A Near-Optimal Dynamic Power Sharing Scheme for Self-Reconfigurable Modular Robots

Chi-An Chen, Thomas Collins, Wei-Min Shen

Abstract— **This paper proposes a dynamic and near-optimal power sharing mechanism for self-reconfigurable modular robots that successfully extends the operating time of sets of connected modules. The proposed method achieves near-optimal results even when each module only knows the power status of its immediate neighbors (those to which it is directly connected) rather than the power status of every module in the robotic system. The proposed method is validated in physics-based simulation environments and will be implemented on real robotic hardware developed at the Polymorphic Robotics Laboratory at the University of Southern California. It is also compared with current state-of-the-art power sharing mechanisms. Simulated results show that the proposed method allows for longer operation time than with alternative state-of-the-art methods.**

I. INTRODUCTION

Effective power allocation is essential in distributed robotic systems. An effective dynamic power sharing scheme delays the failure of modules with low power, allowing the entire system to function for longer periods of time. Maximizing operation time is vital in many autonomous robotic applications, including search and rescue and space robotic systems.

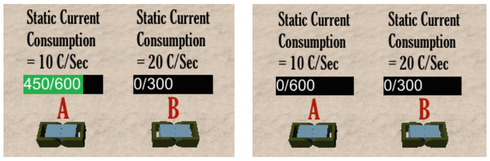
However, implementing such a mechanism is a difficult problem. In general, each module in the system may have a different power level and different power requirements, depending on the configuration and the task being performed. Mathematically speaking, dynamically finding an optimal or near-optimal distribution of power to the modules that extends their collective operation time is a challenging optimization problem that must be performed almost continuously during the operation of the robotic system.

Most of the time, in distributed robotic systems, it is extremely challenging to realize this theoretical maximum operation time. This paper proposes a dynamic, near-optimal power sharing technique, which, given a set of connected modular robots with different power statuses, can effectively approximate the theoretical maximum operation time through power sharing between each member in the self-reconfigurable robotic system by using only local communication (that is, communication only between directly connected modules). In this method, each modular robot calculates a local remaining charge capacity average and local current consumption average using its own status information and that of its direct neighbor modules. This local information is used as an estimate of global battery capacity average and global power consumption average. These estimates determine the module power sharing decisions necessary for the system to operate near its theoretical maximum operation time.

This work first develops some important theory related to optimal power sharing in self-reconfigurable robots (Section III). This investigation paves the way for the proposed method, which uses local information to share power in a near-optimal, dynamic way (Section IV). Section V provides an evaluation of the proposed method as well as comparisons with current state-of-the-art power sharing methods in the literature. Section VI concludes and details plans for future work. Related work is discussed next.

II. RELATED WORK

Energy optimization techniques have been investigated in robotic systems to improve and maximize a host of functions, for example optimal foot placement locations [1] and task specific manipulator motions [2]. In distributed robotic systems, the importance of energy has been described in various aspects. For example, in order to achieve a self-maintained mechanism for cooperative robots, a battery exchange algorithm based on a probabilistic model has been described and verified in [3]. Two optimal energy sharing polices for self-reconfigurable robotic systems were introduced in pioneering work done by Dr. Raja Humza Qadir [4] [5]. In the first power sharing policy (Policy 1), each robotic module stores an energy distribution table which records its state of charge (SOC) and its direct neighbor's SOC. Each module initiates power sharing with a neighbor automatically if that neighbor's energy is lower than the energy sharing threshold (E_{th}) and the energy of the module itself is larger than E_{th}. In contrast, in power sharing Policy 2, each modular robot not only keeps its direct neighbor's SOC, but also records its direct neighbor's direct neighbor's SOC. Again, power sharing is initiated automatically if the module has energy more than E_{th} and the direct neighbor or their direct neighbor's energy has energy less than E_{th}. The proposed method considers global energy information which is essential for determining which modules have low power. This is done using local information. With this information, the modules can share power more precisely and reduce power waste due to the redundant power sharing that occurs in less informed methods. This paper compares static power sharing (where the donor is always the most energetic one in the system), Policy 1 [4] [5], Policy 2 [4] [5], and the proposed method under a variety of energy distribution scenarios and a variety of mechanical load distributions. These methods are also compared against a baseline mechanism in which power is not shared at all. In all cases, operating time is measured, providing a means of comparing the different methods.



