**2010 IEEE International Conference on Robotics and Automation**
**Anchorage Convention District**
**May 3-8, 2010, Anchorage, Alaska, USA**

# Trajectory Generation and Steering Optimization for Self-Assembly of a Modular Robotic System

Kevin C. Wolfe, Michael D.M. Kutzer, Mehran Armand, and Gregory S. Chirikjian

***Abstract*— A problem associated with motion planning for the assembly of individual modules in a new self-reconfigurable modular robotic system is presented. Modules of the system are independently mobile and can be driven on flat surfaces in a similar fashion to the classic kinematic cart. This problem differs from most nonholonomic steering problems because of an added constraint on one of the internal states. The constraint properly aligns the docking mechanism, allowing modules to connect with one another along wheel surfaces. This paper presents an initial method for generating trajectories and control inputs that allow module assembly. It also provides an iterative method for locally optimizing a nominal control function using weighted perturbation functions, while preserving the final pose and internal states.**

***Index Terms*— modular robots, self-assembly, nonholonomic motion planning, trajectory optimization**

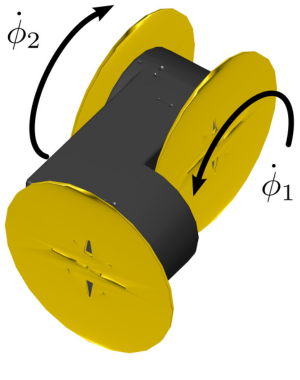

Fig. 1. Single module of the modular robotic system.

## I. Introduction

Self-reconfigurable modular robotic systems have become increasingly popular in recent years [1]. Still, relatively few systems exist that are capable of self-propelled autonomous self-assembly [2], [3]. In this paper we briefly present independently mobile modules of a new modular self-reconfigurable robotic system which has been designed to provide damage repair or mitigation in dangerous environments by rebuilding broken communication, electrical, steam, and/or hydraulic connections. A more detailed description of the modular system, its capabilities, and design can be found in [4].

The above deployment scenario requires that the modules be able to self-assemble. This serves as motivation for the emphasis of this paper which discusses an initial path planning method developed for allowing these modules to dock with one another. In addition, a method for the local optimization of a set of control inputs with respect to driving effort is presented. This problem is nontrivial due to the nonholonomic constraints inherent in the driving kinematics of a single module. The initial control inputs are taken as piecewise continuous functions and then optimized using weighted sinusoidal perturbations. The optimization method presented also uses Jacobian inverse iterations, which have previously been studied in the context of redundant manipulators [5], [6], [7].

Nonholonomic motion planning has been studied from a variety of perspectives. While a few general methods have been proposed for certain classes of systems [8], [9], [10], [11], [12], [13], much of the work has focused on specific systems. In particular, the classic kinematic cart (also referred to as a differential-drive vehicle), which is characterized as possessing two independently-controlled drive wheels that share a common axis of rotation, has been the subject of significant research effort [14], [15]. However, steering and motion planning for this vehicle and others typically only refers to controlling the pose of the vehicle, i.e., its position and orientation [16], [17], [18]. For this application, we are concerned with controlling not just the pose, but also one of the internal states of the vehicle. This additional constraint, which arises from the geometry for connecting modules, adds complexity to an already nontrivial problem.

Modules of this new system, Fig. 1, have three independently controlled wheels, allowing disconnected modules to drive to other single or connected modules for module-to-module docking and assembly of various structures. The assembly of a typical chain-like structure is illustrated in Fig. 2. By controlling the rotational speed of the two wheels that share a common axis of rotation, an individual module can be driven in a similar fashion to the classic kinematic cart. This is possible because of the geometry and weight distribution of the modules; on a horizontal plane, the friction

This research was partially supported under an appointment to the Department of Homeland Security (DHS) Scholarship and Fellowship Program, administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and DHS. ORISE is managed by Oak Ridge Associated Universities (ORAU) under DOE contract number DE-AC05-06OR23100.

This work was also supported in part by NSF grant IIS-0915542 RI: Small: Robotic Inspection, Diagnosis, and Repair and through Internal Research and Development funds provided by the Johns Hopkins University Applied Physics Laboratory (JHU/APL).

