use for specification of the output \( y_j(t) \) of the neuron \( N_j \) (see Equations (15) and (16)), and the second is a definition of the value of the potential function \( u \) in the location of the synapse. The second use is possible since in the considered approach the network is implemented in the form of physical devices with definite locations on a plane. Namely, if the neurons \( N_i \) and \( N_j \) are associated with the mobile robots located in the points \( c_i \) and \( c_j \), then it is assumed that the synapse is located at the geometrical center \( z_{ij} \) of the line connecting these points. Figure 3 shows an example of locations of three neurons \( N_1, N_2, \) and \( N_3 \) and three synapses \( S_1, S_2, \) and \( S_3 \) connecting these neurons.

![Locations of neurons and synapses](image-url)  
**Figure 3.** Locations of three neurons \( N_1, N_2, \) and \( N_3 \) and three synapses \( S_1, S_2, \) and \( S_3 \) connecting the neurons.
In the definition of the potential function, the neurons and, consequently, the robots are considered as obstacles and the potential function in their locations obtains the highest value 1. In the other points of the plane the values of the potential function are defined by an appropriate smoothing. The potential function for the neurons and the synapses shown in Figure 3 with the neurons’ states \( s_1 = -1, s_2 = 1 \) and \( s_3 = 1 \) and the weights \( \omega_{13} = \omega_{12} = -1 \) and \( \omega_{23} = 1 \) is shown in Figure 4.

![Potential function](image)

**Figure 4.** Potential function for the neurons and the synapses shown in Figure 3. The states of the neurons are \( s_1 = -1, s_2 = 1 \) and \( s_3 = 1 \) and the weights of the connections (the states of the synapses) are \( \omega_{13} = \omega_{12} = -1 \) and \( \omega_{23} = 1 \).

It is seen that the locations of the neurons and the location of the third synapse that connects the neurons \( \mathcal{N}_1 \) and \( \mathcal{N}_3 \) potential are positive and the neurons (which are the mobile robots) are repulsed from these locations. However, the locations of the synapses that connect the neuron \( \mathcal{N}_1 \) with the neurons \( \mathcal{N}_2 \) and \( \mathcal{N}_3 \) potential are negative, and the neurons (or robots) are attracted to these locations.

### 3.2. Reactive Learning and Motion of the Neurons

Learning in the considered network is based on the changes of the connectivity between the neurons and on the movements of the neurons with respect to the defined potential function. In addition, it is assumed that the neuron can substitute the synapse, to which location the neuron arrived, which results in creating two new synapses, connecting the arrived neuron with the neurons, which were connected by the substituted synapse. The process of synapse substitution and creation of new synapses is shown in Figure 5.

In the considered scenario, synapse \( S_{12} \) before substitution had a negative value and attracted the neuron. After division, the synapse \( S_{1k} \) starts with the values equivalent to the value of synapse \( S_{12} \) and then changes it to the value calculated by Equation (20) with respect to the states of the neurons \( \mathcal{N}_1 \) and \( \mathcal{N}_k \) (for synapse \( S_{1k} \)) and \( \mathcal{N}_k \) and \( \mathcal{N}_2 \) (for synapse \( S_{2k} \)).

In a similar manner, two neighboring neurons can repulse the neuron with the same sign of the state value. The process, in which the neurons \( \mathcal{N}_1 \) and \( \mathcal{N}_2 \) repulse neuron \( \mathcal{N}_k \), is shown in Figure 6.
Figure 5. The process of synapse substitution and creation of new synapses: (a) neuron \( N_i \), which is connected with the neurons \( N_j \) and \( N_j \) via the synapses \( S_{ik} \) and \( S_{kj} \), moves toward the synapse \( S_{12} \), which connects the neurons \( N_1 \) and \( N_2 \); (b) the synapse \( S_{12} \) is “divided” into two synapses; (c) neuron \( N_j \) substitutes synapse \( S_{12} \) in its location and connects with the neurons \( N_1 \) and \( N_2 \) via new synapses \( S_{1k} \) and \( S_{k2} \).

