Virtual Pheromone Based Network Flow Control For Modular Robotic Systems

Van Tung Le 1 and Trung Dung Ngo 2,*

1 Department of Electronics and Telecommunications, Dalat University, Da Lat 670000, Vietnam; tunglv@dlu.edu.vn
2 The More-Than-One Robotics Lab, University of Prince Edward Island, Charlottetown, PE C1A 4P3, Canada
* Correspondence: tngo@upei.ca; Tel.: +1-902-566-6078

Received: 22 November 2019; Accepted: 25 February 2020; Published: 14 March 2020

Abstract: Guaranteeing data transmission between modules is the key for application development of modular robotic systems. In a multi-channel modular robotic system, intersection modules play an essential role of communication channel selection in controlling data flow toward desired destinations. The gradient-based routing algorithm is an ideal solution to create an one-way communication path from any robotic module to a designated destination. To create bi-directional communication for a communication path of robotic configuration, virtual pheromone-based routing algorithm is a promising mechanism for intersection modules due to its simplicity and distributivity. In this paper, we address a virtual pheromone based network flow control based on the integration of gradient and virtual pheromone-based routing algorithms. We validated this method through an education and entertainment application using our newly developed modular robotic system.

Keywords: modular robotics; gradient-based routing; virtual pheromone-based routing; network flow control; edutainment

1. Introduction

Inspired by using natural ant pheromone in path searching and selection [1], the idea of artificial pheromone has been applied for many robotic systems [2]. Pheromone inspired stigmergy has advantages of simplicity and distributivity so this approach provides a suitable solution for robotic modules with limited computing resources. Artificial pheromone was used for robot foraging, group following, and cooperation in applications of modular robotic systems [3–10]. Light intensity is often used to mimic the level of pheromone concentration for robot coordination [7–10]. Pheromone gradient is used as a mechanism for robot foraging and coverage [3,4]. However, in complex network of robotic modular configuration in which data packets must be transferred through a desired path between modules, stigmergy communication is not sufficient as it does not guarantee the communication path over a designated time.

With the limitations of energy consumption, production cost, and hardware resources, local directional communication is preferable for modular robotics [11]. Infra-red line-of-sight communication becomes a popular communication mechanism of modular robotics because it provides not only the mean for data transmission but also the directional information of physical communication channels that is important for neighbourhood discovery & awareness, link maintenance of neighbour-to-neighbour communication, and routing protocols in a network of physically connected robotic modules. Utilizing the directional information of physical connection between neighbourhood modules, network data flow control can be solely built-up by the local directional communication mechanism.
Bi-directional data flows play the key role in developing applications of modular robotic systems. In a multi-path network of robotic configuration, sending data packets to undesired destinations could lead to unexpected results, thus a communication and routing protocol for inter-communication between robotic modules becomes essential. A number of communication protocols have been designed under the inspiration of pheromone-based communication of ant colony [12–16]. The main objective of these protocols is to create communication paths connecting robotic modules through a mobile ad-hoc network (MANET). However, these protocols require substantial communication hardware resources, thus it is impossibly applied for modular robotic systems using limited communication and computing resources, e.g., infra-red line-of-sight communication mechanism. Indeed, directional channel selection and correct path construction are challenges of modular robotic systems with limited resources that cannot handle complex communication mechanisms.

In this paper, to tackle challenges of path formation and selection of bi-directional communication paths in modular robotic networks, we propose a virtual pheromone-based network flow control that only relies on line-of-sight local communication. This control mechanism is based on the integration of gradient-based and virtual pheromone-based routing algorithms on the foundation of our distributed communication protocol presented in [17]. We formulate the research problem through discussion of technical challenges of establishing communication paths in modular robotic systems in Section 2. In Section 3, we describe our developed virtual pheromone based network data flow control algorithm. Real experiments through an education and entertainment game on our newly developed modular robotic system are addressed in Section 4. Section 5 draws conclusions and future work.

Related Work

Both global and local communication protocols have its own limitation in these existing modular robotic systems. Some types of hybrid communication was exploited but these approaches bring complexity to the hardware infrastructure. As a result, broadcasting becomes the most popular method for data transmission between robotic modules [18]. To illustrate this fact, we provide a comparison of the key properties of communication methods of the existing modular robotic systems including our system as seen in Table 1. We see that application-oriented design is the main research of such systems while communication is only considered as a simple data transmission protocol. Controlling data flow in a given configuration was only studied in a few systems but these routing protocols are relatively simple. We consider these approaches as the top-down design approach, focusing on application rather than system level. Meanwhile, we believe that, in order to enable scalability and flexibility of a modular robotic system for numerous desired applications, we must pay particular attention on a robust communication protocol that is capable of dealing with data flow in a given configuration, which can be also considered as the bottom-up design approach of modular robotic systems.

The idea of virtual pheromone based communication path selection with low computational complexity has been becoming more popular in mobile ad-hoc networks. However, this research direction has been not mature, especially in modular robotics. One typical example can be found with using virtual ants for path construction in a robotic network [19]. The communication path is used for transmitting information between food sources and a nest by mimicking the ant foraging phenomenon. However, virtual ants are sent between the nest and food sources with a fixed rate and a specific packet is required to update the network topology during operation, leading to conflict between virtual ants at the intersection and overused communication resources.

In [20], we have implemented a simplified virtual pheromone technique for information foraging to reducing the excessive virtual ants and updating signal in modular robotics. We developed the virtual pheromone inspired algorithm in which the nest sends out virtual ants, so-called forward ants, and wait for return ants from the food source. All robotic modules along the communication path randomly select communication direction without neighbourhood information. Once virtual ants reach food sources, they return to the nest and accumulate the virtual pheromone concentration on the modules along the path. If a forward ant is lost on the way back, the virtual pheromone concentration
Electronics 2020, 9, 481

is reduced by the effect of danger warning parameter. A communication path can be established with high time-consuming cost. In addition, neighbourhood awareness and link maintenance are neglected in [19,20]. Therefore, although they demonstrated the advantages of the virtual pheromone techniques in robotic networks, it is not realistically practical for modular robotic systems.

