Autonomous 6D-Docking and Manipulation with Non-Stationary-Base Using Self-Reconfigurable Modular Robots

Luenin Barrios, Thomas Collins, Robert Kovac, and Wei-Min Shen

Abstract— **Aggregation of self-reconfigurable robotic modules can potentially offer many advantages for robotic locomotion and manipulation. The resulting system could be more reliable and fault-tolerant and provide the necessary flexibility for new tasks and environments. However, self-aggregation of modules is a challenging task, especially when the alignment of the docking parties in a 3D environment involves both position and orientation (6D), since the bases of docking may be non-stationary (e.g., floating in space, underwater, or moving along the ground), and the end-effectors may have accumulated uncertainties due to many dynamically-established connections between modules. This paper presents a new framework for docking in such a context and describes a solution for sensor-guided self-reconfiguration and manipulation with non-fixed bases. The main contributions of the paper include a realistic experiment setting for 6D docking where a modular manipulator is floating or rotating in space with a reaction wheel and searches and docks with a target module using vision. The movement of the docking parties is a combination of floating and manipulation, and the precision of the docking is guided by a sensor located at the tip of the docking interface. The docking itself is planned and executed by a real-time algorithm with a theoretical convergence boundary. This new framework has been tested in a high-fidelity physics-based simulator, as well as by real robotic modules based on SuperBot. Experimental results have shown an average success rate of more than 86.7 percent in a variety of different 6D-docking scenarios.**

I. Introduction

Self-reconfigurable robotic systems could potentially offer immense time and cost savings in challenging environments such as space or disaster recovery situations. Such systems can repair, extend, or alter their kinematic structures to meet the demands of new and possibly unexpected tasks and environments that are beyond the capabilities of fixed-shaped robots. By self-reconfiguring their modules, such robots could also recover from component failures and offer greater options for deployment and mission execution. Many challenges need to be overcome before this vision can be realized.

One of the key challenges is fully *autonomous docking* among self-reconfigurable modules. Specifically, how can a self-reconfigurable manipulator, which consists of many modules, position and orient its end-effector dock with the dock of a target module to physically connect the two docks together and form a larger system? In any 3D environment, such as space, the docking challenge is amplified in three major ways. First, the problem must be considered in its full 6D form: the 3D position and 3D orientation of the docks to be connected must be reconciled for successful docking. Second, the bases of the docking parties may be non-fixed (e.g., floating in space or underwater). In such situations, a movement of the robot's joints may induce a reactionary force that changes the robot's current base (or pose) substantially. Third, the connections among modules are mostly established dynamically, which may result in accumulated uncertainties at the tip of docking interface.

This paper makes three main contributions toward solving this challenging problem. It presents a realistic experimental environment for 6D docking where a modular manipulator is floating or rotating in space with a reaction wheel and searches and docks with a target module using vision. In this environment, the movement necessary for successful docking is a combination of floating and manipulation. The precision of the docking is guided by a sensor near the tip of the docking interface. The docking itself is planned and executed by a real-time algorithm with a theoretical convergence boundary. This new method has been tested in a high-fidelity simulation environment, as well as with physical robotic modules based on SuperBot [1]. The results have shown an average success rate of more than 86.7 percent in a variety of different 6D-docking scenarios.

The rest of the paper is organized as follows. Section II discusses related work in self-reconfigurable docking, including manipulation planning and control. Section III describes the 6D-docking environment and its challenges. Section IV presents the developed control algorithms for 6D-docking. Section V describes the experimental evaluation of the proposed method, both in physics-based simulation and on real hardware. Section VI concludes the paper with some potential future work.

II. Related Work

A. Self-Reconfigurable Docking

The ability for self-reconfigurable modules to connect and disconnect from one another is extremely important. Designing mechanisms and algorithms that facilitate this ability has been a focus of self-reconfigurable robot research from its earliest days. A plethora of docking hardware designs have been presented in the literature, e.g., [1]–[9], and include physical latches, teeth, and magnets, each with their own advantages and disadvantages. The SuperBot module

Luenin Barrios, Thomas Collins and Wei-Min Shen are with Polymorphic Robotics Lab, Information Sciences Institute, University of Southern California, Los Angeles, U.S.A. lueninba@usc.edu, collinst@usc.edu, and shen@isi.edu. Robert Kovac is an independent consultant for the project. This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited

docks used in our experiments have a specialized tooth-based design called SINGO [10] that allows for two modules to disconnect even if one of the modules fails completely. This fault-tolerance is extremely important in space applications.