Figure 6. The process of repulsion of the neuron and union of the synapse: (a) neurons \( N_j \) and \( N_j \) repulse neuron \( N_i \); (b) synapses \( S_{ik} \) and \( S_{kj} \) move toward each other; (c) synapses \( S_{ik} \) and \( S_{kj} \) are united into the synapse \( S_{12} \).
Similar to the Coulomb law for the electrically charged particles, in the considered model it is assumed that each neuron attracts the neurons with different signs of the states and repulses the neurons with the same sign of the state. Using the value

\[
arp(N_i, N_j) = u^{-1} (s_i \otimes s_j) = u^{-1} (\|\omega_{ij}\|),
\]

(21)
of attraction/repulsion between the neurons \(N_i\) and \(N_j\), where \(s_i\) and \(s_j\) are the states of the neurons \(N_i\) and \(N_j\), respectively, the attraction/repulsion force for these neurons is defined by the formula

\[
F(N_i, N_j) = \lambda \ arp(N_i, N_j) / dist(N_i, N_j),
\]

(22)
where \(dist(N_i, N_j)\) is a geometrical distance between the neurons and \(\lambda\) is an attraction/repulsion coefficient. In the Equation (21), the value \(s_i \otimes s_j\) represents similarity between the states, and in the Equation (22) the distance between the neurons is defined by the metric of the space, where the network acts; in the considered case it is Euclidian distance.

Certainly, attraction/repulsion can be defined differently with respect to the needs and requirements of the network. For example, attraction and repulsion can be defined using the known aggregation function [19]

\[
I_{agr}(N_i, N_j) = -dist(N_i, N_j) \left( a - b \exp \left( -\frac{dist^2(N_i, N_j)}{c} \right) \right),
\]

(23)
where the states of the synapses and of the neurons are expressed by the parameters \(a, b,\) and \(c\), or by using different similarity measures.

3.3. Simulation of the Network Activity

To clarify activity of the defined network with mobile neurons let us consider the following example. Assume that the network is in the gridded square domain of the size \(N \times N = 100 \times 100\) cells and that the number of the neurons in the network is \(n = 25\).

Each neuron \(N_i, i = 1, 2, \ldots, n\), is connected with the neighboring neurons \(N_j, j = 1, 2, \ldots n, j \neq i\), which are located at the distances \(dist(N_i, N_j)\) bounded by certain predefined threshold \(d_{max}\). For the illustrations below we used Euclidian distance and the threshold \(d_{max} = 30\).

If the distance \(dist(N_i, N_j)\) is less than the threshold \(d_{max}\), then the state of the synapse \(S_{ij}\), connecting the neurons \(N_i\) and \(N_j\), which is the weight \(\omega_{ij}\), is calculated according to Equation (20) with respect to the current states \(s_i(t)\) and \(s_j(t)\) of the neurons \(N_i\) and \(N_j\), correspondingly. Otherwise, if the distance \(dist(N_i, N_j)\) is greater than the threshold \(d_{max}\), then it is assumed that the neurons are not connected and the weight \(\omega_{ij}\) is set to zero.

The attraction/repulsion force \(F(N_i, N_j)\) for the neurons \(N_i\) and \(N_j\) is calculated according to Equation (22), where attraction/repulsion coefficient \(\lambda\) is a tenths part of the domain diagonal, that is, \(\lambda = 0.1 \sqrt{2N^2} = 14.14\).

Following the attraction/repulsion forces, the neurons move in the resultant directions by the steps proportional to the values of the resultant attraction/repulsion forces. In the illustrations it is assumed that the lengths \(\delta\) of the steps are equivalent to the values of the attraction/repulsion forces.

The states \(s_i(t)\) of the neurons are initialized by random such that \(s_i(0)\) are drawn from the interval \([-1, 1]\) with respect to uniform distribution, and then these values as well as the weights of the connections are updated using the Equations (15)–(19). Examples of the evolution of the network structures are shown in Figures 7 and 8 (the videos of the evolution of the network structures can be found in Supplementary Materials, Videos S1 and S2, respectively). In Figure 7 the starting configuration is a regular lattice and in Figure 8 the starting configuration is random.
Following the attraction/repulsion forces, the neurons move in the resultant directions by the steps proportional to the values of the resultant attraction/repulsion forces. In the illustrations it is assumed that the lengths $\delta$ of the steps are equivalent to the values of the attraction/repulsion forces.

The states $s_i(t)$ of the neurons are initialized by random such that $s_i(0)$ are drawn from the interval $[-1, 1]$ with respect to uniform distribution, and then these values as well as the weights of the connections are updated using the Equations (15)–(19). Examples of the evolution of the network structures are shown in Figure 7 and 8 (the videos of the evolution of the network structures can be found in Supplementary Materials, video S1 and S2, respectively). In Figure 7 the starting configuration is a regular lattice and in Figure 8 the starting configuration is random.