K. Wolfe, M. Kutzer, M. Armand and G. Chirikjian are with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, Maryland. {kevin.wolfe, mkutzer1, marmand2, gregc}@jhu.edu

M. Kutzer and M. Armand are also with the Milton S. Eisenhower Research & Technology Development Center, Johns Hopkins University Applied Physics Laboratory, Laurel, Maryland. {michael.kutzer, mehran.armand}@jhuapl.edu

**978-1-4244-5040-4/10/$26.00 ©2010 IEEE**

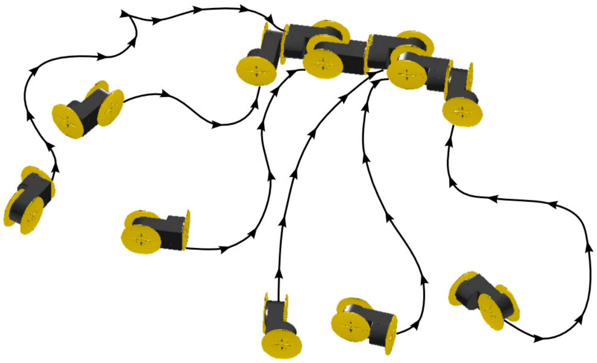

Fig. 2. The motivation for this work is based on developing a method to assemble a group of individual modules. Here, a group of individual modules (lower) drive and dock with one another forming a more complex mechanism (upper).

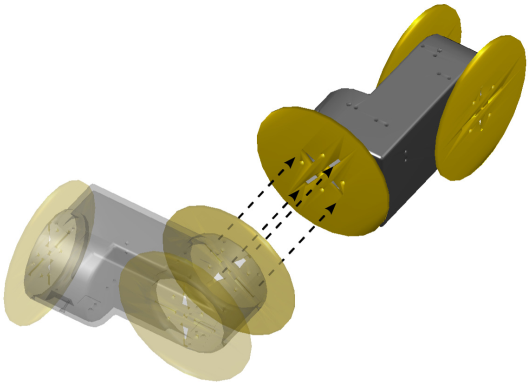

Fig. 3. For two modules to dock, their adjacent wheels must be properly aligned.

experienced by the third wheel is dominated by that of the two drive wheels. Thus, the third wheel is not used for driving control, rather we specify its motion using an equation which minimizes the slip it experiences during driving [4]. Modules of the system dock with one another along the faces of the three wheels. The docking mechanisms are homogeneous across all wheels and modules, allowing any wheel to dock with any other. For two wheels to properly dock, they must be offest by 90° or 270° so that the hooks of one wheel align with the slots on the adjacent wheel as shown in Fig. 3. Thus, for two modules to dock with one another, both their poses and wheel angles must be compatible.

## II. The Module Assembly Problem

### A. Problem Formulation

Similar to the classic kinematic cart, modules of our system are driven and steered by specifying the rotational rates of the two wheels sharing a common axis. We can specify the pose of the cart in the plane using an element of $G \equiv \mathbb{R}^2 \times \mathbb{S}^1$. Throughout this paper the pose of a vehicle will be denoted by $g \in G$, where:

$$g = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}. \tag{1}$$

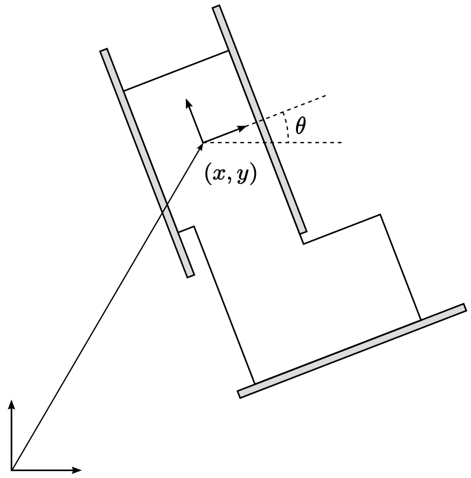

Fig. 4. Pose of an individual module with respect to a fixed world frame.

Here, as illustrated in Fig. 4, $x$ and $y$ represent the distance from a fixed frame in the plane to the midpoint of the line segment connecting the center of each drive wheel. Module orientation, $\theta$, is defined by the angle between the shared axis of rotation and the fixed $x$-axis. We also assign a frame to each wheel with the $z$-axis of each wheel aligned with the axis of rotation and pointing outward from the module. Using this, we are then able to take $\phi_i$ as the angle between the $i$th wheel's local $x$-axis and the horizontal. Assuming a no-slip condition between each of the drive wheels and the ground imposes a nonholonomic constraint. Using this, the system can be modeled using $\dot{\phi}_i$'s (illustrated in Fig. 1) as control inputs. Taking $r$ as the wheel radius and $W$ as the distance between the wheels, we can write:

$$\dot{g}(t) = B(g(t))\ \dot{\phi}(t) \tag{2}$$

where

$$B(g(t)) = \frac{r}{2}\begin{pmatrix} \sin(\theta(t)) & -\sin(\theta(t)) \\ -\cos(\theta(t)) & \cos(\theta(t)) \\ -\frac{2}{W} & -\frac{2}{W} \end{pmatrix} \tag{3}$$

and

$$\dot{\phi}(t) = \begin{pmatrix} \dot{\phi}_1(t) \\ \dot{\phi}_2(t) \end{pmatrix}.$$