Much work has been done in the area of autonomous self-reconfiguration docking algorithms using chain-type, lattice-type, and hybrid self-reconfigurable systems, though these algorithms have been primarily limited to 3D cases (either 3D positioning or 2D positioning with 1D orientation correction). Most studies dealing with docking of separated modules, such as those of M-TRAN [12], [13], relied on flat horizontal floors to compensate the modules' lack of dexterity and other specifically designed connector mechanisms to cover misalignment. This is due to the modules' lack of sensors or precision in actuator control for docking. In [4], Rubenstein, et. al demonstrate autonomous 3D docking between CONRO modules on the ground (2D positioning and 1-D orienting). Yim, et. al provide similar autonomous docking in [11] with additional results shown in 3D with UBot [14], mobile and swarm self-reconfigurable robotics [15]–[18], and robotic boats [19]. More recently, the M-Blocks [7] from MIT's CSAIL have demonstrated docking that could be considered 6D using an internal flywheel and magnetic connectors.

The SuperBot modules used here were designed to combine the advantages of both lattice and chain-based configurations. In this particular application, the sensor-free "blind" lattice-based docking is inapplicable because the modules must use their sensors to align and dock. Although modular and self-reconfigurable manipulators have received attention in the literature [20]–[23], to the best of our knowledge there is no related work on the use of modular self-reconfigurable hardware for precision docking in 6D. It is important to note that the 6D docking problem with a self-reconfigurable manipulator is a very difficult manipulation problem in its own right. Theoretically, the control of a self-reconfigurable or modular manipulator does not differ greatly from traditional manipulator control, but practical concerns (including quickly accumulating sensor and actuator noise, distribution of computation, etc.) render this problem extremely difficult in a real-world setting, necessitating a novel sensor-guided approach. The controller developed in this work addresses this question and provides an approach general enough for any chain-based reconfigurable robot docking based on sensors. Furthermore, by using Superbot's SINGO dock design [10], compensation for some of the errors caused by both the controller and sensor noise is achieved.

B. Non-Fixed Base Manipulation

Space robot manipulation systems with non-fixed bases have been studied in the literature (e.g., [24], [25]). These systems are more difficult than fixed-base systems in terms of dynamics and control. However, the control of self-reconfigurable manipulators without a fixed base is an open research area. The work presented in this paper may represent a first step in this direction toward controlling such self-reconfigurable systems in space, underwater, or in micro-gravity environments by considering the 6D docking scenarios where the base is allowed to freely rotate or move about one or more axes.

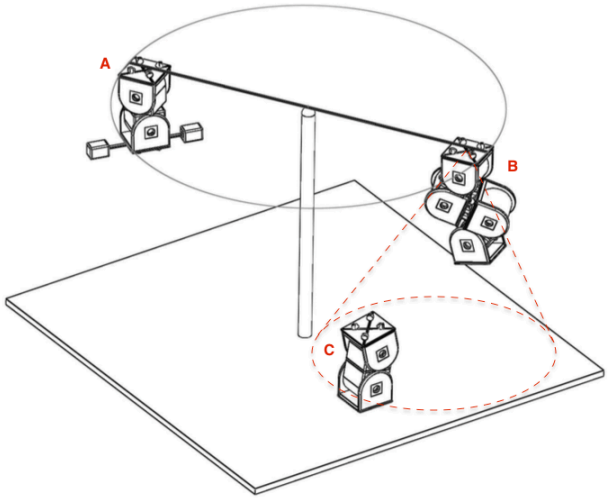

Fig. 1: A physically-simulated micro-gravity environment for 6D-docking experiments.

C. Docking Manipulation by Inverse Kinematics

Solving the inverse kinematics(IK) problem is a key step in performing any manipulation or self-docking task, especially docking with a self-reconfigurable manipulator. The dynamic number of links and potential for hyper-redundancy makes this problem especially difficult for self-reconfigurable manipulators. It is therefore of the utmost importance that we select an inverse kinematics methodology that is well suited to addressing such issues.

Classical approximation approaches to solving the IK problem, such as those based on the Jacobian, suffer from numerical issues around singularities and do not scale well with the number of DOF [26]. This makes them difficult to apply to self-reconfigurable manipulators. The limitations of Jacobian-based approximations have led to a plethora of alternative solution strategies. For a thorough analysis of the multitude of IK techniques applicable to redundant and hyper-redundant manipulators, consult [26].

Following work in [27]–[29] and inspired by the encouraging experimental results provided, we have adopted particle swarm optimization (PSO) as the core component of our inverse kinematics methodology. PSO finds high-quality solutions to the IK problem for self-reconfigurable manipulators very efficiently, according to extensive tests we have performed both in simulation and on real-world hardware.