(a) No Power Sharing Scenario.

(b) Optimal Power Sharing Scenario.

Fig. 1: No Power Sharing Scenario v.s. Optimal Power Sharing Scenario.

III. OPTIMAL POWER SHARING THEORY

A. *Theoretical Maximum Operation Time*

Determining the theoretical maximum operation time for a robotic system is interesting in its own right as well as essential to the proposed method. To obtain the theoretical maximum operation time, one must model the power consumption behavior of the system. For modeling the remaining charge capacity of the battery, one can take the unit of electric charge: Coulomb (Symbol: C) as the representation. For example, if the battery capacity is 1000 mAh (milliampere hour), then the battery has $3600\,C$. If the current consumption is $1\,C$ per second, then the estimated operation time is around 1 hour.

Consider Figure 1(a) as an example. It shows two modules: module A and module B. The "health" bar shows the battery capacity of module A originally had $600\,C$, but at the end of 15 seconds of operation, $450\,C$ are left; meanwhile, module B had $300\,C$ originally and $0\,C$ left after 15 seconds of operation. Module A still has some electric charge left which it could potentially share with module B. Therefore, in this scenario, the two robot system shown did not reach its maximum possible operation time. In contrast, if the modules are able to do perfectly optimal power sharing, the result will be the scenario shown in Figure 1(b) because the total energy has been utilized to keep the two robot system alive and the total remaining charge is equal to zero at the end. In this case, the system reaches its theoretical maximum operation time of 30 seconds.

Following is how to calculate the theoretical maximum operation time (MOT) of the whole system:

$$MOT = \frac{\sum_i RCC_i}{\sum_i CC_i}\,, \tag{1}$$

where RCC_i represents Remaining Charge Capacity of module i and CC_i represents Current Consumption rate of module i. In the scenario of Figure 1(b), the theoretical maximum operation time of the module A and B system can be written as:

$$MOT_{AB} = \frac{RCC_{Module\,A} + RCC_{Module\,B}}{CC_{Module\,A} + CC_{Module\,B}} \tag{2}$$

The result is 30 seconds which is equivalent to the operation time in Figure 1(b).

B. *Local Information*

Consider performing this calculation in a distributed environment made of self-reconfigurable robotic modules. In order to calculate the theoretical maximum operation time of such a system based on equation (1), each module in that system would have to retrieve the battery status and current consumption rate of every other robot module in the system, a slow process that is unreasonable for a dynamic, online power sharing mechanism. Instead, the proposed method approximates the maximum operation time of the robotic system at each module using local information. Each module exchanges messages with its direct neighbor(s) (those to which it is directly connected) to acquire a local average RCC of battery and local average current consumption rate (including its own battery level and current consumption rate). These averages are then used in equation (1) to estimate the maximum operation time of the system. This task is performed at each module at some pre-determined frequency (e.g., every 1/10 of a second). As the number of time steps this procedure is repeated increases, these estimates converge (usually quite quickly) to the true maximum operation time calculated using equation (1). Following is an example of estimating the maximum operation time of a robotic system with three connected modules, module 0, module 1, and module 2 as illustrated in Figure 2. Only the calculation of the average RCC of module 1 is shown. The other averages at different modules would be analogous.

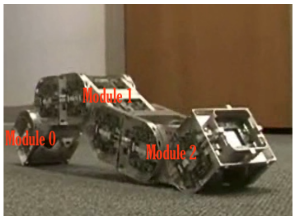


Fig. 2: Caterpillar Configuration Formed by Three SuperBot Modules.

$$\begin{aligned}
RCC^0_{1,avg} &= RCC^0_1,\\
RCC^1_{1,avg} &= \frac{RCC^0_{0,avg} + RCC^0_{1,avg} + RCC^0_{2,avg}}{3},\\
RCC^2_{1,avg} &= \frac{RCC^1_{0,avg} + RCC^1_{1,avg} + RCC^1_{2,avg}}{3},\\
&\vdots\\
RCC^{n+1}_{1,avg} &= \frac{RCC^n_{1,avg} + \sum_{j \in the\,neighbor\,nodes} RCC^n_{j,avg}}{3},
\end{aligned} \tag{3}$$

Generalizing the example described above, the following derives the general equation for the local average RCC battery estimate which modular robot i holds after n time steps:

$$RCC^{n+1}_{i,avg} = \frac{RCC^n_{i,avg} + \sum_{j \in the\,neighbor\,nodes} RCC^n_{j,avg}}{K}, \tag{4}$$

where index i indicates the ith modular robot and K is the head count including the ith modular robot and its direct neighbor robot(s). Notice that the calculation of average current consumption rate is completely analogous.