Table 1. A comparison of communication aspects of modular robotics.

| System Name             | Controller | Communication | Routing Protocol | Structure | Transmission Medium | Year |
|-------------------------|------------|---------------|------------------|-----------|---------------------|------|
| LEGO Mindstorm [21]     | Centralized| Global        | N/A              | 3D        | Physical            | 1998 |
| Triangle [22]           | Centralized| Local         | Gradient         | 2D        | Physical            | 1998 |
| Electronic block [23]   | Distributed| Local         | N/A              | 2D        | Physical            | 2001 |
| Modular Artefact [24]   | Distributed| Local         | N/A              | 2D        | Physical            | 2004 |
| Active Cube [25]        | Centralized| Global        | Serial RS485     | 3D        | Physical            | 2004 |
| Topobo [26]             | Centralized| Global        | N/A              | 3D        | Physical            | 2004 |
| Glume [27]              | Distributed| Local         | N/A              | 3D        | Capacitive          | 2006 |
| YaMoR [28]              | Centralized| Global        | N/A              | 2D        | Wireless            | 2006 |
| roBlocks [29]           | Distributed| Global        | N/A              | 3D        | Physical            | 2006 |
| Spelling bee [30]       | Centralized| Global        | N/A              | 2D        | Physical            | 2007 |
| Senspectra [31]         | Centralized| Global        | Gradient         | 3D        | Physical            | 2007 |
| IBlocks [32]            | Distributed| Local         | N/A              | 3D        | Infrared            | 2008 |
| Audiocubes [33]         | Centralized| Local         | N/A              | 3D        | Infrared            | 2008 |
| Posey [34]              | Centralized| Local         | N/A              | 2D        | Infrared            | 2008 |
| Molecubes [35]          | Hybrid     | Hybrid        | Serial RS232     | 3D        | Physical            | 2008 |
| Playware [36]           | Distributed| Hybrid        | N/A              | 2D        | Infrared            | 2009 |
| iCube [37]              | Centralized| Global        | N/A              | 3D        | Capacitive          | 2012 |
| Fable [38]              | Centralized| Hybrid        | N/A              | 3D        | Wireless            | 2013 |
| moreBot [17]            | Distributed| Local         | Gradient         | 3D        | Infrared            | 2016 |

2. Problem Formulation

Selecting a proper communication path is critical for application development of modular robotic systems. A typical example of multiple communication paths joining at an intersection in a modular robotic system can be seen in Figure 1, where the intersection plays the key role in the network flow control for inter-communication between modules. A complex communication protocol with a memorized routing table or a master-slave mode cannot be applied for modular robotics because physical connections of robotic modules are rapidly reconfigured. A distributed and simplified communication protocol could be an ideal approach to solve this challenge.

In our previous work [17], we designed a distributed communication protocol for modular robotic systems. The protocol was developed in the four-layer hierarchical architecture: physical layer,
link access and management layer, transport layer, and application layer (refer to Figure 2 and detailed explanation in [17]). The physical layer deals with neighbourhood communication between modules. This layer was developed on the hardware characteristics of communication mechanism including data transferring medium, transmission rate, and signal frame. The link access and management layer is responsible for discovering, establishing, and maintaining bidirectional communication links between physically connected neighbours. The transport layer is capable of managing data flow through a network of inter-connected directional communication links. The application layer provides a platform for application development.

Figure 1. Gradient-based routing method: A root is the lowest gradient and data packets are transmitted by following gradient descent from any module to the root, but not vice versa.

The four-layer protocol governs communication functionalities dealing with user-demanded reconfiguration of modular robotics. However, this protocol does not include a routing protocol capable of handling inter-communication of reconfigurable robotic systems for several potential applications. We implemented a gradient-based routing algorithm as a simple but effective communication path construction method for data delivery between modules [39]. We employed “integer” gradient as temporary identifications (ID) for robotic modules. All the modules have no gradients at the beginning but they can be automatically assigned a gradient value in order when any module is activated with the first gradient. Once a designated module is activated as a root by an user, it is initiated with the lowest integer gradient. Gradient of the other modules is automatically increased when the root spreads out its gradient to immediate neighbours. If a module receives more than one gradient from its neighbours, it only selects and processes the lowest one. Eventually, every module in the network has its own gradient, thus a shortest communication path is automatically formed by all the modules following the gradient descent to the root. An example of a network of robotic modules with the gradient-based ID can be seen in Figure 1. Note that some robots might have the same gradient because they belong to different communication paths.

Using the gradient-based routing as shown in Figure 1, there exist three typical intercommunication models between the root and other modules: (1) one-to-many; (2) many-to-one; (3) one-to-one. In the one-to-many model, the root sends data packets to other modules by a broadcasting communication protocol in which data packets can be relayed and transferred by following the gradient ascent. In the many-to-one model, the root receives data packets following the gradient descent from any other modules. In the one-to-one model, data packets are transmitted to the root by following the gradient descent, similar to the many-to-one model, but cannot be transferred from the root to any other modules by following the gradient ascent because there are more than one communication paths with the same gradient ascent. That is, if the root plays the role as a “sensor or “brain” in the “sense-think-act” model of a robotic system, its commands may not properly reach a “brain” for data processing or an “actuator” for actions. Indeed, an adaptive routing algorithm for a bidirectional communication path in a network of physically connected robotic modules is required to develop practical applications.
In the one-to-one model, the gradient-based routing is only capable of generating an one-way communication path from any module to the root, but not from the root to other modules. The intersection module plays a crucial role in relaying and forwarding data packets received from the root to the right directional neighbour of the desired route. However, this module encounters difficulty if it has more than one neighbours with the same gradient. We see two possible technical solutions for the intersection module to handle this issue: broadcasting data packets to all the directional channels and randomly selecting a directional channel to send data packets. The former is a typical case of the one-to-all model in which data packets are forwarded to all the other modules in the network. However, an encoding method must be used to encrypt data packets and communication resource is overloaded, which is not realistic due to limited computing power and communication bandwidth of modular robotic systems. The later relies on randomness of channel selection so it also requires encryption of data packets thus encountering the limitation of overloading bandwidth and missing data packets.

Indeed, both the broadcasting and random selection methods are not ideal solutions for generating a directional communication path from the root to any module in the one-to-one model. In order to control data flow on a desired bidirectional communication path, we need to develop a communication mechanism that is simple enough to be executed on robotic modules, especially on intersection modules, but capable of directing data packets to desired destination at the highest possibility. Inspired from the natural pheromone of ant colonies and our observation on user’s behaviours in modular robotic systems, we propose virtual pheromone-based network flow control for transmitting data packets from the root to any module. We believe that a combination of gradient-based and virtual pheromone-based routing methods is a complete solution for creating a bidirectional data communication path from the root to another module in modular robotics.