III. 6D-Docking Experimental Setup

In order to realistically investigate and develop solutions for the 6D-docking problem, we designed and constructed a physical testing apparatus using self-reconfigurable modules in a simulated micro-gravity (space or underwater) environment. Shown in Figure 1, two robotic systems "A" and "B"

are mounted on the two ends of a nearly frictionless rotatable bar or rig to simulate the flying or floating in an orbit in space. The system A is a single SuperBot module with a reaction wheel (a dumbbell) that can spin continuously with variable speed and directions. The controlled acceleration of this reaction wheel produces a corresponding force that rotates the apparatus bar, thus causing systems A and B to fly in a circular orbit above the table. The system B consists of two SuperBot modules (3 degrees of freedom (DOF) each) and is a manipulator with 6 DOF. One end of B is mounted on the bar and the other end acts as an end-effector with a dock interface and can move freely in space and reach down or around for a target. A camera is mounted on B near the end-effector so that B can look down or around while it is flying in orbit. A target module "C" is placed freely on the table and it has a dock interface facing upward ready to be docked by B. The target also has a visual marker (called an April Tag [30]) for B to see. The objective of this testing is for the manipulator B (with the assistance of A controlling the rotation or orbit of the apparatus bar) to reach and dock with the target C. After docking, C and B will be aggregated together and become a new and larger manipulator with 9 DOF.

This experiment apparatus contains most of the challenges for 6D-docking with non-fixed bases. First, all components are made of real self-reconfigurable modules, meaning the system inherits all the challenges of self-reconfiguring modules. Second, the base of the manipulator B is easily movable or not stationary, and any movement of B's internal joints may cause a change of its center of mass and thus produce a change in its position in orbit. Furthermore, the rotation of the bar is nearly frictionless, making the control of the positions of A and B subject to momentum changes and non-holonomic. The flight in orbit can be controlled to some extent by the velocity of the reaction wheel on A, but it is common that B may either overshoot or undershoot the target. This makes the search for the target challenging, as the system must consider both its current position and momentum for effective control. This is very similar to the control of a boat on water. Once B finds that C is within its reach, its vision system must carefully identify the position and orientation of the target, plan its manipulation using inverse kinematics, and execute the planned movement to dock with the target. During the execution of its internal joint movements, B's base position may be affected, requiring B to communicate with A to fly back in position to align with C again. In short, the manipulation and floating of the system must be integrated coherently in order to achieve a successful docking with the target.

IV. 6D-Docking Control Algorithms

The control of a successful 6D docking involves several stages that must be interwoven with one another. First, the system A+B must search and find the target C. Second, once C is within reach, system B must use its vision to identify the relative position and orientation of C. Third, B must plan its manipulation using inverse kinematics to align its end-effector dock with the dock of C. Fourth, when the two docks are aligned, B must activate its dock motor to physically dock with and aggregate C. During these stages, if B notices that C disappears from the camera's field of view after a manipulation movement, then B must inform A to reactivate the search for C and resume the planning and manipulation for docking after the target is re-acquired.

Algorithm 1: 6-D Docking (FLY-SEARCH-DOCK)

Input:
$[v_{min}, v_{max}]$: the range of speed of reaction wheel
v_i : the current speed of reaction wheel, $v_i = v_{min}$
E : status of target search (∞ for no target is seen)
δ : threshold for target search ($E \leq \delta$ target centered)
v_f : speed jump to produce rotational force, v_f = 1
C: a manipulation path to dock with target

```
1  Function FLY-SEARCH-DOCK()
2      while E > δ do
3          if δ < E < ∞ (target seen, yet uncentered) then
4              v_i = v_i + v_f; // jump to produce force
5              E = TargetInVision(); // check target
6              if E > δ (still uncentered) then
7                  v_f = v_f + 1; // plan a greater force
8                  Gently reduce v_i to v_min; // prepare
9                  continue
10             else
11                 break; // target is at center
12         else
13             Gently reduce v_i to v_min; // prepare jump
14             v_i = v_max; // jump to produce force
15     C = Algorithm 2; // plan a path for 6-D dock
16     while Executing C do
17         if E > δ then
18             goto Line 2
```

The controller for 6D docking is presented in Algorithm 1. The feedback for target searching is provided by the function TargetInVision() using the camera mounted on the 6-DOF manipulator (see B in Fig 1). The vision system maintains a continuous search for a neon-green-colored fiducial marker (associated with the target) in the environment. Tracking these color blobs gives us information about which acceleration direction to produce to move the reaction wheel motor. This feedback allows us to center the end effector over the target marker (Line 2-14), as well as reposition the base of the manipulator should it move during its manipulation to dock with the target (Line 15-18).