C. Supply and Demand Determination

Once the modules have estimated the average global RCC of battery and average current consumption rate (as described above), they are able to estimate the maximum operation time easily (see equation (1)). Once the estimated maximum operation time is obtained, each module uses its current RCC status to determine if it should act as a donor, receiver or passthrough module.

In order to make the system reach the estimated maximum operation time, the proposed method needs to derive the expected remaining charge capacity for each module from equation (5). If the current estimated RCC of battery is larger than the expected RCC which is equal to the maximum operation steps (estimated with local information) times the estimated average current consumption rate, the module will apply equation (6) to determine the amount of power to donate and act as a donor. Otherwise, the module will act as a receiver and use equation (7) to decide how much it will sink. If the difference is sufficiently close to zero, the module will act as a passthrough module, meaning the module just acts as one part of power bus and lets the power from the donator pass through itself to the receivers in the system.

$$\begin{aligned} Max\ Steps &= \frac{RCC_{expected}}{CC_{avg}}, \\ \Rightarrow RCC_{expected} &= Max\ Steps \times CC_{avg}. \end{aligned} \tag{5}$$

If $RCC_{current}$ is larger than $RCC_{expected}$,

$$D.C.A. = RCC_{current} - Max\ Steps \times CC_{avg}. \tag{6}$$

If $RCC_{current}$ is smaller than $RCC_{expected}$,

$$R.C.A. = Max\ Steps \times CC_{avg} - RCC_{current}. \tag{7}$$

D.C.A. and R.C.A. denote Donated Charge Amount and Received Charge Amount respectively. CC_{avg} represents average Current Consumption rate and RCC represents Remaining Charge Capacity.

D. Donor Selection

In order to prevent conflicting power sharing decisions, each time power sharing is to occur, the collective modular robots need to elect a donor before sharing power. Conflicting power sharing means more than one module is trying to donate their power to the receivers, which might lead the power to module(s) with sufficient energy. Therefore, in order to prevent redundant power sharing, it would be essential to elect one donor in a certain time frame. For determining the donor in the system, we have adopted the Distributed Task Negotiation in [6].

Based on section III-B and III-C, once a module decides locally that it is to act as a donor, it will start to broadcast task messages (TM) [6] to its direct neighbors. When a leaf node module receives task message from its direct neighbor, it will create a "child-of" relationship to the sender (the parent node) and send back an ack message to its parent node. Otherwise, the module is at intermediate node and will broadcast the received task message and also create a "child-of" relationship to the sender (parent node).

In the event of a conflict, i.e. when a module receives more than one task message, the module will choose one donor with relatively stronger power and then broadcast the newRoot message (NRM) [6] for the new sender to its other direct neighbors. Meanwhile, the module will delete all previous "child-of" relationships and create a new "child-of" relationship between itself and the new sender.

Once the module receives ack messages from all its child nodes, it will send an ack message to its parent node. Through this process, one donor will collect all ack messages from its child nodes and be elected in a certain time frame and start to share its power to the receiver(s). Figure 3 shows the task generated by P6 is elected through this process. The election of a single donor prevents conflicting power choices from damaging the operation time of the system.

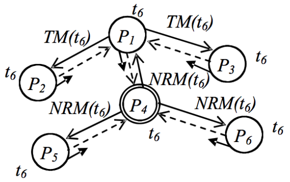


Fig. 3: Distributed Task Negotiation [6].