**Figure 7.** Evolution of the network structure: (a) initial regular configuration of the network, $t = 0$; (b) configuration after the first movement of each neuron, $t = 1$; (c) configuration after two movements of each neuron, $t = 2$; and (d) configuration after hundred movements of each neuron, $t = 100$.

It is seen that initially the regular configuration of the network (Figure 7a) is disturbed by the first movements of the neurons (Figure 7b), and this distortion is enough for serious change of the network configuration at the next step (Figure 7c). Figure 7d shows the neuron groups observed after the 100th step of each neuron.

Evolution of the network configuration starting from random configuration is shown in Figure 8.

It is seen that, similar to the previous example, the first motion of the neurons slightly changes the initial configuration of the network (cf. Figure 8a,b). However, already at the third step the configuration of the network changes seriously (Figure 8c). Figure 8d shows the configuration of the network after the 100th step of each neuron, which demonstrates that the initially connected network is divided to non-connected neurons, which preserve their locations.

In order to clarify the evolution of the network structure, let us consider the following example. Assume that the network includes $n = 4$ interconnected neurons as it is shown in Figure 9.
Figure 8. Evolution of the network structure: (a) initial random configuration of the network, $t = 0$; (b) configuration after the first movement of each neuron, $t = 1$; (c) configuration after two movements of each neuron, $t = 2$; and (d) configuration after hundred movements of each neuron, $t = 100$.

Figure 9. The network with four interconnected neurons.
Assume that the initial states \( s_i(0) \) of the neurons \( N_i, i = 1, \ldots, 4 \), are

\[
s_1(0) = -0.75, \ s_2(0) = -0.25, \ s_3(0) = 0.25 \text{ and } s_4(0) = 0.75.
\]

Then the weights \( \omega_{ij}(0), i,j = 1, \ldots, 4 \), of the connections defined by Equation (20) are defined by the matrix:

\[
\omega(0) = \begin{pmatrix}
0 & 0.2435 & -0.2435 & -0.7383 \\
0.2435 & 0 & -0.0651 & -0.2435 \\
-0.2435 & -0.0652 & 0 & 0.2435 \\
-0.7383 & -0.2435 & 0.2435 & 0
\end{pmatrix}.
\]

Direct application of the Equations (15) and (17)–(19) results in the following states \( s_i(1), i = 1, \ldots, 4 \), at the time \( t = 1 \):

\[
s_1(1) = -0.0256, \ s_2(1) = -0.0485, \ s_3(1) = -0.0817 \text{ and } s_4(1) = -0.9534.
\]

Then, the weights \( \omega_{ij}(1), i,j = 1, \ldots, 4 \), are:

\[
\omega(1) = \begin{pmatrix}
0 & 0.0012 & 0.0021 & 0.0478 \\
0.0012 & 0 & 0.0040 & 0.0905 \\
0.0021 & 0.0040 & 0 & 0.1518 \\
0.0478 & 0.0905 & 0.1518 & 0
\end{pmatrix}.
\]

By the same manner, the states \( s_i(1), i = 1, \ldots, 4 \), at the time \( t = 2 \) are

\[
s_1(2) = 0.0498, \ s_2(2) = -0.0910, \ s_3(2) = -0.1450 \text{ and } s_4(2) = -0.2477.
\]

The weights \( \omega_{ij}(2), i,j = 1, \ldots, 4 \), at this time are

\[
\omega(2) = \begin{pmatrix}
0 & 0.0046 & 0.0073 & -0.0126 \\
0.0046 & 0 & 0.0133 & -0.0231 \\
0.0073 & 0.0133 & 0 & -0.0369 \\
-0.0126 & -0.0231 & -0.0369 & 0
\end{pmatrix}.
\]

Finally, already at the time \( t = 3 \) the states become closer to zero

\[
s_1(3) = -0.0008, \ s_2(3) = -0.0057, \ s_3(3) = -0.0187 \text{ and } s_4(3) = -0.0542,
\]

and the weights obtain the zero values \( \omega_{ij}(3) = 0, i,j = 1, \ldots, 4 \),

\[
\omega(3) = (0).
\]

Thus, the neurons are separate and at the next step, \( t = 4 \), the states also obtain zero values

\[
s_1(4) = 0, \ s_2(4) = 0, \ s_3(4) = 0 \text{ and } s_4(4) = 0,
\]

which are preserved infinitely. Since the movements of the neurons are defined by the similarity between the neurons’ states (see Equation (21)), then the motions are zeroed, and the neurons stay at their locations.