In simulation, it is sufficient to use a simple proportional derivative (PD) controller to position the end-effector over the fiducial marker. We simply use the distance between the center of the image and the center of the color blob picked up by the camera as the error measure. In the hardware environment, however, such a controller is too simple due

to the varying and unpredictable friction of the rotating bar. Additionally, the testing table is not perfectly level giving rise to a search space with valleys that require greater force to navigate. Thus, different amounts of acceleration are needed at different points in orbit in order to swing the rig around the appropriate amount. If the correct acceleration is not given, the rig may rotate opposite of the intended direction, negating any progress made during the reaction-wheel acceleration.

In order to overcome this challenge, we developed a closed-loop acceleration controller (see Line 7 in Algorithm 1) that gradually increases acceleration (i.e., v_f for jump in speed) until it detects that visual progress has been made ($E \leq \delta$). This is accomplished by iteratively performing larger and larger jumps. A jump is defined as an instantaneous change in velocity from speed v_i to v_j and may be either upward or downward. Note that upward and downward jumps allow us to produce both positive and negative accelerations. Therefore it is sufficient to spin the reaction wheel in only one direction to achieve bi-directional control of the rig. These sudden velocity changes produce the accelerations, thus forces, needed to move the rig the desired amount in a rough order (similar to row a boat on water). Furthermore, we take advantage of the small yet noticeable friction inherent in the freely rotating joint at the center of the bar to ensure that backward slippage is avoided. This is achieved by gradually or gently decrementing (or incrementing, depending on the direction of motion) the velocity of the rotation wheel such that the position of the rig is stabilized first before continuing with the next speed. This procedure prepares for the next jump without disturbing the current position of the bar. This prevents backwards slippage and allows us to control the rotational direction of the manipulator.

In essence, the camera reads the x,y position of the blob generated by the colored fiducial marker in the camera image. The image x-direction-which is perpendicular to the rotating bar-offset (from the image center) of the blob is used as the error measure and the feedback to the reaction wheel acceleration controller. If the color blob is not in view of the camera, we use the reaction-wheel to rotate the rig in a default direction (Line 14 in Algorithm 1) while keeping our camera on in case the marker is seen. Once the marker is seen, the image is used to determine the acceleration amount (positive or negative) needed to produce the motion of the rig that centers the blob (Lines 4-9 in Algorithm 1). Alignment progress with the visual marker is done by gradually increasing the acceleration through jumping of the motor speed to induce larger and larger torques at the freely rotating joint until sufficient progress is made. After each jump, the reaction wheel speed is gradually and gently reduced, first ensuring stability of position at each speed value, and then continuing with further reductions in speed until the reaction wheel speed has reached its base value. This process repeats itself until the target blob is centered in the camera image. Once the blob marker has been centered within the error margin ($\leq \delta$), the framework proceeds with manipulation of the arm by calling Algorithm 2.

A. Manipulation to Dock

Algorithm 2 is for manipulating the end-effector of the flying robot to dock with the target. It is called when the target is centered at the visual field of the 6-DOF manipulator so that it can proceed to reach and dock with the target. This algorithm first identifies the relative position and orientation of the target with respect to its current base, and then uses particle swarm optimization to solve the inverse kinematics problem to align its end-effector to dock with the target.

Algorithm 2: 6-D Docking (Manipulation)

Input:
F : fitness function, n : number of dimensions (manipulator DOF), m : number of particles, N : max iterations, h : quality threshold, W : W-space target point
Output:
C: solution to IK problem for W

```
1  Function PSOIK()
2      numIts := 0;
3      bestFitness := MAX_FLOAT_VALUE;
4      qualityMet := false;
5      Initialize m particles x_i uniformly in search space;
6      foreach i := 1...m do
7          Initialize m best particles p_i := x_i;
8          if F(p_i) < bestFitness then
9              C := p_i;
10             bestFitness := F(C);
11     while numIts < N and !qualityMet do
12         foreach i := 1...m do
13             foreach j := 1...n do
14                 update element j of particle i using
                   basic PSO update equations;
15                 Clamp particle back into search space, if
                   necessary;
16             fitOld := F(p_i);
17             fitNew := F(x_i);
18             if fitNew < fitOld then
19                 p_i := x_i;
20                 if fitNew < bestFitness then
21                     C := x_i;
22                     bestFitness := fitNew;
23                     if bestFitness <h then
24                         qualityMet := true;
25         numIts++;
26     return C;
```