3. Virtual Pheromone Based Network Flow Control

This virtual pheromone based network flow control is a combination of gradient-based and virtual pheromone-based routing algorithms. The gradient-based routing forms an one-way directional communication path from any module to the root. Meanwhile, the virtual pheromone-based routing acts in association with the gradient-based routing algorithm in controlling data flow from the root to any module. Note that both gradient and virtual pheromone-based routing algorithms must be executed on the foundation of the physical layer and link access & link management layer in the distributed communication protocol [17]. and their probability of data communication is conditionally constrained with the probability of the neighbourhood discovery & awareness, and link maintenance. We explain how the virtual pheromone based network flow control algorithm is developed as follows:

3.1. Neighbourhood Discovery & Awareness and Link Maintenance

We briefly summarize the key characteristics of the neighbourhood discovery & awareness and the link management to show the probability of transmitting data packets. More detailed information of this layer can be found in [17].

*Neighbourhood discovery & awareness:* All the modules are considered in free state at the beginning; that is, a module may have maximum N physical communication channels available for the neighbourhood discovery & awareness. They periodically send out “hello” signal through its physical channels to search for neighbours. When “echo” and “confirmation” signals have been received, the involved modules change the stage from free to busy. The channels receiving “hello” are marked passive and the other are marked as active. As a result, a vacant channel with the newly identified neighbour is updated to occupied state. The number of physical communication channels available for the neighbourhood discovery process on such a module is reduced to n, n < N. If a module does not receive an “echo” in a directional channel, it continues searching on other channels. Here, the deadline for echo to be received is set equal to three time the duration of closing interval $t_c$. 
When two robotic modules are physically connected, their directional channels are aligned for line-of-sight communication. On a robotic module, signals are only sent over a selected channel for a short period of time, hence \( P_s = 1 \). A successful neighbourhood discovery process on a module occurs when signals are correctly received. The probability for a successful neighbourhood discovery process between two free modules is

\[
P_C = P_s \times P_r \times F_{tr}
\]

\[
= \frac{1}{N} \times F_{tr} = \frac{1}{N} \times (1 - \frac{t_c}{t_{w1}})
\]  

(1)

where the closing time \( t_c \) is caused by the receiving robot when it performs the discovery process on other channels; the processing time \( t_{w1} \) is the interval between two consecutive discovery processes when the robots are in free state. The time factor \( F_{tr} \) represents the serial processing characteristic of robotic modules.

When free modules send out “hello” signal, but the connected modules might be busy searching for new neighbours on other channels that have not yet occupied, the probability for a robot in busy state to discover a robot in free state is

\[
P_C = P_s \times P_r \times F_{tr}
\]

\[
= \frac{1}{n} \times (1 - \frac{t_c}{t_{w1}})
\]  

(2)

Note that \( n = N \) when the robot is in free state as in Equation (1).

**Link maintenance**: For a robot in busy state, the closing time is added up due to the link maintenance process. The time factor becomes \( F_{tr} = F_d \times F_m = (1 - t_c / t_{w2}) \times (1 - t_c / t_{w3}) \), where the time factor \( F_d \) is used for the discovery process with \( t_{w2} \) as the processing time in busy state, and the time factor \( F_m \) is used for the maintenance process with \( t_{w3} \) as the processing time of the link maintenance process. As a result, the probability for a robot to recognize a neighbourhood module in busy state is

\[
P_C = P_s \times P_r \times F_{tr}
\]

\[
= \frac{1}{n} \times (1 - \frac{t_c}{t_{w2}}) \times (1 - \frac{t_c}{t_{w3}})
\]  

(3)

At this state, the robotic module has recognized the direction of its neighbour from the neighbourhood discovery process, thus the probability for a link maintenance signal can be received by its neighbour is calculated as:

\[
P_C = P_s \times P_r \times F_{tr}
\]

\[
= 1 \times (1 - \frac{t_c}{t_{w2}}) \times (1 - \frac{t_c}{t_{w3}})
\]  

(4)

### 3.2. Gradient-Based Routing Algorithm

After discovering to obtain neighbourhood information, we explain how the gradient-based routing algorithm is used to generate communication paths for delivery of data packets from any module to the root.

An example of a configuration of physically connected robotic modules with automatically generated gradient-based ID can be seen in Figure 1. The root is activated with the lowest gradient, i.e., zero, and the gradient is dispersed over its neighbour so the other robots receive their corresponding gradient, which is increased from the root. Some modules might have the same gradients because they belong to different paths. A gradient dispersion algorithm can be seen in Algorithm 1.
Algorithm 1 Network gradient generation

```
1: procedure GRADIENT INITIATION
2:     if module ← gradient request then
3:         root gradient generation
4:         broadcasting gradient to neighbours
5:     else if module ← gradient from neighbour then
6:         check current state
7:             if empty gradient then
8:                 process gradient
9:                 propagating gradient to neighbours
10:             else if existing gradient is greater then
11:                 update gradient
12:                 propagating gradient to neighbours
13:             else if existing gradient is smaller then
14:                 keep current gradient
15:         end if
16:     end if
17: end procedure
```

Once the gradient has been generated, it is automatically adjusted in a distributed fashion by a path recovery mechanism. If a module is manually removed from the network configuration, e.g., $B_2$ in Figure 2a, its neighbours are aware of disconnections through the link maintenance process. $B_2$ has a smaller gradient than $B_3$, so $B_3$ requests a new gradient from its remaining neighbours. Meanwhile, $B_1$ remains its existing gradient as it has smaller gradient than $B_2$. Hence, a new path is formed with the updated gradient of involved robots. On the other hand, if a new module is added to the network configuration, e.g., $B_2$, it collects all the gradients from its immediate neighbours and selects the lowest one. To let its neighbour update their gradients, this module will transmit its gradient to all of its immediate neighbours. As a result, a communication path is constructed by involved modules without a centralized control from a master. This path recovery mechanism can be applied for any distributed modular robotic systems.

While local communication between neighbourhood modules is well managed by the physical layer and the link access & management layer [17], a path recovery mechanism with its gradient recovery procedure is used to deal with reconfiguration of communication networks. An algorithm of communication path recovery can be seen in Algorithm 2.

Algorithm 2 Communication path recovery mechanism

```
1: procedure GRADIENT RECOVERY
2:     ID ← gradient
3:     start:
4:     if current state ← neighbour then
5:         send gradient to new neighbour
6:     else if lost neighbour then
7:         check current state
8:             if lost neighbour has greater gradient then
9:                 keep current gradient
10:             else if lost neighbour has smaller gradient then
11:                 clear current gradient
12:                 request gradient from other neighbours
13:         end if
14:     end if
15: end procedure
```
Figure 2. A network of robotic modules with communication paths established through the gradient algorithm: (a) Multiple communication paths with the shortest one formed between the root and other modules. (b) A new path selected when the shortest path is broken.

