

Distributed Grasp Synthesis for Swarm Manipulation with Applications to Autonomous Tugboats

Joel M. Esposito

Abstract— Assume a swarm of mobile robots is in the act of transporting a large object in the plane, by applying unilateral forces to the perimeter of that object. We address the question of where a new robot, joining the group, should establish contact with the object to maximally improve the manipulation capabilities of the swarm. Inspired by the literature on multi-fingered hands, we synthesize a grasp by incrementally optimizing a grasp quality function. We adapt the quality function in several important ways to accommodate the distributed nature of the swarm problem. We show that the objective function is quasi-concave, which has important implications for uniqueness and scalability of the solution; and present a solution methodology. We apply the resulting framework to the example of a large swarm of autonomous tug boats towing a barge, taken from our larger research program.

I. INTRODUCTION

Large groups of relatively simple mobile robots, known as *robot swarms*, can offer more capability, flexibility and robustness in certain applications as compared with using a single larger, special-purpose robot. One example of such an application is that of *non-prehensile cooperative manipulation*, where a group of non-articulated mobile robots attempts to transport a larger object in the plane, by applying forces to its perimeter. Applications include material transport, tug boats towing disabled ships, and search and rescue robots clearing rubble and moving victims to safety. In such applications it can be difficult to design a single robot large enough to achieve an enveloping grasp and powerful enough to complete the task. The advantages of the swarm are: (1) its ability to distribute applied forces over a large area; and (2) the maximum wrench the swarm can exert increases linearly as the number of swarm members increases.

The challenge in swarm manipulation lies in the fact that, unlike multi-fingered robot hands, each robot must select its contact point and applied forces independently – perhaps with limited information about the locations and actions of the other swarm members. Thus grasps must be synthesized and executed in a distributed fashion. Army ants transporting prey offer a proof of concept in nature [23] and studies have shown that they do not communicate directly while manipulating an object. Roboticians have begun to study their behavior [2].

J. Esposito is an Associate Professor of Systems Engineering at the U.S. Naval Academy, Annapolis, MD, USA, esposito@usna.edu, <http://www.usna.edu/Users/weapsys/esposito/>. This work was supported by ONR grant N0001405WRY20391.

Inspired by our work on autonomous tugboats [4], in this paper we address the problem of incrementally synthesizing planar grasp configurations for an N-robot swarm, as depicted in Figure 1. Specifically, we address the question of where a new robot, joining the group, should establish contact with the object in order to maximally improve the manipulation capabilities of the swarm. Once the robots are in contact with the object, we assume they have a method of attachment that enables them to apply bounded, unilateral forces, though we do not address force control here (see our group's other work [19]). In Section II we review related work. In Section III we give a formal problem statement. Section IV shows that this problem is equivalent to a quasi-concave optimization problem and discusses the implications of this. An example problem motivated by our autonomous tugboat project is solved in Section V.

II. BACKGROUND AND RELATED WORK

Related work can be partitioned into two categories: *decentralized approaches* used in swarming and flocking work, where each robot makes its own decisions; and *centralized approaches* used in manipulation, such as those used for multi-fingered hands controlled by a single processor.

A. Decentralized Approaches

Decentralized swarm control is an active area of research (ex. [15], [22] and [3]). However, those frameworks are limited to kinematic (*i.e.* position and velocity) objectives, such as collective motion (*i.e.*, flocking), sensor coverage and mapping tasks; ignoring forces and contact mechanics.

Cooperative object manipulation, both prehensile and non-prehensile ([18], [12]) has certainly been considered before in the context of small groups of robots (usually 1-3). Though it is generally unclear how to extend these to the distributed setting. Larger groups have been studied using a behavior-based framework ([7], [9], [16]), but no attempt at formal grasp synthesis was made. Two very closely related works on *distributed* swarm manipulation are [21], and [17]. Within these works, controllers were designed to force robots to surround the object. Termed “caging”, the inter-robot spacing was constrained to be small enough that it is impossible for the object to “escape”, meaning that as the robots move, so must the object. While this approach is decentralized, the primary difference with our work is that they essentially treat the task as a position control problem, ignoring the

dynamic forces, actuator limitations, contact mechanics and momentum, limiting its applicability to marine settings or heavy object transport. Our other work [20] addresses how, once the swarm members are in position, the applied forces should be selected to track a trajectory.