Particle swarm optimization is a swarm-based metaheuristic optimization algorithm that has been demonstrated to be quite effective in solving difficult and high-dimensional optimization problems in a number of diverse domains [31], [32]. The basic idea of PSO is that a swarm of m particles, each n-dimensional, performs an independent search in the

space of possible n-dimensional solutions while exchanging information with one another to accelerate the search. Each particle i is a point x_i in this search space with a certain velocity v_i and has an associated fitness given by the objective function (F) value at that point $F(x_i)$. Particles move around in this search space randomly but sampling is biased toward a random weighted average of the best position achieved by any particle in the swarm g and the best position achieved by each particle individually, p_i. This focuses random searches on areas of the search space where a global optimum is expected to be.

To apply PSO to IK problems for any serial manipulator with any number of DOF n, our algorithm PSOIK searches directly in the space of possible manipulator joint angles for an optimal set of joint angles, and uses Equation 1 as a fitness (objective) function to be minimized.

$$F(\mathbf{q}) = w_p \cdot posError + w_o \cdot orientError \qquad (1)$$

This function F maps a set of joint angles in $\mathbb{R}^n$ to a real number, which is the fitness of joint angles $\mathbf{q}$. $posError$ is the magnitude of the position error between the end-effector pose corresponding to $\mathbf{q}$ (given by the forward kinematics model) and the target end-effector pose. $orientError$ is the magnitude of the vector of Euler angles representing the rotational difference between the pose of the end-effector corresponding to $\mathbf{q}$ and the target end-effector pose. w_p and w_o are nonnegative weights trading off the relative importance of each error term. It is evident that the optimal set of joint angles has a fitness function value of 0. Thus, the closer a particle's fitness is to 0, the better that set of joint angles is. Algorithm 2 illustrates the steps necessary to apply PSO to the inverse kinematics problem for a serial manipulator with an arbitrary number of DOF n. Algorithm 2 uses Equation 1 as the PSO fitness function F. The experiments conducted here used 1000 particles with 500 iterations. For a more detailed discussion of the application of PSO to IK problems see [33].

PSO is a metaheuristic that is often chosen for solving difficult optimization problems in because of its efficiency and ease of implementation. Recently, a theoretical proof of its convergence has become possible by combining PSO with some known converging framework such as branch and bound. Using these properties, we could create a globally convergent IK solution for self-reconfigurable manipulators of any size that is guaranteed to find a solution in a finite amount of time (provided a solution exists and the algorithm is given an error tolerance that is strictly greater than zero).

V. Experimental Evaluations and Analysis

To evaluate the performance of the framework presented above, we have designed and implemented four categories of test experiments C1-4 as shown in Fig 2.

- Category C-1 includes the test cases where the target module is directly beneath the manipulator arm and its dock is normal to the end-effector dock of the arm. Distances between the two docks are arranged to be either near (25mm) and far (100mm) in different runs.

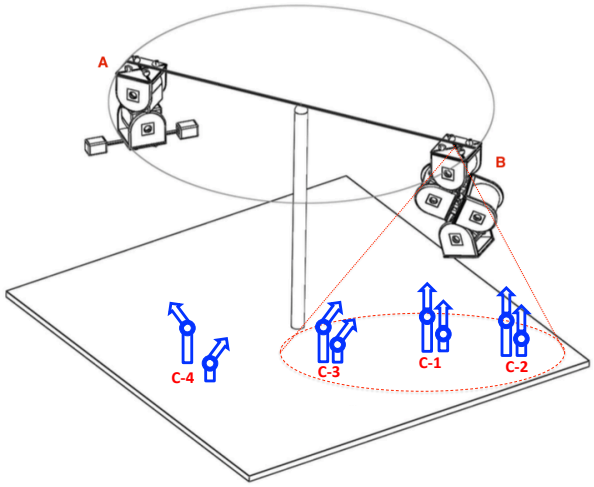

Fig. 2: Four experimental test categories C1-4.

- Category C-2 are the test cases similar to C-1, except the target module's dock is no longer directly beneath the manipulator arm but displaced by half a module distance either to the left or right of the end-effector.
- Category C-3 includes test cases where the target dock lies on the reachable working sphere in the workspace of the end-effector. The target dock angle was approximately 25 degrees from the normal of the plane intersecting the sphere as shown in Fig 2. Note that both lateral and sagittal cases on the spherical workspace were tested and that any excessive push will cause the target module to fall down on the table. This eliminates any advantage arising from simply pushing down toward the table.
- Category C-4 include the test cases where the target is randomly placed on the table with positions and orientation from C-1 through C-3. This category requires the robots A and B to perform the complete circular search and manipulation for a successful 6-D docking.