Without using the gradient-based routing algorithm, the probability for a data packet to be transferred from any module to the root through a communication path of \( k \) connected robotic modules by a random physical channel selection on any robotic module is:

\[
P_T = \prod_{i=1}^{k-1} P_i = \prod_{i=1}^{k-1} \frac{1}{N^2}
\]

In contrast, once the gradient generation has been successfully achieved through the network, the probability of delivering a data packet through a communication path of \( k \) connected robotic modules only relies on the probability of the neighbourhood discovery & awareness and link maintenance processes as calculated in Section 3.1. Using these processes, the physical channel matching between neighbourhood robotic modules has been successfully achieved prior to one-way directional communication path establishment, thus we can increase the probability of delivering a data packet on such a communication path of \( k \) connected robotic modules:

\[
P_T = \prod_{i=1}^{k-1} P_{Ci} = \prod_{i=1}^{k-1} F_d \times F_m
\]

On the opposite direction, data packets may not be precisely delivered from the root to any robot because the intersection modules do not know which direction it should forward such data packets through. We apply a virtual pheromone-based routing algorithm for the intersection modules to direct data packets from the root to any module as explained in the next section.
3.3. Virtual Pheromone-Based Routing Algorithm

In nature, ants lay their pheromone on the trail. Ants use pheromone as implicit communication to search for the optimal path between food sources and their nest. At the beginning, ants randomly spread their pheromone out during their searching and exploration. Ants instinctively select their paths based on the pheromone concentration—the highest probability is selected. The more ants on a path, the higher the pheromone concentration on that direction, and the higher the pheromone concentration, the higher possibility the number of ants selecting this way. There might exist multiple paths between food sources and the nest but the path with the highest pheromone concentration becomes an optimal path as it is highly selected by ants. The pheromone concentration on an existing path is automatically evaporated over time, thus ants can search for new food sources when less food available on the existing food sources. In general, number of ants travelling on a path is proportional to the pheromone concentration of such a path. Therefore, ant pheromone-based path formation and selection is done in a distributed fashion.

Inspired from ant pheromone-based path formation and selection, we develop a routing algorithm using virtual pheromone for selecting and creating communication paths in modular robotics. On a robotic module, a data packet is considered as a virtual ant when received in or sent out from a physical communication channel, thus virtual pheromone of such a virtual ant is accumulated on such a physical channel of the robotic module. The more virtual ants going through a physical channel, the higher virtual pheromone concentration accumulated on it. To maintain the balance of the overall virtual pheromone on a module, increasing pheromone concentration on one physical channel leads to decreasing pheromone concentration on other channels. Hence, virtual pheromone concentration becomes the key factor for physical channel selection on a robotic module: the higher pheromone concentration on a physical channel, the higher possibility the channel being selected as the communication path. A simple scenario of selecting a physical communication channel is illustrated in Figure 3.

Assuming a module with $N$ neighbouring modules connected on $N$ physical channels, an analytical model of virtual pheromone on a physical communication channel $i$ is composed of the following fundamental parameters:

The pheromone deposition rate $d$: A virtual ant passing through a physical communication channel $i$ leaves an amount of virtual pheromone on the channel. Consequently, the pheromone concentration $C_i$ of such a channel is adjusted:

$$C_i(t_2) = C_i(t_1) + (N - 2)d$$

(7)

with $N > 2$ is the condition for virtual pheromone to operate on each channel. The pheromone concentration of the other channels $C_j$ is re-adjusted accordingly:
The virtual pheromone concentration represents the amount of information passing through a physical communication channel of a robotic module. The higher pheromone concentration on the channel, the higher possibility the channel being selected for data communication. The probability of a physical communication channel $i^{th}$ of $N$ channels on a robotic module is selected:

$$P_i(t) = \frac{C_i(t)}{C_i(t) + \sum_{j=1, j\neq i}^{N-2} C_j(t)}$$

(12)

where $d$ and $e$ can be selected by users in different applications.

On a robotic module, the probability $P_i$ is the decision factor for which physical communication channel is selected to forward data packets on the directional communication path from the root to any module in a robotic configuration.

### 3.4. Virtual Pheromone-Based Network Flow Control

The gradient-based routing algorithm [39] is capable of generating one-way directional communication path from any module to a root in a network of physically connected robotic modules. The probability of transmitting data packets from any module to the root by following the gradient decent is guaranteed at the probability of the neighbourhood discovery & awareness and link maintenance process, which is mostly based on the user’s behaviours. However, data packets cannot be precisely delivered from the root to any module in the network by following the gradient ascent because the intersection modules may not identify the proper physical communication channel for such packets. The virtual pheromone mechanism can be used as a probabilistic switching for the directional channel selection, especially for the intersection modules responsible for relaying data packets to proper communication channels in the direction from the root to other modules.

The probability of forwarding data packets in the direction from the root to any module is equal on all the physical directional channels at the initial stage:
\[ P_i = \frac{1}{N-1} \]  

where \( N - 1 \) channels are available because one channel is used to receive data packets from the root.

The virtual pheromone adjustment mechanism on the intersection modules starts differentiating the level of virtual pheromone concentration, so the concentration level on a channel \( i \) receiving data packets is adjusted by Equation (10) and the probabilistic switching of the channel \( i \) calculated by Equation (12). The virtual pheromone concentration plays a key role as the decision making factor whether this channel is selected for forwarding data packets from the root to any module.

In summary, the gradient-based routing algorithm is used to create any-to-root communication paths while the virtual pheromone-based routing algorithm is applied to form root-to-any communication paths. A combination of the gradient and virtual pheromone-based routing algorithms is a complete solution of generating bi-directional communication paths for the network flow control in modular robotics. Hence, we define this method as the virtual pheromone based network flow control.

4. Experiments and Results

4.1. Hardware Platform

Our robotic modules, named moreBot, are designed for research and edutainment purposes. The module is 7 cm cubic shape, fabricated by a 3D printer using ABS plastic material. On each face, four pairs of magnet cylinders are used to secure the physical connection between modules. A moreBot module, so called building block, is equipped with several sensors and actuators, e.g., an accelerometer, light detectors, microphones, RGB LEDs, a buzzer and differential wheels. On the face’s center, a pair of infra-red transceiver is employed for line-of-sight communication and distance ranging measurement. Each moreBot is governed by a XMEGA128A3U micro-controller running at the frequency of 32 MHz. Thanks to the Universal Synchronous Asynchronous Receiver Transmitters (USART) built-in function, each pair of infra-red transceiver is directly managed by an USART port for independent directional communication. The interface with a computer is done via USB or RF device. The building blocks can be programmed through the standard PDI or the boot-loader via the USB port. One moreBot block is powered by four AA batteries. A set of moreBot can be seen in Figure 4.