IV. NEAR-OPTIMAL DYNAMIC POWER SHARING

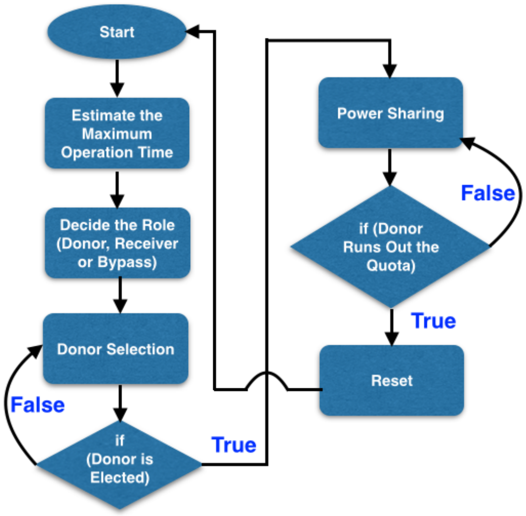


Fig. 4: Near-Optimal Power Sharing Algorithm

A. Software Design

In this section, the proposed near-optimal power sharing scheme is detailed.

As shown in Figure 4, the algorithm begins by having each module estimate the maximum theoretical operating

Algorithm 1: Near-Optimal Dynamic Power Sharing Algorithm

```
Input: None
Output: None
while true do
    if ($RCC_{neighbors} - RCC_{itself} > Threshold_{RCC} \,||\, CC_{neighbors} - CC_{itself} > Threshold_{CC}$) then
        1. update the local average of the battery capacity and the current consumption.
    else
        1. estimate the theorectic operation time.
        2. based on the theorectic operation time to decide the role.
    if ($role\, is\, decided$) then
        if ($leader\, is\, not\, elected$) then
            1. elect the leader in the period.
        else if ($leader\, is\, elected$) then
            1. power sharing.
            if ($leader\, runs\, out\, the\, budget$) then
                1. reset the local information.
                2. reset the role.
                3. reset the time.
```

time of the system using the local communication described in section III-B. After acquiring an estimate of the system's maximum possible operation time, every module determines its role: to act as a donor, a receiver or a passthrough module as discussed in III-C. If the module is to act as a donor, it will broadcast a message to its direct neighbors to express that it wants to be a candidate donor in the current time step. The receivers and the passthrough modules will receive the messages from these candidate donors and a single donor will be chosen as described in III-D. In the process of donor selection, the receivers and passthrough modules will decide how to react to the messages depending on their positions in the system. If the module is a "leaf" module (connected to only one other module), once the message is received, it will reply with an ack message to its only neighbor (parent) node immediately. If it is not a "leaf", the module will wait until all ack messages are collected and then send an ack message back to its parent node. Once a donor is nominated, the rest of the modules will choose whether to close or open their switches in order to share power. Based on equation (6) or (7), each donor and receiver knows the amount of supplying or receiving to do in each time step. The time step ends when the required amount of power is shared. The modules in the system will keep updating their estimates of the global average battery power and global average power consumption, leading to more and more precise estimates and driving the whole system near the theoretical maximum operation time. This process is described in the pseudo code in Algorithm 1, where, e.g., $Threshold_{RCC}$ represents the threshold of the remaining charge capacity difference between the robot itself and its direct neighbor(s). $Threshold_{CC}$ is defined analogously. Both values should be set very small, e.g. < 1.

B. Hardware Design

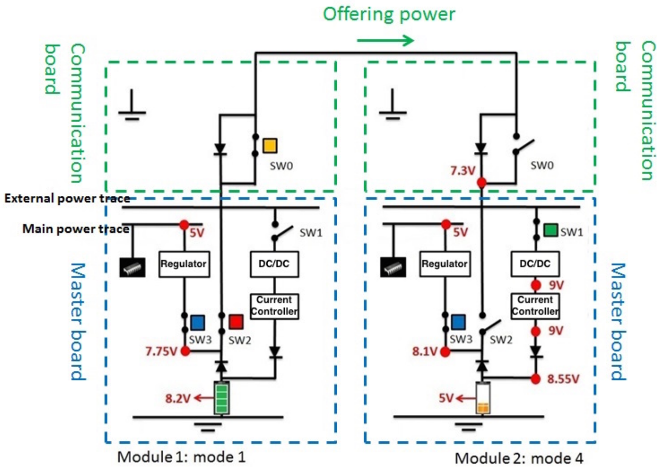


Fig. 5: Power Sharing Simplified Circuit for SuperBot [7]

Figure 5 shows a simplified circuit of the power sharing mechanism on SuperBot [7]. The power sharing function block on each robot includes a DC/DC converter and current controller. When the modular robots share power, the power flow has to go through DC/DC converter. However, the efficiency of the regular DC/DC converter is only about 80% to 90%, so it is important to reduce the number of times that power sharing occurs in a real-world implementation. This will help reduce any loss resulting from the power sharing procedure itself.