We have experimented these four categories in both high-fidelity physics-based simulation [34], as well as in physical real-world using the SuperBot self-reconfigurable modules.

In simulation, we have modeled the rotational bar/rig with the reaction wheel, 6-DOF manipulator, frictionless joint, and target module and tested our manipulation algorithms in a noise-free environment shown in Fig. 3(left). We have performed 10 runs in each category, and the results have shown a 100 percent successful rate in 6D docking. Although these runs are noise-free, they validated the theoretical correctness of the proposed control algorithm.

The hardware experiments are more challenging due to the inherent noise and uncertainties of the hardware. Fig. 3(right) shows the hardware experimental setup. The SuperBot module labeled "Reaction Wheel" produces the controlled force for the rotational bar, while at the other end of the bar, two connected SuperBot modules are connected as a 6-DOF manipulator with a docking interface at the tip of the arm.

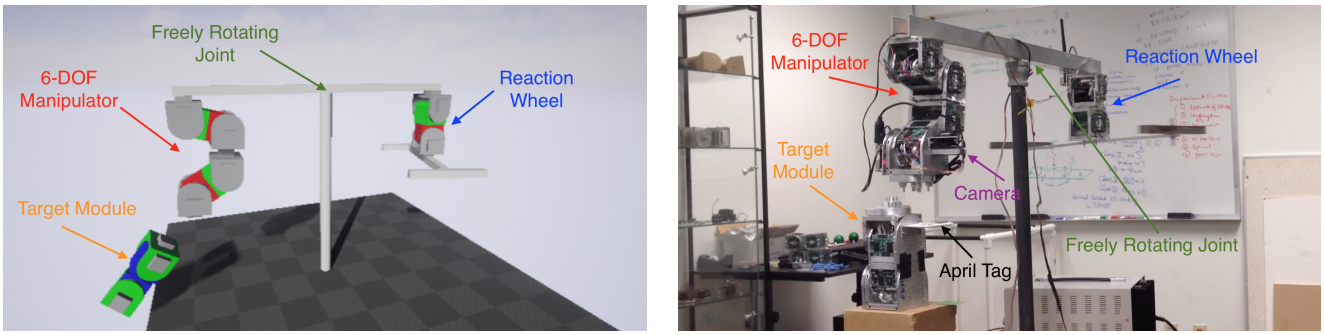

Fig. 3: The simulation and hardware experiments for 6D-docking in simulated micro-gravity environments with a reaction wheel, a 6-DOF manipulator, and a target module. LEFT: A snapshot of the simulation experiments. RIGHT: A snapshot of the hardware experiment with SuperBot self-reconfigurable modules.

TABLE I: Experimental Testing Categories and Results

Category	Distance	# Runs	Success (Sim)	Success (HW)
C-1	Near	10	10	9
C-1	Far	10	10	9
C-2	Left	10	10	10
C-2	Right	10	10	7
C-3	Back	10	10	9
C-3	Front	10	10	10
C-3	Left	10	10	8
C-3	Right	10	10	8
C-4	*	10	10	8

Using this hardware setting, we have performed a total of 20 trial runs for each category with the target module placed on different height. A successful 6D-docking experiment is defined as to be able to dock the target module and lift it up from the table into air. Table 1 lists the results of these experiments. As we can see, an average successful rate across all categories are 86.7%. These results demonstrate the feasibility and functionality of autonomous 6D-docking in the physical 3D world. With the two SuperBot modules 6-DOF manipulator, these experiments demonstrated the capability of docking with the target in some difficult reorientations and displacements. For example, some of Category C-2 cases would be normally impossible to reach with a limited degree of freedom arm. A representative video of these physical experiments is shown here http://www.isi.edu/robots/starcell/6d-docking-demo.mp4.