![Figure 4. moreBot—an affordable multi-modal modular robotic platform for education and entertainment.](image)

4.2. Preliminary Results

Neighbourhood discovery & awareness: In this section, we demonstrate the neighbourhood discovery & awareness and link maintenance processes between the building blocks.
In general, robotic modules gather information of its immediate neighbours to establish the neighbourhood communication between them. We used color from the RGB LEDs to indicate the number of physical connections or communication channels of the building blocks. Blinking *Red* indicates unconnected blocks while steady *Blue, Green, Cyan, Yellow, Magenta or White* indicate number of neighbourhood connections from 1 to 6, respectively.

To illustrate the effectiveness of the neighbourhood discovery & awareness, a random morphology is constructed in Figure 5. At the initial stage as shown in Figure 5a, five blocks were in *free* state while they were blinking in *Red*. The robotic blocks recognized its neighbours only when they were physically connected. As soon as the block discovered a neighbour, its color changes accordingly to indicate the current state. As depicted in Figure 5b, those blocks were in steady *Blue* because they had one immediate neighbour while the middle one displayed *Green* as it recognized the second neighbour. At the third stage, we joined two isolated chains of building blocks together to make an united network with three terminals as seen in Figure 5c. One block turned to *Cyan* because it had found three neighbours. Finally, when we disconnected one block as shown in Figure 5d, the isolated block turned to blinking *Red* because it lost all neighbours while its former neighbour turned from *Cyan* to *Green* as it only had two neighbours.

![Figure 5](image)

*Figure 5.* Demonstration of the neighbourhood awareness process. (a) Blocks are in free state, blinking in *Red*. (b) Different number of neighbours is shown in different colors. (c) Joining two chains of blocks. (d) Separation of a block, block states are updated and color is changed accordingly.

In the physical layer, the standard data frame 8N1 consisting of 8-bit data, non-parity, and 1 stop bit is used. We temporarily excluded the error checking function in this experiment. Table 2 shows parameter setting of the communication process.

| Communication Parameters | Value      |
|--------------------------|------------|
| USART Baud-rate          | 19,200     |
| Closing time $t_c$       | 1.8 ms     |
| Processing time (free state) $t_{w1}$ | 300 ms    |
| Processing time (busy state) $t_{w2}$ | 700 ms    |
| Processing time (link maintenance) $t_{w3}$ | 1100 ms   |
In the second layer, we set \( t_{w1} \) to 300 ms for robotic modules in free state. The \( t_{w1} \) is the average time between two consecutive discovery signals, which is also a processing time for a channel. A discovery signal is completely transmitted within 1.8 ms, which is the closing time \( t_c \). We configured the modules to send only one signal, equivalent to 0.6% of \( t_{w1} \) for reducing energy consumption. The probability for modules in free state to discovery its neighbours with a single discovery signal is

\[
P_{D1} = P_s \times P_r \times F_{tr} = \frac{1}{N} \times (1 - \frac{t_c}{t_{w1}}) = \frac{1}{6} \times (1 - 1.8/300) = \frac{1}{6} \times 0.994 \approx 0.167
\]  

(14)

When a robotic module transmits a signal, its receiving channels must be closed to avoid infra-red reflection and interference. Thus the time it needs to complete sending a signal is also its closing time \( t_c \). Beside, all the robots are identical so \( t_c \) is similar to all the modules.

The neighbourhood discovery process continues with the unoccupied channels of the building blocks at greater time interval \( t_{w2} \) to maximize the time slot for data communication and minimize the power consumption. On the established channels, the link maintenance process is activated with the processing time \( t_{w3} \). We have the probability for the robots in free state to discover a topology of connected modules as:

\[
P_{D2} = P_s \times P_r \times F_{tr} = \frac{1}{N} \times (1 - \frac{t_c}{t_{w2}}) \times (1 - \frac{t_c}{t_{w3}}) = \frac{1}{6} \times (1 - 1.8/700) \times (1 - 1.8/1100) = \frac{1}{6} \times 0.997 \times 0.998 \approx 0.995
\]  

(15)

The closing time is short as compared to the processing time, thus its effect on the overall result is unnoticeable. Reducing the processing time leads to increasing the chance to discover neighbours but also increasing the energy consumption. In opposition, increasing processing time might affect the time response of the robotic system to end users.

In the link maintenance process, the probability for a maintenance signal successfully received by its connected neighbour is calculated:

\[
P_M = P_s \times P_r \times F_t = 1 \times (1 - \frac{t_c}{t_{w2}}) \times (1 - \frac{t_c}{t_{w3}}) = 0.997 \times 0.998 \approx 0.995
\]  

(16)

We applied the common hand shake technique for local neighbour-to-neighbour communication process. If the receiver \( B \) does not involve in other communication process, the probability for receiving a data packet from the sender \( A \) is

\[
P_C = P_A \times P_B \times F_t = 1 \times (1 - \frac{t_c}{t_{w2}}) \times (1 - \frac{t_c}{t_{w3}}) = 0.997 \times 0.998 \approx 0.995
\]  

(17)

Even both the neighbourhood discovery and link maintenance active during the communication process, neighbour-to-neighbour data transmission still remains at the high possibility.
Gradient-based routing for any-to-root communication path: We designed an edutainment application to illustrate the path formation based on the gradient-based routing algorithm. The Phonic Game is developed to help children studying language through tangible programming by replacing passive wood blocks by the moreBot, which provides an innovative teaching and learning method.

moreBot are converted into the alphabet character building blocks. We embedded five different characters to five faces of each block. Using the accelerometer to identify the top face, the character on the top face is selected to represent the building block in the Phonic Game. We also designed one special block with a loud speaker to play the recorded soundtracks. The Phonic Game is used to examine the path formation process by the following demonstrations.

Transferring data from any module to the root: We proved that the gradient-based routing provides a reliable method for transferring data packets from any module to the root. The first mode of the Phonic Game is about the word generation. We started the word generation game by plugging the speaker module at one end of the robotic configuration. The block was activated by the user to become the root, so it broadcasted the gradient to its neighbours sequentially. The gradient recovery mechanism helped all the building blocks adjust their gradients whenever the network morphology was reconfigured. We utilized the infra-red ranging function on the top face only for user-block interaction, preventing interference of infra-red signals on the other faces.