The dynamics of the neural network with mobile neurons allows both its usage in traditional machine learning applications, and as a model of swarm activity. In the next section we implement the second possibility and apply the constructed network for specification of the swarm activity.
4. Robot States and Movements

In the suggested approach, activity of the group of mobile robots is modeled using a neural network with mobile neurons. The states of the robots are associated with the states of the neurons, and the neurons’ assembling is associated with the robots’ swarming and creating subgroups in the swarm.

Motion of the neurons can be mapped into the motion of the robots in two ways: either by direct transformation of the neuron’s movements into the movements of the robot or by the use of separate networks for specification of the movements in different dimension. In the first case, the activity if the swarm mimics the activity of the network presented in Section 3.3, while in the second case the swarm dynamics demonstrate additional properties; below we consider the second definition of the swarm activity.

4.1. Robot States under Uncertainty

Consider a mobile robot acting on a plane and assume that the state of the robot is defined by its location and heading. The robot can move one step forward, turn left or right and then move one step forward in a new direction and stay in its current location with current heading.

If the state of the robot is certain, then the next state of the robot is defined by the chosen movement: if the movement is certain, then the next state is also certain and if the movement is uncertain, the next state is uncertain with the same grade of uncertainty. If the state (either the robot’s location or heading or both) is uncertain, then the next state of the robot is also uncertain and for an uncertain movement it includes both uncertainty of the state and of the movement.

In the probabilistic terms, uncertainties of the states and movements are defined by the states’ and transitions’ probabilities. Then the dynamics of the are is governed by the Markov process, which specifies the probabilities of the next robot’s locations and heading. In the terms of subjective trusts, the robot’s dynamics is defined by similar manner, but instead of the Markov process the subjective Markov process [13] is used.

Denote by \( \xi(t) = (\xi_1(t), \xi_2(t)) \) the vector of the decision-maker’s trusts regarding the robot’s heading at time \( t \). It is assumed that
- The trust vector \( \xi(t) = (1, 1) \) means that the heading of the robot is necessary “↑”;
- The trust vector \( \xi(t) = (-1, -1) \) means that the heading of the robot is necessary “↓”;
- The trust vector \( \xi(t) = (1, -1) \) means that the heading of the robot is necessary “←”;
- The trust vector \( \xi(t) = (-1, 1) \) means that the heading of the robot is necessary “→”.

The intermediate values \( \tau_{ij}(t) \in (-1, 1) \), \( i, j = 1, 2 \), specify the levels of the trust that the robot has certain direction.

The trusts of the decision-maker about the turns of the robot are represented by the matrix

\[
T(t) = \begin{bmatrix}
\tau_{11}(t) & \tau_{12}(t) \\
\tau_{21}(t) & \tau_{22}(t)
\end{bmatrix},
\]  

(24)
of the trusts such that
- The trust matrix \( T(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \) means preserving current direction of the robot with necessity;
- The trust matrix \( T(t) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \) means turn left with necessity;
- The trust matrix \( T(t) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \) means turn right with necessity.

The intermediate values \( \tau_{ij}(t) \in (-1, 1) \), \( i, j = 1, 2 \), specify the levels of the trust that the robot turns in certain direction.
Notice that the suggested notation coincides (up to inambiguous mapping) with the usual definition of headings by the vectors $\xi = (0, 1) \triangleq \uparrow$, $\xi = (0, -1) \triangleq \downarrow$, $\xi = (-1, 0) \triangleq \leftarrow$ and $\xi = (1, 0) \triangleq \rightarrow$, and of the turns by the rotation matrix

$$R = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix},$$

(25)

where $\varphi$ is a rotation angle such that $R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ means “preserve the current heading”, $R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ means “turn left” and $R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ means “turn right”. In addition, the trust vector $\bar{\xi}(t)$ and the matrix $\mathcal{T}(t)$ can be considered in the terms of quantum controlled mobile robots and the observations of their states [28] that provides an additional framework for the suggested model.