V. SIMULATION AND EVALUATION

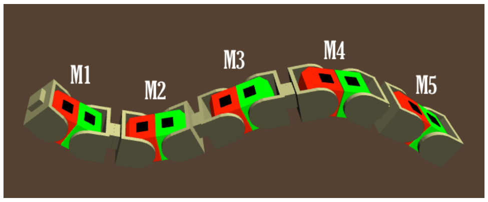
Before implementing the proposed algorithm into hardware, a number of simulations were performed using the Unity game engine[1]. This was both to verify the logic of the proposed method and to compare it with existing power sharing schemes. In the simulation, the efficiency of the DC/DC converter has also been taken into consideration.

Several SuperBot modules were created, arranged and connected as a "caterpillar" (simple serial chain) shape as illustrated in Figure 6(a). For the simulation in Unity, as shown in Figure 6(b), the green bar indicates the module's health, i.e. remaining charge capacity. Energy transfer is visualized as a red bullet shooting from module(s) to module(s) to better understand the power sharing process. Once any module depletes its power completely, the entire system is considered to have stopped functioning. Once this has occurred, the power sharing operations that occurred during the simulated run are recorded.

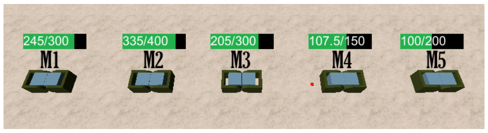
Several other power sharing strategies were implemented (including [4] [5]) in in this simulation environment. The numerical results of the comparison of all these methods (including the proposed method, labeled Near Optimal) are

---

[1] https://unity3d.com

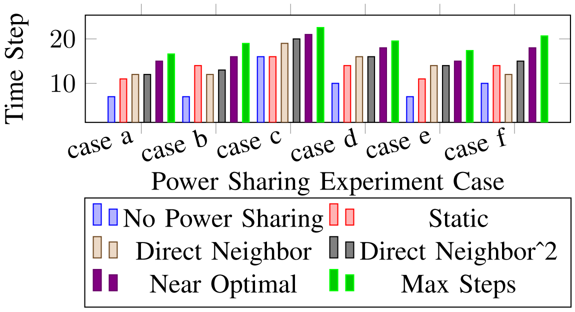


(a) Caterpillar Shape Formed By 5 SuperBot Modules.

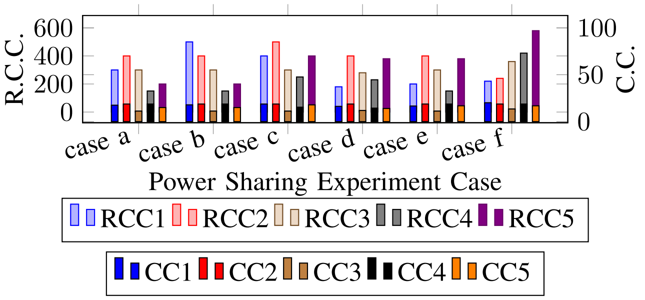


(b) Power Sharing Simulation in the Caterpillar Configuration.

Fig. 6: Power Sharing Simulation for SuperBot Modules in the Caterpillar Configuration.

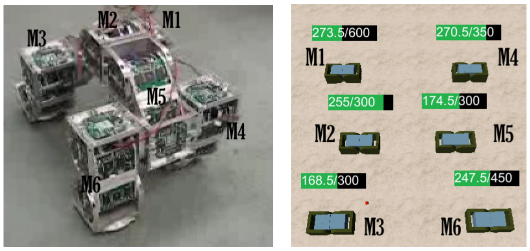


(a) Time Step Results for the Selected Power Sharing Schemes.