Our goal is to develop a control function $\dot{\phi}(t)$ for $t \in [0, T]$ to drive the vehicle from any initial pose to any other goal pose. In doing so, we seek a solution that also has a constraint on one of the wheels, allowing the wheel to be docked with a second stationary module or chain of modules at time $T$. Thus, we add an additional constraint on $\phi_d$ (where $d$ is 1 or 2) based on a desired docking configuration. Inasmuch, we require

$$\phi_d(T) \in \{\phi_f, \phi_f + \pi\} \tag{4}$$

where $\phi_f$ is a potential configuration for the docking of wheel $d$. At present, we will look at this problem for a single module in the unobstructed plane. Finally, we require that the docking method specify the driving module to approach the stationary module on a straight path for the last $2r$ of the trajectory. This helps to ensure a feasible trajectory so that the two wheels do not collide.

With respect to optimization, our approach seeks to minimize the cost functional $J$ where

$$J(\dot{\phi}(t),T)=\frac{1}{2}\int_0^T\left(\dot{\phi}_1^2(t)+\dot{\phi}_2^2(t)\right)dt. \tag{5}$$

The integrand of this cost functional closely relates to the total kinetic energy of the vehicle at time $t$. Thus, minimizing this cost can be thought of as analogous to minimizing the effort required to go from an initial pose $g(0)=(x_0,y_0,\theta_0)^{\mathsf{T}}$ to an end pose $g(T)=(x_f,y_f,\theta_f)^{\mathsf{T}}$.

### B. Controllability

Consider a new configuration space $\hat{G}\equiv\mathbb{R}^2\times\mathbb{S}^1\times\mathbb{S}^1$. We can then specify the pose of the vehicle along with the value of one of the internal states, $\phi_i$ using $\hat{g}\in\hat{G}$ where

$$\hat{g}=\begin{pmatrix}x\\y\\\theta\\\phi_i\end{pmatrix}.$$

The new system can then be represented by

$$\dot{\hat{g}}=f_1\dot{\phi}_1+f_2\dot{\phi}_2 \tag{6}$$

where

$$f_1=\frac{r}{2}\begin{pmatrix}\sin(\theta(t))\\-\cos(\theta(t))\\-\frac{2}{W}\\\delta_{i1}\end{pmatrix}\quad\text{and}\quad f_2=\frac{r}{2}\begin{pmatrix}-\sin(\theta(t))\\\cos(\theta(t))\\-\frac{2}{W}\\\delta_{i2}\end{pmatrix}. \tag{7}$$

Here, $\delta_{ij}$ represents the Kronecker delta function.

It is easy to show that the *controllability Lie algebra* as defined in [8] is rank four. Thus, the system is small-time locally controllable. This also means that there exists at least one control function to drive the system defined in (6) from any initial state to any desired end state. It is also interesting and important to note that a system defined for $(x,y,\theta,\phi_1,\phi_2)^{\mathsf{T}}$ would not be controllable because of a holonomic constraint that exists between $\theta$ , $\phi_1$, and $\phi_2$.

### C. Initial Trajectory Generation

Obtaining a solution to the problem posed in Section II-A is nontrivial due to the added internal constraint on one of the two drive wheels. We start by considering sub-trajectories consisting of straight lines and pivots about wheel $d$. Straight trajectories can be obtained using a control input of

$$\dot{\phi}(t)=\omega_{sj}\begin{pmatrix}-1\\1\end{pmatrix}\quad\text{for}\quad\omega_{sj}\in\mathbb{R}. \tag{8}$$

While pivots about wheel $d$ can be obtained using

$$\dot{\phi}(t)=\omega_{pj}\begin{pmatrix}-\delta_{d1}\\\delta_{d2}\end{pmatrix}\quad\text{for}\quad\omega_{pj}\in\mathbb{R}. \tag{9}$$

A trajectory that alternates between pivot and straight sub-trajectories can be obtained using a piecewise constant input of the form

$$\dot{\phi}(t)=\begin{cases}\omega_{p1}(-\delta_{d1}\ \delta_{d2})^{\mathsf{T}}, & \text{for } t\in[0,t_{p1}]\\ \omega_{s1}(-1\ \ 1)^{\mathsf{T}}, & \text{for } t\in(t_{p1},t_{s1}]\\ \omega_{p2}(-\delta_{d1}\ \delta_{d2})^{\mathsf{T}}, & \text{for } t\in(t_{s1},t_{p2}]\\ \vdots & \\ \omega_{sn}(-1\ \ 1)^{\mathsf{T}}, & \text{for } t\in(t_{pn},t_{sn}]\end{cases}. \tag{10}$$

Note that without any constraints, we can construct a trajectory to drive a module from any $g(0)$ to any $g(T)$ using only two pivot inputs and one straight input. However, with the constraint on wheel $d$ given in (4) and the need for the last $2r$ of the trajectory to be straight, we must add a second straight input at the end of the trajectory, i.e., $n=2$ in (10). This driving strategy is demonstrated in Fig. 5.