B. Centralized approaches

The literature on traditional grasping and manipulation with multi-fingered robot hands is vast. It addresses most of the mechanics issues mentioned above however it is difficult to apply to swarms for two reasons. First, most proposed applications involve a small number of contacts (usually 2-3, but no more than 5 fingers), which occasionally permit analytical solutions. Second, each of the contacts is controlled by a centralized decision maker, and centralized power supply. We review some fundamental concepts used in the sequel. A wrench is a generalized force. For planar grasps it takes the form $w = [F_x, F_y, M_z]^T$. Force closure is defined as the ability of a grasp to resist or apply a wrench in an arbitrary direction. Due to the unilateral constraint that pushing (pulling) forces are positive (negative), in order to achieve force closure the set of possible applied wrenches at all the contacts wrench must *positively span* \mathbb{R}^3 ; this condition is easily tested (see [14]). In the absence of motion constraints on the fingers, stating that the grasp is a force closure configuration, is equivalent to saying that the position and orientation of the object is small time locally controllable, under the constraint that the applied forces are nonnegative. It is a well known result that at least 4 contacts [14] are required to satisfy this condition for the class of objects and contact types considered here. Therefore, in this paper we assume that the swarm size $N \geq 4$.

Force closure grasps are not unique; so, once the closure criterion is met, a secondary *grasp quality function* can be defined. Common choices include “Max Normal Force” and “Min Analytic Center” [6], [11], which are important robustness measures for friction assisted grasping. In the case of the swarm manipulation problem considered here, we adopt the “Max Transfer” quality function, [8], [5], [13], [24], which measures the ability of the grasp to resist or apply an arbitrary net wrench. The approach used here, synthesizing a grasp by applying some numerical optimization method to the quality function, has been used in grasp analysis and synthesis before, see for example [10] and [11].

The primary differences between the work described here and related work on multi-fingered hands is the distributed nature of swarm operation. As such:

- the number of contacts is *significantly* larger than three;
- the quality function must use an L_∞ norm to take into account distributed actuator limitations;
- the placement of each contact is not coordinated by a centralized decision maker, although our approach does require knowledge of the positions of the entire swarm.

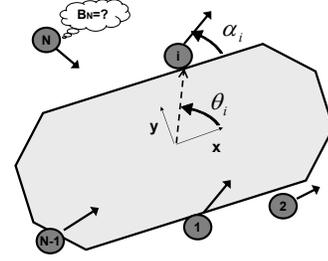


Fig. 1. Swarm Manipulation Scenario. $N - 1$ robots (dark circles) are attached to the object (shaded polygon); and can apply forces at some fixed incident angles. The position of the i^{th} robot is determined by angle θ_i and the push direction by the angle α_i . The N^{th} robot joining the group must compute its best attachment configuration.

III. PROBLEM STATEMENT

A. Notation: Contact Configuration Description

Assume the object to be manipulated (see Figure 1) is defined by a closed convex polygon $\mathcal{O} \subset \mathbb{R}^2$. Its boundary is denoted by $\partial\mathcal{O}$. Define a body-fixed coordinate frame (\hat{i}, \hat{j}) attached to the centroid of \mathcal{O} , called the *object frame*. Since \mathcal{O} is convex we can parameterize contact points on the boundary using the angle θ , measured counter clockwise, relative to the x-axis of the object frame $\{x(\theta)\hat{i} + y(\theta)\hat{j} \in \partial\mathcal{O}, \forall \theta \in \mathbb{S}^1\}$. Define a second parameter $\alpha \in \mathbb{S}^1$ which indicates the direction of a robot’s push force, measured counter clockwise relative to the x-axis of the object frame. Then the contact configuration of the i^{th} robot can be described by a vector