The turns of the robot in the terms of the trusts are defined by the same manner as its turns using rotation matrix. Namely, if $\bar{\xi}(t) = (\xi_1(t), \xi_2(t))$ is the trust vector of the robot’s state at time $t$, then the trust vector $\bar{\xi}(t + 1) = (\xi_1(t + 1), \xi_2(t + 1))$ of the state at time $t + 1$ is defined as follows

$$\bar{\xi}(t + 1) = \bar{\xi}(t) \otimes_{\phi} \mathcal{T}(t),$$

(26)

where, similar to the usual multiplication rule of the vector and the matrix,

$$\xi_1(t + 1) = (\xi_1(t) \otimes_{\phi} \tau_{11}(t)) \oplus_{\phi} (\xi_2(t) \otimes_{\phi} \tau_{21}(t)),$$

$$\xi_2(t + 1) = (\xi_1(t) \otimes_{\phi} \tau_{12}(t)) \oplus_{\phi} (\xi_2(t) \otimes_{\phi} \tau_{22}(t)).$$

(27)

Evolution of the trusts defined by Equation (26) is known as the subjective Markov process [13].

In order to define transition trusts $\tau_{ij}(t)$, $i, j = 1, 2$, let us associate the values $\xi_1(t)$ and $\xi_2(t)$ as the states $s_1(t)$ and $s_2(t)$ of the Tsetlin neurons $N_1$ and $N_2$ (see Section 3.1), which are connected with themselves and with each other. Then, following the definitions of the neuron’s functions (15)–(19), the transition trusts are defined as

$$\tau_{11}(t) = \oplus_{\omega_{k1}=1}^{s_1(t)} (s_{k1} \otimes_{\phi} \omega_{k1}(t))$$

and

$$\tau_{12}(t) = s_{1}(t) \otimes_{\phi} \omega_{12}(t),$$

or as

$$\tau_{21}(t) = \oplus_{\omega_{k1}=1}^{s_1(t)} (s_{k1} \otimes_{\phi} \omega_{k1}(t))$$

(28)

$$\tau_{22}(t) = s_{2}(t) \otimes_{\phi} \omega_{12}(t),$$

(29)

with respect to the type of synapses—$u$-synapses (Equation (15)) or $a$-synapses (Equation (16))—used in the connections between the neurons.

Connections of the neurons are shown in Figure 10.

**Figure 10.** Connection of the neurons that represent the state of the robot.
In the figure, the transition trusts $\tau_{ij}(t)$, $i, j = 1, 2$, represent internal controls in the robot and the trusts $\tau_{ij}(t)$, $i, j = 3, \ldots, k, \ldots, l$, represent connections of the robot with the other robots in the swarm.

Finally, let us consider the mobility of the neurons and its use for definition of the robots’ swarming. Let $R_1$ and $R_2$ be two mobile robots moving on a plane by conducting the indicated above motions.

Assume that the state of each robot is defined by the corresponding trust vector: vector $\xi^1_i(t) = (\xi^1_{1i}(t), \xi^1_{2i}(t))$ for the robot $R_1$ and vector $\xi^2_i(t) = (\xi^2_{1i}(t), \xi^2_{2i}(t))$ for the robot $R_2$. As above, let us associate the elements of the trust vectors $\xi^1_i(t)$ and $\xi^2_i(t)$ with the states of the neurons. In other words, we assume that the state of each robot is defined by the pair of neurons, that is $R_1 = (N_1^1, N_2^1)$ and $R_2 = (N_1^2, N_2^2)$, and the states $s^i_j(t)$ of the robots are associated with the trusts $\xi^i_j(t)$, $i, j = 1, 2$.

Then, the attraction and repulsion between the robots are the combination of the attraction and repulsion between each pair of the neurons associated with the robots. Using Equation (22), the attraction/repulsion force $F(R_1, R_2)$ between the robots consists of four forces

$$F(R_1, R_2) = \left\{ F\left(N_1^1, N_2^1 \right) : i, j = 1, 2 \right\}.$$  

(30)

The scheme of attraction and repulsion of the robots is shown in Figure 11.

![Figure 11](image-url)

Figure 11. Attraction/repulsion forces between the robots. The bold arrows indicate the heading of the robots.

It is seen that each robot together with the applied forces can be considered as a dipole. Consequently, the group of the robots forms a dipole dynamical system [24] governed by the rules based on extended uninorm and absorbing norm. Below, we present numerical simulations of the behavior of such system.

4.2. Simulation of the Robots’ Motion and Swarming

Activity of the group of the robots was simulated in a similar setting as the above considered activity of the neural network with mobile neurons. The group includes $n = 25$ robots $R_i = (N_1^i, N_2^i)$, $i = 1, 2, \ldots, n$, acting in the gridded square domain of the size $N \times N = 100 \times 100$.