The observed successes of these experiments provide convincing evidence for the validation and versatility of the proposed framework and the SuperBot self-reconfigurable modules. In tandem with our control algorithms, the manipulator arm was able to overcome the local and aggregate error of the modular robots to perform precise docking. Even in cases where the end-effector dock was inexactly positioned/reoriented relative to the target dock, the grooves present in the teeth of the SuperBot dock design facilitated the sliding of the two docks into a secure and tight connection. This extra error margin compensated for lateral errors arising from the rotation wheel control or camera and also for vertical errors caused by the environment and the dynamic shift of weight of the manipulator arm and rig. The flexibility of such a system demonstrated some new advanced and reliable features in multi-task demanding environments and provides further evidence for advantages over fixed-shaped and task-specialized robots. It is important to note that the limited failure cases observed are attributable to imprecisions in the path planning solutions that sometimes lead the end-effector dock through motions that bump the target dock. These imprecisions arise from the camera and April Tag reading, but more importantly arise from the inherent limitations of using a 6-DOF arm to plan motions in a very complex workspace. However, these impediments are minor and can be obviated by increasing the dexterity of the arm with additional modules and also through inclusion of obstacle-avoidance techniques in the motion planning.

VI. Conclusions and Future Work

This paper presented a novel 6D docking framework for non-fixed base manipulators of self-reconfigurable robots and reported the first successful demonstration of fully autonomous 6D docking using self-reconfigurable robot modules. The evaluations are conducted in physics-based simulation and real-world hardware, demonstrating their validity and real-world applicability to self-reconfigurable robot systems. As such, it marks an important step toward autonomous manipulation and 6D docking of self-reconfigurable modules in space and other challenging environments.

Recently, we have extended our work to include full 6D manipulator aggregation. Additional experiments have been performed to activate the target module once docking has been achieved. The aggregation of the target module with the manipulator through docking allows the system to form a larger manipulator capable of greater dexterity and function. Thus, the new larger manipulator can be used for tasks that the original manipulator could not perform without the additional degrees of freedom provided by the new module. Our work is therefore also the first demonstration of full 6D module aggregation on self-reconfigurable robot hardware and greatly increases the power of self-reconfigurable robotic manipulators to solve problems in difficult or highly dynamic environments. The link: http://www.isi.edu/robots/starcell/Dock-Activate.mp4 shows

an example module aggregation video while all videos can be accessed through http://www.isi.edu/robots/starcell/.

There is however, much future work to be done in this research area. For example, a tighter integration of the reaction wheel control with the manipulation algorithm may be necessary when the arm's movement inevitably changes the center of mass of the entire system. For now, they are primarily independent, with the reaction-wheel control algorithm initiating the manipulation algorithm autonomously once the arm has converged on the target. We have performed initial experiments in which one or more additional "looks" are performed during the execution of Algorithm 2. If the camera detects that the target is no longer sufficiently lined up, the reaction wheel takes over to converge upon it again before returning control to the manipulation algorithm. This would allow for accommodation of scenarios where the target module is also moving or on a frictionless rotatable platform. Initial results of such hybrid approaches are promising, but a more principled approach must be determined to combine these two components.

REFERENCES

[1] B. Salemi, M. Moll, and W.-M. Shen, "Superbot: A deployable, multi-functional, and modular self-reconfigurable robotic system," in *Proceedings of the IEEE International Conference on Intelligent Robotic Systems (IROS)*, 2006.

[2] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-tran: Self-reconfigurable modular robotic system," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 4, pp. 431–441, 2002.

[3] E. H. Østergaard, K. Kassow, R. Beck, and H. H. Lund, "Design of the atron lattice-based self-reconfigurable robot," *Autonomous Robots*, vol. 21, no. 2, pp. 165–183, 2006.

[4] M. Rubenstein, K. Payne, P. Will, and W.-M. Shen, "Docking among independent and autonomous conro self-reconfigurable robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, pp. 2877–2882.

[5] A. Sproewitz, A. Billard, P. Dillenbourg, and A. J. Ijspeert, "Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 4259–4264.

[6] N. Eckenstein and M. Yim, "Design, principles, and testing of a latching modular robot connector," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*. IEEE, 2014, pp. 2846–2851. [Online]. Available: http://modlabupenn.org/wp-content/uploads/latching˙xface˙overlay.pdf

[7] J. W. Romanishin, K. Gilpin, and D. Rus, "M-blocks: Momentum-driven, magnetic modular robots," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 4288–4295.

[8] J. Davey, N. Kwok, and M. Yim, "Emulating self-reconfigurable robots-design of the smores system," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4464–4469.

[9] N. Eckenstein and M. Yim, "Area of acceptance for 3-D self-aligning robotic connectors: Concepts, metrics, and designs," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 1227–1233.

[10] W.-M. Shen, R. Kovac, and M. Rubenstein, "Singo: A single-end-operative and genderless connector for self-reconfiguration, self-assembly and self-healing," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 4253–4258.

[11] M. Yim, B. Shirmohammadi, J. Sastra, M. Park, M. Dugan, and C. J. Taylor, "Towards robotic self-reassembly after explosion," *Departmental Papers (MEAM)*, p. 147, 2007.