$$B_i(\theta_i, \alpha_i) = \begin{bmatrix} \sin \alpha_i \\ \cos \alpha_i \\ -y(\theta_i) \cos \alpha_i + x(\theta_i) \sin \alpha_i \end{bmatrix}$$

and a configuration of N robots is described by the matrix $B \in \mathbb{R}^{3 \times N}$, constructed by concatenating the column vectors B_1, \dots, B_N . The net wrench on the object, $w = [F_x, F_y, M_z]^T$ can be computed as $w = Bu$ where $u = [u_1, \dots, u_N]^T \in U$ is a vector of input *push force magnitudes* for each robot. The set U is described as $0 \leq u_i \leq u_{\max}, \forall i \in [1, N]$ and models the power limitations of the robots as well as the fact that we do not run the actuators in reverse.

Note that there is no constraint on the allowed push directions α_i , as there would be in the case of friction assisted grasping, since we assume the robots have some attachment method. These assumptions model how tugboats are operated in practice: they always “tie up” to the barge fixing their attachment point, They often have thrusters which allow them to select a precise thrust direction, the props are not run in reverse while pushing, and there is some limit on the safe maximum engine RPM.

B. Grasp Synthesis

We use the *Max-Transfer* grasp quality function [5], [13] defined as

$$Q(B) = \min_w \max_{u \in U | w = Bu} \frac{\|w\|}{\|u\|_\infty}. \quad (\text{III.1})$$

In words, the quality function measures the swarm's ability to apply, or resist, the worst-case net wrench, w , using the minimum u . Obviously a larger value of Q is desirable. Figure 2 provides a graphical interpretation.

It is very important to note that in the case of u , we select the L_∞ norm to reflect individual swarm members' power limitations. Selecting the appropriate norm on w is classically difficult due to a unit mismatch between forces and torques. Let $\|w\| = \sqrt{w^T A w}$, where $A = \text{diag}[a_1, a_2, a_3]$. Note that $a_1 \neq a_2$ can be used to weight certain preferential directions of motion over others. For example, in the tug boat application we can weight forward thrust more heavily than lateral thrust capability – reflecting the fact that most marine vessels are designed to move along the bow direction. Also, a_3 must be chosen to resolve the unit mismatch between forces and torques. A common choice is $a_3 = 1/r_{\max}^2$ where r_{\max} is $1/2$ the length of the principle axis of the object to be manipulated. With no loss of generality, we set A equal to the identity matrix for the remainder of this paper, in which case it reverts to the standard Euclidian norm.

Some simplification of eq. III.1 is possible. Since all norms are homogeneous, the ratio in eq. III.1 is not affected by pure scaling of w or u . Therefore, with no loss in generality we can consider the case when $\|u\|_\infty = 1$ vice u_{\max} , removing the denominator from consideration. Therefore an equivalent definition of the Max-Transfer function is as follows.

Definition 3.1: Max-Transfer Grasp Quality Function Let

$$Q(B) = \min_{w \in \partial W} \|w\|, \quad (\text{III.2})$$

where the *Wrench Set* $W = \{w | w = Bu, 0 \leq u_i \leq 1\}$ is the compact set of all wrenches that can be generated with a given configuration B , and ∂W denotes the boundary of that set. Note that this function can be interpreted as the maximum wrench that can be generated in the worst case direction.

C. Problem Statement

Assumptions: Robots $1, \dots, N-1$ are currently in contact with the object. Robot N wishes to establish contact with the object. It knows the object geometry, \mathcal{O} , the total number of robots in the swarm, N , and the actuator limitation u_{\max} . There is a wireless network that allows the robots to share information $((\theta_i, \alpha_i))$ as needed.