In general, the further the building blocks counted from the root, the higher the gradient-based ID, as shown in Figure 6. When the user interacted with a building block, the representing character i.e. ‘a’ or ‘b’ was embedded into the data packet sent to the neighbours towards the root by following the gradient descent. Along the communication path, the representing characters of the other building blocks were also inserted into the data packets. Hence the final packet reaching the root contained a string of characters of all the building blocks in the communication path in order of the gradient descent counted from the block activated by the user. The speaker module verified this string whether it was a correct word or not in reference to the embedded dictionary. If a meaningful word was found, the speaker played the corresponding recorded soundtrack, e.g., ‘strong’ or ‘study’. Note that the position of every character was unique in a correct word, thus the speaker module only recognized the word and spoken it out if and only if the gradient-based routing algorithm guaranteed the precision of data collection and packaging on the route from the block initialized by the user to the root by using the any-to-root directional communication path.

Transferring data from root to any module: The second mode of Phonic Game is about spelling. When a correct word was checked, the speaker module should spell each character of the word in order along with lighting up the corresponding module of such character. Therefore, the building blocks involved in the word must be able to relay and deliver the commands from the root to other blocks correctly and timely. A root-to-any directional communication path is expected to build up for this assignment. However, the gradient mechanism could create the same gradient-based ID to
different modules. Therefore, data packets smoothly flew back to the root from different communication paths but did not correctly flow from the root to expected modules because the intersection module, character $t$ in this example, did not know which the directional communication channel should be selected. Therefore, we demonstrate how the virtual pheromone-based routing algorithm is capable of solving this issue in the next section.

4.3. Virtual Pheromone-Based Routing Algorithm

To validate the virtual pheromone-based routing algorithm on root-to-any directional communication path, we set up a scenario as shown in Figure 7. A network of building blocks was formed by three paths representing three words: s-t-r-o-n-g, s-t-u-d-y, and s-t-a-r. Using the gradient generation mechanism, three any-to-root communication paths were established and the module $t$ acted as the intersection module. This module played the key role in transmitting data packets from the root to any module using the virtual pheromone concentration mechanism.

Figure 7. A snapshot of Phonic Game with a robotic configuration of three branches forming three different words, i.e., study, strong and star.

In the spelling mode, characters on the modules must be correctly lighted up in order, corresponding to spelling sound. For instance, once the word study was selected by the user, the lighting commands should reach the character $s, t, u, d$ and $y$ synchronously to sound it on the speaker module. Because the blocks $s$ and $t$ were on the unique path from the root, they received spelling commands without concerning the other directions. The intersection block $t$ had to select the correct direction to relay the lighting commands from the root to the blocks $u, d$ and $y$, consequently. However, the same gradient were assigned to the blocks $u, r, a$, so broadcasting spelling commands for the word study could cause unexpected blocks $r$ and $a$ to light up with the probability of random directional channel selection:

$$P_i = \frac{1}{N-1} = \frac{1}{3}$$ (18)

To solve this issue, we applied the virtual pheromone based path selection for the intersection block. Initially, the virtual pheromone concentration on all the communication channels were equal, thus we took the probability of channel selection $P = 1/3$ as the baseline for the probabilistic switching of the intersection block. In this case, the module $t$ equally sent data packets to all the directions at the initial stage. When a user interacted with a building block i.e., $y, g$, or $r$ of a word i.e., study, strong or star while playing the spelling game, data packets transmitted from the building block to the root through a physical communication channel on the module $t$. The virtual pheromone concentration on such a channel was increased with an amount of pheromone deposition while the virtual pheromone concentration on the other channels were decreased accordingly so that the total amount of the virtual pheromone on module $t$ was unchanged (100%). The more the user interacted with a building
block, the more data packets transmitted to the root on the channel due to its pheromone probability. As a result, when the root sent spelling commands through the communication path of the selected word, e.g. study, the building block $t$ chose the directional physical channel towards the building block $u$ based on the probability of the virtual pheromone concentration accumulated on this channel. Note that the virtual pheromone concentration on such a channel was automatically decreased toward the baseline by the evaporation parameter over time if the user stopped interacting with the building block. The virtual pheromone parameters are shown in Table 3.

Table 3. Virtual pheromone parameters.

| Parameters               | Value               |
|-------------------------|---------------------|
| Maximum ant rate        | 1 ants/second       |
| Deposition rate $d$     | 2 unit/ant          |
| Evaporation rate $e$    | 0.5 unit/second     |
| Delay                   | 25 ms / module      |
| Concentration upper limit | 95%                |
| Concentration lower limit | 5%                 |

Note that parameters of the virtual pheromone concentration can be changed according to system requirements. High deposition rate could quickly raise the virtual pheromone concentration up to the saturation level leading to fast establishment of root-to-any communication path. High deposition rate could also increase switchability of communication paths according to users’ behaviour but might create fluctuation due to the short time reaction. In contrast, low deposition rate requires more time to form root-to-any communication paths. However, a communication path established with the low deposition rate should be expected if we wish to receive a large number of data packets from the chosen direction. Similarly, high evaporation rate enables to switch communication paths according to users’s behaviours but creates fluctuation because of the conflict between deposition and evaporation. In contrast, low evaporation rate could reduce flexibility of communication paths since the virtual pheromone concentration is still maintained even when the selected communication path is not used for a while. Selection of deposition and evaporation rates is an important issue for a specific application.

To illustrate the real time analysis of the virtual pheromone concentration, we collected data from the module $t$ and displayed through the Matlab graphical user interface (GUI). Using the basic parameters in Table 3, the virtual pheromone concentration levels on three physical communication channels was presented as in Figure 8. A video demonstration of this experiment can be found in the link (https://youtu.be/ph2a8JSnHdk).

Figure 8. Demonstration of the virtual pheromone based network data flow control in Phonic Game using the Matlab GUI. Virtual pheromone levels are real-time visualized in continuous lines and bar graphs.
One research question would be posed: *What happens if the deposition rate is not constant?* To answer this, we implemented *three deposition functions* in terms of *pseudo-linear, pseudo-quadrature, pseudo-logarithm functions*. Figure 9 shows the virtual pheromone concentration of these deposition functions measured on the same system for comparative analysis. First, we applied the pseudo-step function for the modular robotic system. Pseudo-linear function can be seen as one-step response function enabling the virtual pheromone concentration to reach the threshold with only one step.