For an input of the form given in (10), we can rewrite the docking wheel constraint from (4) as

$$\phi_0-\phi_f+\sum_{j=1}^{n}\omega_{sj}(t_{sj}-t_{pj})=k\pi\quad\text{for some}\quad k\in\mathbb{Z}. \tag{11}$$

Also, we can express the requirement for the last $2r$ of the trajectory being straight as

$$|\omega_{sn}|(t_{sn}-t_{pn})\geq 2. \tag{12}$$

We now look at constructing a trajectory for an input of the form in (10) for $n=2$. Because wheel $d$ does not translate during pivot sub-trajectories, we can use its initial position, $p_0=(x_{d0},y_{d0})^{\mathsf{T}}$, and final desired position, $p_f=(x_{df},y_{df})^{\mathsf{T}}$, to reformulate the problem. To do this, first consider the parametric equation of a line, $\ell_f$,

$$\begin{pmatrix}x\\y\end{pmatrix}=s\begin{pmatrix}-\sin\theta_f\\\cos\theta_f\end{pmatrix}+\begin{pmatrix}x_{df}\\y_{df}\end{pmatrix}. \tag{13}$$

We know that without the constraints we can construct a trajectory to drive a module from $g(0)$ to any $g(t_1)$ using two pivots and one straight input. Inasmuch, we can do so for $g(t_1)$ such that

$$(x_d(t_1),y_d(t_1))^{\mathsf{T}}\in\ell_f\quad\text{and}\quad\theta(t_1)=\theta_f. \tag{14}$$

Thus the problem can be further reduced to a search for a point $p_1=(x_d(t_1),y_d(t_1))^{\mathsf{T}}$, illustrated in Fig. 5, over the points of $\ell_f$ for $|s|\geq 2r$ such that

$$\phi_0-\phi_f+\text{dir}_{01}\frac{|p_1-p_0|}{r}+\text{dir}_{1f}\frac{|p_f-p_1|}{r}=k\pi\quad\text{for}\quad k\in\mathbb{Z} \tag{15}$$

where $\text{dir}_{01}=\text{sgn}(\dot{\phi}_d(t_{s1}))$, $\text{dir}_{1f}=\text{sgn}(\dot{\phi}_d(t_{s2}))$, and $|\cdot|$ represents the Euclidean norm. Note that $\text{dir}_{1f}$ is not a free parameter but a function of $s$ and $d$.

Given an appropriate $p_1$ and $\text{dir}_{01}$ we determine the minimum angles that the module must turn through to correctly orient the module for the two straight sub-trajectories. These are given by

$$|\Delta\theta_{01}|=\arccos\left\langle\begin{pmatrix}-\sin\theta_0\\\cos\theta_0\end{pmatrix},\text{dir}_{01}\left(\frac{p_1-p_0}{|p_1-p_0|}\right)\right\rangle \tag{16}$$

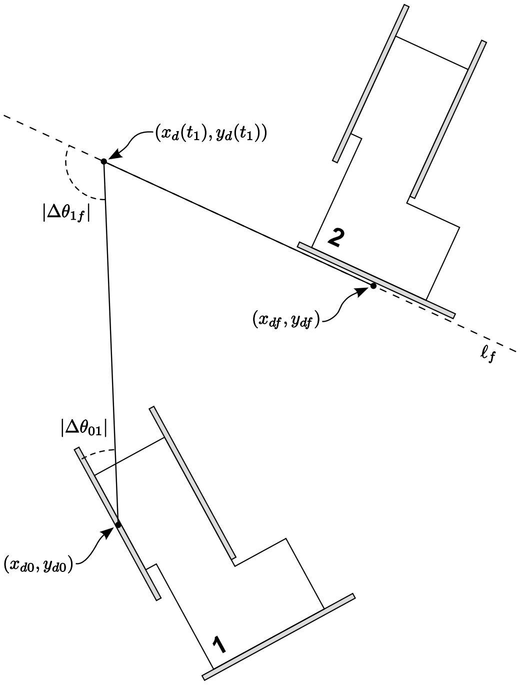

Fig. 5. The initial driving algorithm uses the line $\ell_f$ to search for valid trajectories for wheel $d$. Here we illustrate $p_0$, $p_1$, and $p_f$ for Module 1 docking with Module 2.

and

$$|\Delta\theta_{1f}| = \arccos\left\langle \mathrm{dir}_{01}\left(\frac{p_1 - p_0}{|p_1 - p_0|}\right), \begin{pmatrix} -\sin\theta_f \\ \cos\theta_f \end{pmatrix} \right\rangle. \tag{17}$$

The sign of $\Delta\theta_{01}$ and $\Delta\theta_{1f}$ is obtained using the sign of the $z$-component of

$$\begin{pmatrix} -\sin\theta_0 \\ \cos\theta_0 \\ 0 \end{pmatrix} \times \mathrm{dir}_{01} \begin{pmatrix} x(t_1) - x_0 \\ y(t_1) - y_0 \\ 0 \end{pmatrix} \tag{18}$$

and

$$\mathrm{dir}_{01} \begin{pmatrix} x(t_1) - x_0 \\ y(t_1) - y_0 \\ 0 \end{pmatrix} \times \begin{pmatrix} -\sin\theta_f \\ \cos\theta_f \\ 0 \end{pmatrix} \tag{19}$$

respectively.