Each neuron $N_1^i$ and $N_2^i$ of each robot $R_i$ is connected with the neurons $N_1^j$ and $N_2^j$ of the neighboring robot $R_j$, $j = 1, 2, \ldots, n_i \leq n - 1, j \neq i$, which are located at the distances less or equal to the threshold $d_{max}$. As above, in the simulations the distances are Euclidian, and the threshold is $d_{max} = 30$. The value of the attraction/repulsion coefficient $\lambda$ is also $\lambda = 0.1 \sqrt{2N^2} = 14.14$. 

Entropy 2022, 24, x FOR PEER REVIEW 16 of 22
In the first simulations, the states $s_i(t)$ and $s_j(t)$ of the neurons are initialized by random, such that both $s_i(t)$ and $s_j(t)$ are drawn from the interval $[-1, 1]$ with respect to uniform distribution and then these values as well as the weights of the connections are updated using Equations (15)–(19). Figure 12 shows the locations of the robots starting from the ordered configuration (the video can be found in Supplementary Materials, Video S3). In this figure and the figures below the neuron $\mathcal{N}_1$ is depicted by a white circle and the neuron $\mathcal{N}_2$ is depicted by a black circle.

Figure 12. Locations of the robots starting from the position in ordered configuration: (a) initial locations, $t = 0$; (b) locations after the first movement of each robot, $t = 1$; (c) locations after the second movement of each robot, $t = 2$; and (d) locations after the 100th movement of each robot, $t = 100$.

It is seen that similar to the behavior of the neural network with mobile robots (see Figure 7), the initially ordered configuration of the robots’ locations (Figure 12a) is disturbed by the first movements of the robots (Figure 12b), and this distortion increases with the next movement (Figure 12c). Figure 12d shows configuration of the robots’ locations after the 100th movement of each robot.

Locations of the robots starting from random configuration are shown in Figure 13 (the video can be found in Supplementary Materials, Video S4).
Figure 13. Locations of the robots starting from the position in random configuration: (a) initial locations, \( t = 0 \); (b) locations after the first movement of each robot, \( t = 1 \); (c) locations after the second movement of each robot, \( t = 2 \); and (d) locations after the 100th movement of each robot, \( t = 100 \).

It is seen that the initially dense group of the robots (Figure 13a) diffuses by the first and the second movements of the robots (Figure 13b,c). Figure 13d shows scattered configuration of the robots’ locations after the 100th movement of each robot.

Notice that in both scenarios the states of both \( N^1_i \) and \( N^2_i \) in each \( i \)th robot, \( i = 1, 2, \ldots, 25 \), were initialized equivalently by random values from the interval \([-1, 1]\). In other words, the robots were not directed, and this fact is seen in the figures, where the neurons \( N^1_i \) and \( N^2_i \) attracted and repulsed with no concern to their index.

In the final simulations, in contrast, the states \( s^1_i(t) \) and \( s^2_i(t) \) of the neurons are initialized by random, such that \( s^1_i(0) \) are drawn from the interval \([0, 1]\) and \( s^2_i(t) \) are drawn from the interval \([-1, 0]\) with respect to uniform distribution. Thus, in these scenarios the robots are directed in such a manner that the neurons with the states of opposite signs attract and the neurons with the states of the same signs repulse. The states and the weights of the connections are updated using the Equations (15)–(19).

As above, in the simulations the robots started from the locations in ordered configuration and from the locations in random configuration. Figure 14 shows the locations of the robots starting from the ordered configuration (the video can be found in Supplementary Materials, Video S5).
boundary values; as a result, the connections between the robots are preserved. Together with that, the states and connections of some of the robots converge to the steady state values such that the robots keep their locations.

\[(a)\] \[(b)\] \[(c)\] \[(d)\] Figure 14. Locations of the directed robots starting from the position in ordered configuration: (a) initial locations, \(t = 0\); (b) locations after the first movement of each robot, \(t = 1\); (c) locations after the second movement of each robot, \(t = 2\); and (d) locations after the 100th movement of each robot, \(t = 100\).

It is seen that the movement of the robots is similar to the movement in the scenario illustrated by Figure 12, with the expected difference in attraction and repulsion of the robots’ neurons with equal and different signs depicted by black and white circles, respectively.