Second, the pseudo-quadrature function slowly increased the virtual pheromone concentration by a little amount for the first few steps, then quickly got up to the maximum value. In a dynamic network where a dominant path is not clear, using the pseudo-quadrature function ensures that a desired communication path is achievable. Meanwhile, the pseudo-logarithm function showed fast reaction to the network reconfiguration. The virtual pheromone concentration quickly raised to the high level, and then slowed down to keep the channel stable when it reached the threshold. This pseudo-logarithm function is helpful for switchability of communication paths but might introduce turbulence when the distinction between paths are not clear. Hence, system performance can be adjusted according to the selected deposition rate.

**Figure 9.** Different functions of deposition rate measured at the intersection block on different physical channels using Equations (7) and (8): Range 1, 2: Pseudo-linear deposition rate creates linear growth of the virtual pheromone concentration on the selected channels. When no data packets were transmitted over this channel, the pheromone concentration was slowly decreased to the initial values by the evaporation. Range 3: Pseudo-quadrature deposition functions with slow increasing rate at the beginning. Range 4: Pseudo-logarithm deposition function has opposite rate to pseudo-quadrature function with high increasing rate at the beginning. Range 5: Pseudo-step function abruptly changed the virtual pheromone concentration, which was unlikely with natural pheromone but useful in some applications.

In particular, we found that this step function is suitable for our *Phonic* game because any request from any branches can be immediately activated due to the fast switching between communication paths at the threshold of 95%. In addition, if we set the upper threshold to 100% and the lower threshold to 0%, probability of choosing communication paths by randomness did not exist according to user’s point of view. The flexibility of system was secured by the evaporation rate because the virtual
pheromone concentration automatically went back to the balance level after a while if communication path was no longer used.

We do not consider a fixed rate in this research because our protocol is adaptable to users’ interaction. Moreover, we simplified the protocol with only two fundamental parameters: pheromone deposition and evaporation. We have taken real-world issues of the characteristics of modular robotic systems and human behaviours into account to design the virtual pheromone based protocol and successfully demonstrated it through an edutainment game. When we carried out experiments, we found that the duration between two consecutive configuration changes is a critical factor of the developed routing and communication protocol. Since it requires a certain interval for a robotic module to accumulate or evaporate its virtual pheromone, the robotic configuration must be unchanged during such a duration. With the Phonic game, we examined that it usually took 2–3 s for the virtual pheromone on the intersection to converge to a stable rate without any local minimum or maximum. The duration is quite reasonable and acceptable for development of phonic games in the real world.

5. Conclusions

We presented the novel virtual pheromone based network data flow control for modular robotic systems. We proved the effectiveness of gradient and virtual pheromone-based routing algorithms in dealing with communication path selection of reconfigurable robotic systems through an edutainment game. We believe that this virtual pheromone based routing protocol is sufficient enough for building-up and maintaining bidirectional communication paths in many modular robotic systems. In the future, on the one hand, we will examine the developed algorithm with more complex configuration of our modular robotic system to find out its possible limitations. In addition, we will extend our investigation of this algorithm with an auto-adjustable parameter selection mechanism. On the other hand, we have planned to focus more on developing new education and entertainment games to bring our technological advantages to the society.

Author Contributions: V.T.L. designed the robotic building blocks and implemented the communication protocol. T.D.N. designed the research methodology and managed the whole project. Both contributed to the experiments, data collection and presentation, and manuscript.

Funding: We acknowledge the support of the Natural Sciences and Engineering Council of Canada, (NSERC RGPIN-2017-05446) and IDEAS program.

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

References

1. Goss, S.; Aron, S.; Deneubourg, J.L.; Pasteels, J.M. Self-organized shortcuts in the Argentine ant. Naturwissenschaften 1989, 76, 579–581. [CrossRef]
2. Alers, S.; Tuyls, K.; Ranjbar-Sahraei, B.; Claes, D.; Weiss, G. Insect-inspired robot coordination: Foraging and coverage. Artif. Life 2014, 14, 761–768.
3. Hrolenok, B.; Luke, S.; Sullivan, K.; Vo, C. Collaborative foraging using beacons. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 3-Volume 3. International Foundation for Autonomous Agents and Multiagent Systems, Toronto, ON, Canada, 10–14 May 2010; pp. 1197–1204.
4. Russell, K.; Schader, M.; Andrea, K.; Luke, S. Swarm robot foraging with wireless sensor motes. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, Istanbul, Turkey, 4–8 May 2015; pp. 287–295.
5. Purnamadjaja, A.H.; Russell, R.A. Guiding robots’ behaviors using pheromone communication. Auton. Robot. 2007, 23, 113–130. [CrossRef]
6. Fujisawa, R.; Imamura, H.; Hashimoto, T.; Matsuno, F. Communication using pheromone field for multiple robots. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 1391–1396.
7. Payton, D.; Estkowski, R.; Howard, M. Pheromone robotics and the logic of virtual pheromones. In *International Workshop on Swarm Robotics*; Springer: Berlin, Germany, 2004; pp. 45–57.

8. Garnier, S.; Tache, F.; Combe, M.; Grimal, A.; Therault, G. Alice in pheromone land: An experimental setup for the study of ant-like robots. In Proceedings of the 2007 IEEE Swarm Intelligence Symposium, Honolulu, HI, USA, 1–5 April 2007; pp. 37–44.

9. Mayet, R.; Roberz, J.; Schmickl, T.; Crailsheim, K. Antbots: A feasible visual emulation of pheromone trails for swarm robots. In *International Conference on Swarm Intelligence*; Springer: Berlin, Germany, 2010; pp. 84–94.

10. Arvin, F.; Krajník, T.; Turgut, A.E.; Yue, S. COSΦ: Artificial pheromone system for robotic swarms research. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 407–412.

11. Ahmadzadeh, H.; Masehian, E.; Asadpour, M. Modular Robotic Systems: Characteristics and Applications. *J. Intell. Robot. Syst.* **2016**, *81*, 317–357. [CrossRef]

12. Gunes, M.; Sorges, U.; Bouazizi, I. ARA-the ant-colony based routing algorithm for MANETs. In Proceedings of the International Conference on Parallel Processing Workshops, Vancouver, BC, Canada, 21 August 2002; pp. 79–85.

13. Baras, J.S.; Mehta, H. A probabilistic emergent routing algorithm for mobile ad hoc networks. In Proceedings of the WiOpt’03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Sophia-Antipolis, France, 3–5 March 2003.