For a given $T$ and the piecewise constant driving structure specified in (10), the minimum cost trajectory is obtained using a constant $|\dot{\phi}(t)|$. Thus, we desire that the relation $\omega_{pj} = \sqrt{2}\omega_{sj}$ hold for all $j$. Inasmuch, we can define a constant, $\omega_s$, for the magnitude of both straight trajectories. Using this we can see that

$$\begin{aligned} T &= \Delta t_{p1} + \Delta t_{s1} + \Delta t_{p2} + \Delta t_{s2} \\ &= \frac{\Delta\theta_{01} W}{|\omega_{p1}| r} + \frac{|p_1 - p_0|}{|\omega_{s1}| r} + \frac{\Delta\theta_{1f} W}{|\omega_{p2}| r} + \frac{|p_f - p_1|}{|\omega_{s2}| r} \\ &= \frac{|p_1 - p_0| + |p_f - p_1|}{\omega_s r} + \frac{(\Delta\theta_{01} + \Delta\theta_{1f}) W}{\sqrt{2}\omega_s r}. \end{aligned} \tag{20}$$

This gives

$$\omega_s = \frac{\sqrt{2} r}{T(\sqrt{2}(|p_1 - p_0| + |p_f - p_1|) + (\Delta\theta_{01} + \Delta\theta_{1f}) W)}. \tag{21}$$

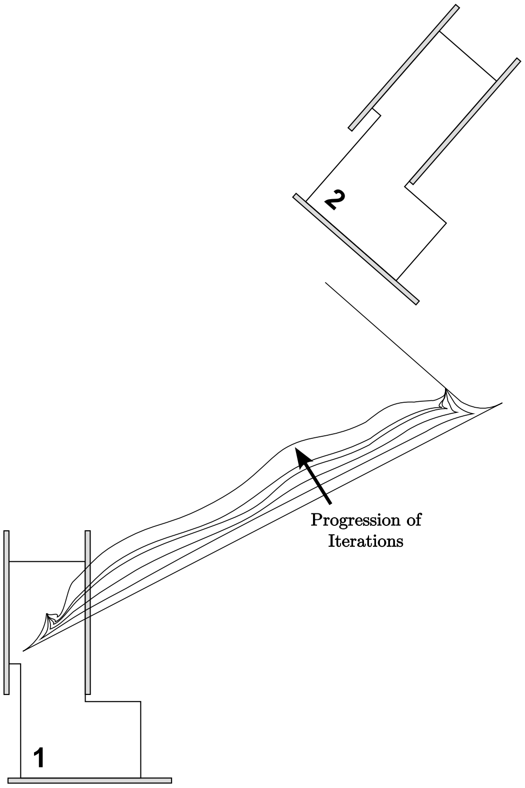

Fig. 6. Using the sinusoidal perturbation basis functions, an initial nominal path for docking Module 1 with Module 2 is locally optimized.

Finally, we define a set

$$S = \{(s, \mathrm{dir}_{01}) : |s| \geq 2r,\ \mathrm{dir}_{01} \in \{-1, 1\},\ \text{Eqs. (13), (15)}\}$$

and minimize $J$ over $S$. Then given a particular $(s, \mathrm{dir}_{01})$ pair, we can determine an input control function using (10), (13), (16), (17), (18), (19), (20), and (21).

### *D. Iterative Local Optimization*

Once any nominal control input is given that satisfies the desired end pose and wheel angle constraint, we can use weighted perturbations to minimize the cost of the input in an iterative fashion. Fig. 6 demonstrates how iteratively perturbing the control inputs can take a straight nominal trajectory and incrementally alter it to reduce the cost without violating the problem constraints.