Finally, locations of the robots starting from random configuration are shown in Figure 15 (the video can be found in Supplementary Materials, Video S6).

As it was expected, in this scenario, the motion of the robots is similar to the motion illustrated by Figure 13, with the already-mentioned difference in the repulsion because of the difference between the signs of the neurons’ states.

The simulations demonstrate that starting from arbitrary locations the robots demonstrate variative motion in the group. The states of the neurons in the robots differ from the boundary values; as a result, the connections between the robots are preserved. Together with that, the states and connections of some of the robots converge to the steady state values such that the robots keep their locations.

The observed simulated behavior of the robots follows the behavior of the suggested neural network with mobile neurons, which governs each of the robots’ neurons, which allows consideration of decision making and the actions in the same framework.
algebra with parameterized unary and binary aggregators, which implement multi-valued logical operations.

5. Discussion

The suggested approach of the control of mobile robots considers decision making and operating as a unified process based on the novel method of handling imperfect information and acting under uncertainty. The method is based on the extension of recently developed algebra with parameterized unary and binary aggregators, which implement multi-valued logical operations.

Formally, the suggested method follows the general scheme of fuzzy control (see, e.g., [29,30]), but, in contrast to the usual techniques, it is focused on the algebraic model and on the relation between possibilities, probabilities, and necessities. The measures of uncertainty are considered as subjective trusts, which specify both decisions and actions of the system.

Control of the robots’ motion in the group is based on the neural network with the mobile neurons. The idea of such a network with mobile neurons was suggested by Apolloni et al. [18]. In addition to the usual activity of neural network, in this network the mobility of the neurons is applied, and it both defines and is defined by the connectivity between the neurons. In the suggested model the network consists of the Tsetlin neurons, and connections between the neurons and their mobility are governed by the states of the neurons. The activity of the network is defined in the developed algebra of the subjective trusts.

Figure 15. Locations of the directed robots starting from the position in random configuration: (a) initial locations, \( t = 0 \); (b) locations after the first movement of each robot, \( t = 1 \); (c) locations after the second movement of each robot, \( t = 2 \); and (d) locations after the 100th movement of each robot, \( t = 100 \).
From a control point of view, each robot is specified as a dipole [24], which originally was used by Tchieu et al. in the model of behavior of fish flocks. In the suggested model, the robot consists of two mobile neurons acting in the neural network with obvious limitation led by the size of the robot. Such a model allows to consider the control of the robot by the same techniques as the control of quantum controlled mobile robots [6,18], but with stressing the difference in the topology of the actions’ space.

Since the suggested model applies subjective measures of uncertainty, control is considered from the internal point of view and follows the topology of a non-oriented surface. Such topology most likely differs from the topology of the controls used in the performance centered model by Saeidi [22,23] and of the controls in the model of quantum control [28,31]; these issues will be addressed in future studies.

Finally, the paper does not address teleologic issues or specific applications and can be considered as a proof of concept. Numerical simulations have demonstrated the feasibility of the suggested method. At the next steps, the suggested method is planned to be implemented in the search and foraging algorithms. In the simple case of following toward a single target, the exact values of the trusts can be defined using the recently calculated probabilities of altruistic and egoistic steps of the agents [11], while in more complicated scenarios, additional efforts and comparisons of the suggested method with the known techniques will be required.