[12] S. Murata, K. Kakomura, and H. Kurokawa, "Docking experiments of a modular robot by visual feedback," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 625–630.

[13] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, and S. Murata, "Distributed self-reconfiguration of m-tran iii modular robotic system," *The International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 373–386, 2008.

[14] Y. Zhu, H. Jin, X. Zhang, J. Yin, P. Liu, and J. Zhao, "A multi-sensory autonomous docking approach for a self-reconfigurable robot without mechanical guidance," *International Journal of Advanced Robotic Systems*, vol. 11, 2014.

[15] H. Wei, Y. Cai, H. Li, D. Li, and T. Wang, "Sambot: A self-assembly modular robot for swarm robot," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 66–71.

[16] W. Wang, Z. Li, W. Yu, and J. Zhang, "An autonomous docking method based on ultrasonic sensors for self-reconfigurable mobile robot," in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*. IEEE, 2009, pp. 1744–1749.

[17] P. Won, M. Biglarbegian, and W. Melek, "Development of an effective docking system for modular mobile self-reconfigurable robots using extended kalman filter and particle filter," *Robotics*, vol. 4, no. 1, pp. 25–49, 2015.

[18] M. Kutzer, C. Brown, D. Scheidt, M. Armand, K. Wolfe, M. Moses, and G. Chirikjian, "Reconfigurable robotic system with independently mobile modules," *Johns Hopkins APL technical digest*, vol. 28, no. 3, pp. 270–271, 2010.

[19] T. T. J. S. J. D. J. G. V. K. J. Paulos, N. Eckenstein and M. Yim, "Automated self-assembly of large maritime structures by a team of robotic boats," *IEEE Transactions on Automation Science and Engineering*, vol. 12, p. 958968, July 2015.

[20] S.-k. Yun and D. Rus, "Self assembly of modular manipulators with active and passive modules," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 1477–1482.

[21] T. Martinez-Marin and T. Duckett, "Robot docking by reinforcement learning in a visual servoing framework," in *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*, vol. 1. IEEE, 2004, pp. 159–164.

[22] W.-H. Zhu, T. Lamarche, E. Dupuis, and E. Martin, "Modular robot manipulators with precision control."

[23] W. K. Chung, J. Han, Y. Youm, and S. Kim, "Task based design of modular robot manipulator using efficient genetic algorithm," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 507–512.

[24] H. Liu, B. Liang, W. Xu, Z. Di, and X. Wang, "A ground experiment system of a free-floating robot for fine manipulation," *Int J Adv Robotic Sy*, vol. 9, no. 183, 2012.

[25] S. A. A. Moosavian and E. Papadopoulos, "Free-flying robots in space: an overview of dynamics modeling, planning and control," *Robotica*, vol. 25, no. 05, pp. 537–547, 2007.

[26] S. Yahya, M. Moghavvemi, and H. A. F. Mohamed, "Redundant manipulators kinematics inversion," vol. 6, pp. 5462–5470+, 2011.

[27] B. Durmus, H. Temurtas, and A. Gun, "An inverse kinematics solution using particle swarm optimization," in *Proc. of sixth International Advanced Technologies Symposium (IATS?11), Turkey*, 2011, pp. 193–197.

[28] N. Rokbani and A. M. Alimi, "Ik-pso, pso inverse kinematics solver with application to biped gait generation," *arXiv preprint arXiv:1212.1798*, 2012.

[29] N. Rokbani and A. Alimi, "Inverse kinematics using particle swarm optimization, a statistical analysis," *Procedia Engineering*, vol. 64, no. 0, pp. 1602 – 1611, 2013.

[30] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[31] J. Sun, C. Lai, and X. Wu, *Particle Swarm Optimisation: Classical and Quantum Perspectives*, ser. Chapman & Hall/CRC Numerical Analysis and Scientific Computing Series. Taylor & Francis, 2011.

[32] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *J. Artif. Evol. App.*, vol. 2008, pp. 4:1–4:10, Jan. 2008.

[33] T. Collins and W.-M. Shen, "Paso: An integrated, scalable pso-based optimization framework for hyper-redundant manipulator path planning and inverse kinematics," Information Sciences Institute, University of Southern California, Tech. Rep. ISI-TR-697, 2015.

[34] T. Collins, N. O. Ranasinghe, and W.-M. Shen, "Remod3-D: A high-performance simulator for autonomous, self-reconfigurable robots," Tokyo, Japan, Nov. 2013.