Given a particular control input $\dot{\phi}(t)$ for $t \in [0, T]$ a trajectory can be obtained using (2) and (3). We can specify a new perturbed set of wheel angle functions

$$\Phi(t) = \phi(t) + \sum_{j=1}^{N} (\epsilon_j \vec{e}_1 + \epsilon_{j+N} \vec{e}_2) \psi_j(t) \tag{22}$$

where $\vec{e}_i$'s are the standard basis vectors, $\psi_j(t)$'s are perturbation basis functions, and $\epsilon_j$'s are the weighting factors used for optimization. This can be represented as follows

$$\Phi(t) = \phi(t) + \Psi(t)\vec{\epsilon} \quad \text{for} \quad \vec{\epsilon} \in \mathbb{R}^{2N} \tag{23}$$

$$\text{where} \quad \Psi(t) = \begin{pmatrix} \psi_1(t) \cdots \psi_N(t) & \vec{0}^{\mathsf{T}} \\ \vec{0}^{\mathsf{T}} & \psi_1(t) \cdots \psi_N(t) \end{pmatrix}.$$

We can now look at the perturbed pose of the module at time $T$ which is given by

$$g(T,\vec{\epsilon}) = g(0,\vec{0}) + \int_0^T B(g(\tau))\dot{\Phi}(\tau,\vec{\epsilon})\ d\tau. \tag{24}$$

The partial derivative of this pose with respect to $\epsilon_j$ evaluated at $\vec{\epsilon}=\vec{0}$ can be numerically approximated by

$$\left.\frac{\partial g}{\partial \epsilon_j}\right|_{\vec{\epsilon}=\vec{0}} \approx \frac{1}{\varepsilon}\Big(g(T,\varepsilon\vec{e}_j) - g(T,\vec{0})\Big) \tag{25}$$

for $\varepsilon << 1$. Using these partial derivatives for the end pose of the module we can establish a constraint on the perturbation weights, $\vec{\epsilon}$, of

$$\begin{aligned} \left(\frac{\partial g}{\partial \epsilon_1}\cdots\frac{\partial g}{\partial \epsilon_{2N}}\right)\vec{\epsilon} &= \begin{pmatrix}0\\0\\0\end{pmatrix} \\ \frac{\partial g}{\partial \vec{\epsilon}^{\mathsf{T}}}\vec{\epsilon} &= \vec{0}. \end{aligned} \tag{26}$$

This ensures that the end pose is unchanged by the perturbations.

Now, to minimize the cost given in (5) we can use Lagrange multipliers and the Hamiltonian,

$$H = J + \lambda^{\mathsf{T}} M\vec{\epsilon} \tag{27}$$

where

$$M = \frac{\partial g}{\partial \vec{\epsilon}^{\mathsf{T}}}.$$

It is easily verifiable that the cost functional can be written in the form

$$J(\dot{\Psi}(t),T) = J(\dot{\phi}(t),T) + \vec{c}^{\mathsf{T}}\vec{\epsilon} + \vec{\epsilon}^{\mathsf{T}}A\vec{\epsilon} \tag{28}$$

for

$$\vec{c}^{\mathsf{T}} = \int_0^T \dot{\phi}(\tau)^{\mathsf{T}}\dot{\Psi}(\tau)\ d\tau \quad\text{and}\quad A = \frac{1}{2}\int_0^T \dot{\Psi}(\tau)^{\mathsf{T}}\dot{\Psi}(\tau)\ d\tau.$$

As such, local minimization can be performed if we can determine $\vec{\epsilon}$ and $\vec{\lambda}$ for which

$$\frac{\partial H}{\partial \vec{\epsilon}} = 0 \quad\text{and}\quad \frac{\partial H}{\partial \vec{\lambda}} = 0. \tag{29}$$

If we simultaneously consider the constraint given in (26), both are attained by finding $\vec{\epsilon}$ and $\vec{\lambda}$ such that

$$\begin{pmatrix} A & M^{\mathsf{T}} \\ M & 0 \end{pmatrix}\begin{pmatrix}\vec{\epsilon}\\ \vec{\lambda}\end{pmatrix} = \begin{pmatrix}-\vec{c}\\ \vec{0}\end{pmatrix}. \tag{30}$$

We note that solutions are not assured to exist if the square matrix on the left-hand-side of (30) is not full rank. However, we can ensure that $A$ is full rank through our choice of perturbation basis functions. The choice of basis functions also affects the rank of $M$.

It is also important to consider the assumption made in taking the partial derivatives in (25), which requires $\epsilon_j$'s be small. This can be overcome through an iterative process. To do this, we first determine $\vec{\epsilon}$; then we uniformly scale the perturbation weights to limit the norm of the $\vec{\epsilon}$. This has the effect of taking a small step in the direction of minimization. We then take $\dot{\Psi}(t)$ to be our new nominal control input for the next iteration.