**Supplementary Materials:** The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/e24060790/s1, Videos S1–S6.

**Author Contributions:** Conceptualization, E.K. and A.R.; methodology, E.K. and A.R.; software, E.K.; validation, E.K. and A.R.; writing—original draft preparation, E.K.; writing—review and editing, A.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Maxwell, J.C. On governors. Proc. R. Soc. Lond. 1868, 16, 270–283.
2. Aliev, R.; Huseynov, O. Decision Theory with Imperfect Information; World Scientific: Singapore, 2014.
3. Feldbaum, A.A. Theory of Optimal Automated Systems; PhysMatGiz: Moscow, Russia, 1963.
4. Aoki, M. Optimization of Stochastic Systems; Academic Press: New York, NY, USA; London, UK, 1967.
5. Åström, K.J. Introduction to Stochastic Control Theory; Academic Press: New York, NY, USA, 1970.
6. Bertsekas, D.P.; Shreve, S.E. Stochastic Optimal Control: The Discrete Time Case; Academic Press: New York, NY, USA, 1978.
7. Dubois, D.; Prade, H. Possibility Theory; Plenum: New York, NY, USA, 1988.
8. Zadeh, L.A. Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst. 1978, 1, 3–28. [CrossRef]
9. Kahneman, D.; Tversky, A. Prospect theory: An analysis of decision under risk. Econometrica 1979, 47, 263–292. [CrossRef]
10. Ruggeri, K.; Ali, S.; Berge, M.L.; Bertoldo, G.; Björndal, L.D.; Cortijos-Bernabeu, A.; Davison, C.; Demić, E.; Esteban-Serna, C.; Friedemann, M.; et al. Replicating patterns of prospect theory for decision under risk. Nat. Hum. Behav. 2020, 4, 622–633. [CrossRef] [PubMed]
11. Hassoun, M.; Kagan, E. On the right combination of altruism and randomness in the motion of homogeneous distributed autonomous agents. Nat. Comput. 2021. [CrossRef]
12. Kagan, E.; Rybalov, A. Subjective trusts and prospects: Some practical remarks on decision making with imperfect information. Oper. Res. Forum 2022, 3, 19. [CrossRef]
13. Kagan, E.; Rybalov, A.; Yager, R. Sum of certainties with the product of reasons: Neural network with fuzzy aggregators. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 2022, 30, 1–18. [CrossRef]
14. Yager, R.; Rybalov, A. Uninorm aggregation operators. Fuzzy Sets Syst. 1996, 80, 111–120. [CrossRef]
15. Batyrshin, I.; Kaynak, O.; Rudas, I. Fuzzy modeling based on generalized conjunction operations. IEEE Trans. Fuzzy Syst. 2002, 10, 678–683. [CrossRef]
16. Fodor, J.; Rudas, I.; Bede, B. Uninorms and absorbing norms with applications to image processing. In Proceedings of the Information Conference SISY, 4th Serbian-Hungarian Joint Symposium on Intelligent Systems, Subotica, Serbia, 29–30 September 2006; pp. 59–72.
17. Kagan, E.; Rybalov, A.; Siegelmann, H.; Yager, R. Probability-generated aggregators. *Int. J. Intell. Syst.* **2013**, *28*, 709–727. [CrossRef]
18. Apolloni, B.; Bassis, S.; Valerio, L. Training a network of mobile neurons. In Proceedings of the International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 1683–1691.
19. Gazi, V.; Passino, K.M. *Swarm Stability and Optimization*; Springer: Berlin, Germany, 2011.
20. Kagan, E.; Shvalb, N.; Ben-Gal, I. (Eds.) *Autonomous Mobile Robots and Multi-Robot Systems: Motion-Planning, Communication, and Swarming*; John Wiley & Sons: Chichester, UK, 2019.
21. Fregnac, Y. Hebbian cell ensembles. In *Encyclopedia of Cognitive Science*; Nature Publishing Group: Berlin, Germany, 2003; pp. 320–329.
22. Saeidi, H. Trust-Based Control of (Semi) Autonomous Mobile Robotic Systems. Ph.D. Thesis, Clemson University, Clemson, SC, USA, 2016.
23. Saeidi, H.; Wang, Y. Incorporating trust and self-confidence analysis in the guidance and control of (semi) autonomous mobile robotic systems. *IEEE Robot. Autom. Lett.* **2018**, *4*, 239–246. [CrossRef]
24. Tchieu, A.A.; Kanso, E.; Newton, P.K. The finite-dipole dynamical system. *Proc. R. Soc. A* **2012**, *468*, 3006–3026. [CrossRef]
25. Fodor, J.; Yager, R.; Rybalov, A. Structure of uninorms. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **1997**, *5*, 411–427. [CrossRef]
26. Klement, E.; Mesiar, R.; Pap, E. *Triangular Norms*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2000.
27. Hopfield, J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef] [PubMed]
28. Kagan, E.; Ben-Gal, I. Navigation of quantum-controlled mobile robots. In *Recent Advances in Mobile Robotics*; InTech: Rijeka, Czech Republic, 2013; pp. 311–326.
29. Jantzen, J. *Foundations of Fuzzy Control*; John Wiley & Sons: Chichester, UK, 2013.
30. Passino, K.M.; Yurkovich, S. *Fuzzy Control*; Addison-Wesley: Menlo Park, CA, USA, 1998.
31. Benioff, P. Quantum Robots and Environments. *Phys. Rev. A* **1998**, *58*, 893–904. [CrossRef]