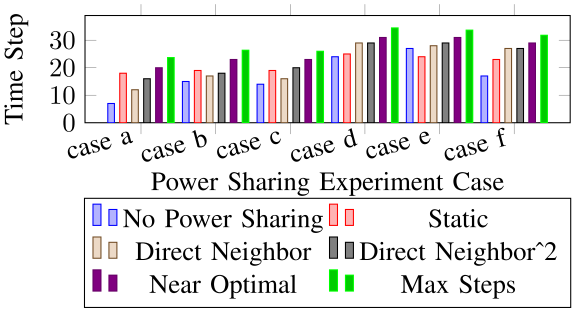
(b) Battery Capacity (B.C.) and Current Consumption (C.C.) of the Modules.

Fig. 7: Power Sharing Simulation Numerical Results for Caterpillar Configuration.

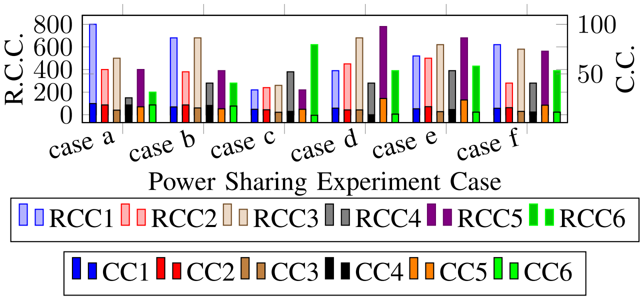


(a) SuperBot in Quadruped Configuration. (b) Quadruped Walker Power Sharing Simulation.

Fig. 8: Power Sharing Simulation for SuperBot Modules in the Quadruped Shape.



(a) Time Step Results for the Selected Power Sharing Schemes.



(b) Battery Capacity (B.C.) and Current Consumption (C.C.) of the Modules.

Fig. 9: Power Sharing Simulation Numerical Results for Quadruped Configuration.

given in Figure 7(a). 6 kinds of cases (a to f) were randomly generated for comparing the selected power sharing schemes. In each case, modules have different (randomly assigned) remaining charge capacity (R.C.C.) values and current consumption (C.C.) values as listed in Figure 7(b), where, e.g., RCC1 and CC1 represent the remaining charge capacity and current consumption for module 1. According to the results shown in Figure 7(a), the number of time steps of the proposed method is always closest to the theoretical maximum operation time. This is particularly true in cases a, b and f, where energy distribution is very uneven.

In another experiment, power sharing in a quadruped configuration of SuperBot modules was simulated. The quadruped configuration implemented in real hardware is illustrated in Figure 8(a). The power sharing simulation is illustrated in Figure 8(b). With the same concept as shown in Figure 7, we also have created 6 kinds of random cases (a to f). The results of this experiment, given in Figure 9 (a), also shows that the proposed near-optimal power sharing scheme is superior to alternative methods in terms of increasing the operation time of the entire system. In general, the number of power sharing times of the static method is usually between 100 to 250, however, the proposed method significantly reduces the sharing times to around 40. Although the other two methods have similar power sharing times to the proposed method, the result in Figure 9(a) shows that power sharing schemes that only consider surrounding neighbors may not be sufficient to extend the operation time to near-optimal levels. Cases a, b and c as shown in Figure 9 (b), in which the power on the limbs (RCC1, RCC3, RCC5, RCC6) is much different than the power on the body (RCC2, RCC4), demonstrates the proposed method can consistently extend the operation time to near the maximum possible values.

In a final set of experiments, the proposed method was validated against a baseline method (no power sharing) in the physics-based self-reconfigurable simulator ReMod3D [8]. Both a rolling track configuration of six SuperBot modules and a quadruped configuration of five SuperBot modules (as in Figure 8(a)) locomoting on a flat surface were simulated. In each simulation, two systems of modules, one that performed no power sharing and one that used the proposed near-optimal power sharing mechanism, performed the same gait starting at the same location. Screenshots of the rolling track simulations are shown in Figure 10. The initial battery levels and consumption rates were set in the program. The computation and message passing were distributed as it would be on real hardware. The color of module indicates how much battery it has remaining. High battery values correspond to green, while low battery values correspond to red. When the modules all turn white, at least one module has failed completely (indicating total system failure), and the modules "power down".

In these final experiments, the proposed near-optimal power sharing scheme always resulted in the robotic system traveling farther than without power sharing. The difference in distance depended, of course, on the initial battery levels at each module and the consumption rates set in the program.