We note that because the end pose constraint in (26) uses a linearized Jacobian, small errors in relative pose between iterations may cause significant drift after many iterations. This can be corrected by forcing the new end pose after the $k$th iteration, $g(T,\vec{\epsilon}^k)$, back to the end pose of the original input, $g_f = (x_f, y_f, \theta_f)^{\mathsf{T}}$, if the difference between the two exceeds a specified threshold. This forcing is performed using the Jacobian resulting from evaluating (25) about $\vec{\epsilon}^k$ resulting from (30) for the $k$th iteration:

$$\left(\left.\frac{\partial g}{\partial \vec{\epsilon}^{\mathsf{T}}}\right|_{\vec{\epsilon}=\vec{\epsilon}^k}\right)\left(\vec{\epsilon}^{\tilde{k}} - \vec{\epsilon}^k\right) = g_f - g(T,\vec{\epsilon}^k). \tag{31}$$

Here $\vec{\epsilon}^{\tilde{k}}$ is the updated $\vec{\epsilon}$ for the $k$th iteration.

For our particular problem, suppose we have a control function that satisfies the constraints discussed in Section II-A. We must ensure that the last $2r$ of the trajectory is unchanged. To do this, we consider the optimization to take place between $t \in [0,T')$ where $T'$ represents the largest $t$ for which $|(x_f,y_f)^{\mathsf{T}} - (x(t),y(t))^{\mathsf{T}}| \geq 2r$. Also, to maintain $\phi_d(T') = \Phi_d(T')$, we choose the following periodic perturbation basis functions

$$\psi_j(t) = \frac{\sin\left(j\frac{t}{T'}\pi\right)}{\frac{j\pi}{T'}}. \tag{32}$$

Since $\psi_j(T') = 0$ in (32) for all $j$, using this set of perturbation basis functions ensures that the wheel angle at time $T$ is also unchanged. Thus, if the constraints for docking are satisfied using the control function $\phi(t)$, they will also be satisfied by $\Phi(t)$. It is also important to note that because we choose frequencies that are integral multiples of $\frac{t}{T'}\pi$, the perturbation functions are linearly independent.

## III. Discussion and Future Work

We have presented a method for generating and optimizing control inputs used for the self-assembly of individual modules of a new self-reconfigurable robotic system. The initial control input generation presented in Section II-C provides a method for steering a module from any initial pose to any final pose while ensuring that the docking wheel constraint is properly observed. This method largely relies on the distance traveled by the desired docking wheel. We can combine this with the local optimization technique described in Section II-D to generate a trajectory and steering function that is locally optimized with respect to total driving effort. We note that this method does not guarantee globally optimal results. Rather, optimality and the final steering cost is limited by the choice and number of perturbation basis functions utilized. Fig. 7 illustrates this along with simulation results of this technique for our module.

The amount of improvement attained through local optimization is also dependent on the desired change in pose and the starting and target wheel angle. This is illustrated in Fig. 8. This figure also demonstrates how the cost function is reduced with each iteration until a local minimum is reached with respect to the perturbation basis functions.

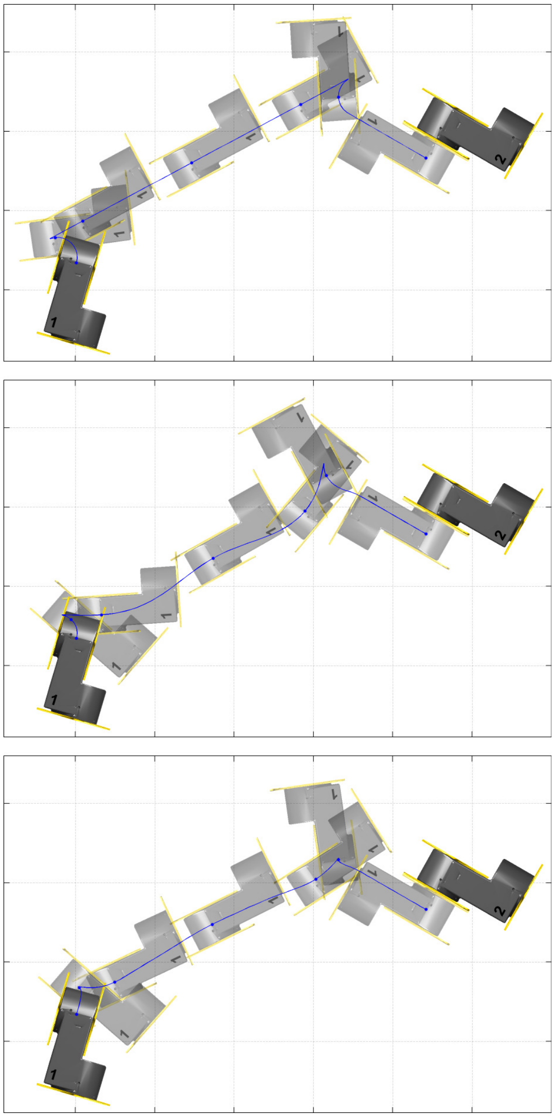

Fig. 7. (Top) Initial trajectory, $J = 21.35$, for Module 1 docking with Module 2. (Middle) Perturbed trajectory, $J = 18.91$, using $N = 5$. (Bottom) Perturbed trajectory, $J = 16.80$, using $N = 10$.