14. Agüero, C.; Cañas, J.M.; Ortuno, M.; Matellán, V. Design and implementation of an ad-hoc routing protocol for mobile robots. *Turk. J. Electr. Eng. Comput. Sci.* **2007**, *15*, 307–320.

15. Di Caro, G.A.; Ducatelle, F.; Gambardella, L.M. Wireless communications for distributed navigation in robot swarms. In *Workshops on Applications of Evolutionary Computation*; Springer: Berlin, Germany, 2009, pp. 21–30.

16. Kambayashi, Y. A review of routing protocols based on ant-like mobile agents. *Algorithms* **2013**, *6*, 442–456. [CrossRef]

17. Le, V.T.; Ngo, T.D. A Distributed Communication Protocol for Modular Robotic Systems. In Proceedings of the 2016 IEEE International Conference on Advanced Intelligent Mechatronics, Banff, AB, Canada, 12–15 July 2016.

18. Seo, J.; Paik, J.; Yim, M. Modular Reconfigurable Robotics. *Annu. Rev. Control. Robot. Auton. Syst.* **2019**, *2*, 63–88. [CrossRef]

19. Campo, A.; Gutiérrez, Á.; Nouyan, S.; Pinciroli, C.; Longchamp, V.; Garnier, S.; Dorigo, M. Artificial pheromone for path selection by a foraging swarm of robots. *Biol. Cybern.* **2010**, *103*, 339–352. [CrossRef] [PubMed]

20. Le, V.T.; Ngo, T.D. Virtual pheromone based information foraging in modular robotics. In Proceedings of the 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Kuala Lumpur, Malaysia, 12–15 November 2014; pp. 379–384.

21. Klassner, F.; Anderson, S.D. Lego MindStorms: Not just for K-12 anymore. *IEEE Robot. Autom. Mag.* **2003**, *10*, 12–18. [CrossRef]

22. Gorbet, M.G.; Orth, M.; Ishii, H. Triangles: Tangible interface for manipulation and exploration of digital information topography. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Los Angeles, CA, USA, 18–23 April 1998; pp. 49–56.

23. Wyeth, P.; Wyeth, G.F. Electronic blocks: Tangible programming elements for preschoolers. In *IFIP TC. 13 International Conference on Human-Computer Interaction*; IOP Press: Lausanne, Switzerland, 2001; Volume 1, pp. 496–503.

24. Ngo, T.D.; Lund, H.H. Modular artefacts. In Proceedings of the Component-Oriented Approaches to Context-aware Computing (ECOOP 2004), Oslo, Norway, 14–18 June 2014.

25. Watanabe, R.; Itoh, Y.; Asai, M.; Kitamura, Y.; Kishino, F.; Kikuchi, H. The soul of ActiveCube: Implementing a flexible, multimodal, three-dimensional spatial tangible interface. *Comput. Entertain. (CIE)* **2004**, *2*, 15. [CrossRef]

26. Raffle, H.S.; Parkes, A.J.; Ishii, H. Topobo: A constructive assembly system with kinetic memory. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vienna, Austria, 24–29 April 2004; pp. 647–654.
27. Parkes, A.; LeClerc, V.; Ishii, H. Glume: Exploring materiality in a soft augmented modular modeling system. In Proceedings of the CHI’06 Extended Abstracts on Human Factors in Computing Systems, Montreal, QC, Canada, 22–27 April 2006; pp. 1211–1216.

28. Möckel, R.; Jaquier, C.; Drapel, K.; Dittrich, E.; Upegui, A.; Ijspeert, A. YaMoR and Bluemove—An autonomous modular robot with bluetooth interface for exploring adaptive locomotion. In Climbing and Walking Robots; Springer: Berlin, Germany, 2006; pp. 685–692.

29. Schweikardt, E.; Gross, M.D. roBlocks: A robotic construction kit for mathematics and science education. In Proceedings of the 8th International Conference on Multimodal Interfaces, Banff, AB, Canada, 2–4 November 2006; pp. 72–75.

30. Dekel, A.; Yavne, G.; Ben-Tov, E.; Roschak, Y. The spelling bee: An augmented physical block system that knows how to spell. In Proceedings of the International Conference on Advances in Computer Entertainment Technology, Salzburg, Austria, 13–15 June 2007; pp. 212–215.

31. LeClerc, V.; Parkes, A.; Ishii, H. Senspectra: A computationally augmented physical modeling toolkit for sensing and visualization of structural strain. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 28 April–3 May 2007; pp. 801–804.

32. Nielsen, J.; Lund, H.H. Modular robotics as a tool for education and entertainment. Comput. Hum. Behav. 2008, 24, 234–248. [CrossRef]

33. Schiettecatte, B.; Vanderdonckt, J. AudioCubes: A distributed cube tangible interface based on interaction range for sound design. In Proceedings of the 2nd International Conference on Tangible and Embedded Interaction, Bonn, Germany, 18–20 February 2008; pp. 3–10.

34. Weller, M.P.; Do, E.Y.L.; Gross, M.D. Posey: Instrumenting a poseable hub and strut construction toy. In Proceedings of the 2nd International Conference on Tangible and Embedded Interaction, Bonn, Germany, 18–20 February 2008; pp. 39–46.

35. Zykov, V.; William, P.; Lassabe, N.; Lipson, H. Molecubes extended: Diversifying capabilities of open-source modular robotics. In Proceedings of the IROS-2008 Self-Reconfigurable Robotics Workshop, Nice, France, 22–26 September 2008.

36. Lund, H.H.; Marti, P. Designing modular robotic playware. In Proceedings of the International Symposium on Robot and Human Interactive Communication, Toyama, Japan, 27 September–2 October 2009; pp. 115–121.

37. Goh, W.B.; Kasun, L.; Fitriani, Tan, J.; Shou, W. The i-Cube: Design considerations for block-based digital manipulatives and their applications. In Proceedings of the Designing Interactive Systems Conference, Newcastle, UK, 12–15 June 2012; pp. 398–407.

38. Pacheco, M.; Moghadam, M.; Magnusson, A.; Silverman, B.; Lund, H.H.; Christensen, D.J. Fable: Design of a modular robotic playware platform. In Proceedings of the 2013 IEEE International Conference on IEEE Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 544–550.

39. Le, V.T.; Ngo, T.D. Gradient-Based Routing Protocol for Modular Robotic Systems. In Proceedings of the 2020 IEEE/SICE International Symposium on System Integration, Honolulu, HI, USA, 12–15 January 2020.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).