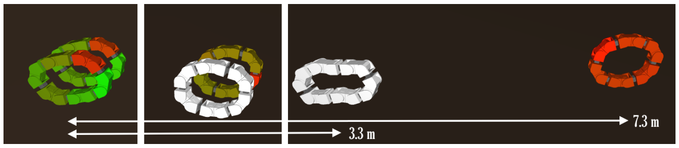


Fig. 10: Rolling track simulation in ReMod3D. On the left, we see the initial state of the simulation. The battery levels of each module are different, as illustrated by the colors (green means high battery, red low battery). The middle image shows the point at which the robot system that does not use power sharing (front) fails. The image on the right shows the near-optimal power sharing rolling track locomoting far past the one that does not use power sharing.

VI. CONCLUSION

This paper introduced a novel, near-optimal, dynamic power-sharing mechanism that outperformed current state-of-the-art power sharing methods in a number of numerical and physics-based simulations. The key to the method's success is its use of local approximations to reduce computational load on the robotic modules. These local approximations grow closer and closer to the true global values they approximate the longer the method is run. This method eliminates redundant power sharing and effectively extends the operation time of the cooperative robotic system to its theoretical maximum value.

VII. FUTURE WORK

The power sharing scheme discussed above will be applied to SuperBot hardware. There are still some challenges which need to be overcome.

A. Battery Power and Power Consumption Information

Firstly, in hardware circuits, we can easily detect the voltage level, however, sometimes the voltage level doesn't reflect the real remaining charge capacity. In simulation, we can assume remaining charge capacity and the current consumption of each modular robot are given, but it would be difficult to detect the remaining charge capacity of each modular robot in real time. Secondly, a relatively good estimation of the power consumption of each modular robot would be required in order to implement this near-optimal power sharing scheme.

B. Group Power Sharing

In our simulation, once a module is elected as a donor, it can share power to any module in the system. However, if the system includes enormous numbers of modules, it might not be practical for certain donors to share power to certain receivers over a long distance. One reason for this is the diode involved in sharing power. When power passes through one module, the voltage level drops around 0.5 volts. Thus, if the donor has a fully charged 8 volt battery and its power passes through 15 modules, the voltage remaining is only 0.5 volts, outside the voltage range of the DC/DC converter inside SuperBot modules. In addition, passing power through multiple diodes increases power waste. Consequently, consolidating the power sharing area might make the proposed scheme more practical. Better hardware circuits may be also considered to reduce the loss of power transformation through passthrough modules.

C. Eliminating Communication Between Modules

The communication between modules involved in this method might consume a certain amount of power (not currently modeled in simulation). Reducing the communication between modules in this method will have a direct impact on the actual running time of a physical system using our proposed mechanism.

REFERENCES

[1] L. Barrios and W.-M. Shen, "Phase space planning and optimization of foot placements in rough planar terrains," Seattle, WA, May 2015.

[2] O. Maimon, E. Profeta, and S. Singer, "Energy analysis of robot task motions," *Journal of Intelligent and Robotic Systems*, vol. 4, no. 2, pp. 175–198, 1991. [Online]. Available: http://dx.doi.org/10.1007/BF00440418

[3] T. D. Ngo and H. Schioler, "Sociable mobile robots through self-maintained energy," in *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, vol. 3, Oct 2006, pp. 2012–2017.

[4] R. H. Qadir, *Self-Sufficiency of an Autonomous Reconfigurable Modular Robotic Organism*. Springer Publishing Company, Incorporated, 2014.

[5] H. Qadir Raja and O. Scholz, "Dynamic Power Distribution and Energy Management in a Reconfigurable Multi-Robotic Organism," *ArXiv e-prints*, July 2012.

[6] B. Salemi, P. Will, and W.-M. Shen, "Autonomous discovery and functional response to topology change in self-reconfigurable robots," in *Complex Engineering Systems: Science Meets Technology*, D. Braha, A. A. Minai, and Y. Bar-Yam, Eds., 2006, pp. 364–384.

[7] C.-A. Chen, A. Kamimura, L. Barrios, and W.-M. Shen, "Dynamic power sharing for self-reconfigurable modular robots," Oxford, UK, Aug. 2013.

[8] T. Collins, N. O. Ranasinghe, and W.-M. Shen, "Remod3d: A high-performance simulator for autonomous, self-reconfigurable robots," Tokyo, Japan, Nov. 2013.