The methods developed here do not yet incorporate obstacle avoidance which will be addressed in future work. We must also consider the problem of assembling multiple modules, including order of assembly. In addition, future investigation may include methods for performing small corrections for wheel slippage and other modeling error. Finally, we will experimentally test the methods developed here and in our future work to determine the reliability and repeatability of the assembly process.

## Acknowledgments

The authors would like to thank Mr. Matthew Moses for his helpful insights and comments.

## References

[1] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems: Challenges and opportunities for the future," *IEEE Robot. Autom. Mag.*, pp. 43–52, Mar. 2007.

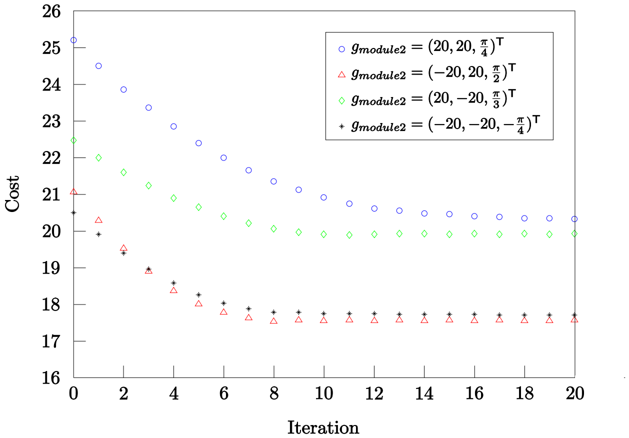

Fig. 8. The cost, $J$, with respect to the iteration of the optimization process for several end poses. End poses are given as the pose of a second stationary module. The four simulations assume a wheel 2 to wheel 2 docking arrangement, $\phi_2(0) = 0$ for both modules, $g_0 = (0, 0, 0)^{\mathsf{T}}$, N=15, and $T = 10$.

[2] R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1115–1130, Dec. 2006.

[3] R. Groß and M. Dorigo, "Self-assembly at the macroscopic scale," *Proc. of the IEEE*, vol. 96, no. 9, pp. 1490–1508, Sep. 2008.

[4] M. D. Kutzer, M. Moses, C. Y. Brown, D. H. Scheidt, G. S. Chirikjian, and M. Armand, "Design of a new independently-mobile reconfigurable modular robot," in *IEEE International Conf. on Robotics and Automation*, 2010. In press.

[5] C. A. Klein and C.-H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 13, no. 3, pp. 245–250, Mar./Apr. 1983.

[6] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proc. IEEE Conf. Robotics and Automation*, St. Louis, MO, Mar. 1985, pp. 722–725.

[7] Y. S. Chung, M. Griffis, and J. Duffy, "Repeatable joint displacement generation for redundant robotic systems," *ASME Tran. Mechanical Design*, vol. 116, no. 1, pp. 11–16, Mar. 1994.

[8] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[9] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: Steering using sinusoids," *IEEE Trans. Autom. Control*, vol. 38, no. 5, pp. 700–716, May 1993.

[10] J.-P. Laumond, Ed., *Robot Motion Planning and Control*. London: Springer, 1998.

[11] J. Ostrowski, "Steering for a class of dynamic nonholonomic systems," *IEEE Trans. Autom. Control*, vol. 45, no. 8, pp. 1492–1498, Aug. 2000.

[12] A. M. Bloch and N. H. McClamroch, "Control of mechanical systems with classical nonholonomic constraints," in *Proc. IEEE Conf. on Decision and Control*, 1989, pp. 201–205.

[13] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch, "Control and stabilization of nonholonomic dynamic systems," *IEEE Trans. Autom. Control*, vol. 37, no. 11, pp. 1746–1757, Nov. 1992.

[14] D. J. Balkcom and M. T. Mason, "Time optimal trajectories for bounded velocity differential drive vehicles," *Int. J. Robotics Research*, vol. 21, pp. 199–217, 2002.

[15] H. Chitsaz, S. M. LaValle, D. J. Balkcom, and M. T. Mason, "Minimum wheel-rotation paths for differential-drive mobile robots," *Int. J. Robotics Research*, vol. 28, pp. 66–80, Jan. 2009.

[16] J.-P. Laumond, "Feasible trajectories for mobile robots with kinematic and environment constraints," in *Intelligent Autonomous Systems, An Int. Conf.* Amsterdam: North-Holland Publishing Co., 1987, pp. 346–354.

[17] J.-P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Trans. Autom. Control*, vol. 10, no. 5, pp. 577–593, Oct. 1994.

[18] J. Barraquand and J.-C. Latombe, "On nonholonomic mobile robots and optimal maneuvering," in *Proc. IEEE International Symposium on Intelligent Control*, 1989, pp. 340–347.