Problem: Distributed Grasp Synthesis (full information) Given the pushing angles $\alpha_1, \dots, \alpha_{N-1}$ and contact points $\theta_1, \dots, \theta_{N-1}$, compute the new swarm member's configuration vector B_N such that

$$\max_{B_N} Q(B_N; B_1, \dots, B_{N-1}). \quad (\text{III.3})$$

IV. APPROACH

To solve the grasp synthesis problem it is first necessary to understand the set W . Based on the properties of W we can develop an appropriate approach to maximizing Q .

A. Geometry of the Wrench Set

Since U is a closed convex polyhedral set, it follows that its image under B is a closed convex polyhedral set in \mathbb{R}^3 . W can be constructed by taking the Minkowski sum of the columns of B : $\{B_1, \dots, B_N\}$. *Zonohedra* are polyhedra described by Minkowski sums of a finite list of vectors, known as *generators*. They have the following important properties that will be exploited later:

- given N generator wrenches, the polyhedron has $P = N(N-1)$ facets in \mathbb{R}^3 , each of which is a parallelogram;
- the outward unit normals of the facets are

$$n_{jk} = \frac{B_j \times B_k}{\|B_j \times B_k\|}, \quad \forall j \neq k \in [1, \dots, N] \times [1, \dots, N];$$

- since $B_j \times B_k = -B_k \times B_j$ the polyhedron exhibits *central symmetry*; and
- the corresponding distances from the origin to the hyperplanes that comprise the facets are

$$d_{jk} = \sum_{i=1}^N \max(0, n_{jk} \cdot B_i).$$

- Also note that the origin is always contained in W . If it lies in the interior of W it implies that B is a force closed grasp. Else, it lies on the boundary and some $d_{jk} = 0$.

Eq. III.3 asks us to find the smallest wrench on the boundary of W , which is equivalent to finding the closest point on the surface of the polygon to the origin. It is easily shown that the closest point on a convex polyhedra to a point in its interior lies on a facet, not a vertex. Therefore,

$$\begin{aligned} Q(B) &= \min_{jk \in N \times N} d_{jk} \\ &= \min_{jk \in N \times N} \sum_{i=1}^N \max(0, \frac{B_j \times B_k}{\|B_j \times B_k\|} \cdot B_i). \end{aligned} \quad (\text{IV.1})$$

B. Maximizing Q

A function is said to be *quasi-concave* if all its super-level sets are concave. It is a generalization of concavity that includes functions whose derivatives may be zero at some non-extremum points (*i.e.* plateaus). Most importantly, like concave functions, they have a single global maximum; and they are well conditioned to numerical solution techniques.

Proposition 4.1: The objective function Q , defined in eq. IV.1 is quasi-concave in the variable B_N . The rather inelegant proof of this fact is relegated to the Appendix.

Remark 4.2: Q is not strictly concave because, due to the $\max(0, *)$ term in eq. IV.1. When B_N is not in the same half-plane as the outward normal of the closest facet of the wrench

Algorithm 1 Computing grasp B_N to max $Q(B)$.

Compute $Q(B_1, \dots, B_{N-1})$ and normal of closest facet n^* .
 Set $Q_{lower} = Q(B_1, \dots, B_{N-1})$,
 Set $Q_{upper} = Q_{lower} + \|B_N\|_{\max}$, set $Q_{bisect} = Q_{upper}$
while $Q_{upper} - Q_{lower} < \epsilon$ **do**
 Feasible? Find B_N^* such that $Q > Q_{bisect}$
 if Feasible **then**
 $Q_{lower} \leftarrow Q_{bisect}$
 else
 $Q_{upper} \leftarrow Q_{bisect}$
 end if
 Set $Q_{bisect} = (Q_{upper} + Q_{lower})/2$
end while
 Determine attachment (θ_N, α_N) to physically realize B_N^* .

polygon generated from $\{B_1, \dots, B_{N-1}\}$, $\partial Q / \partial B_N = 0$ even though this is clearly not the maximum. In fact, the optimal B_N vector *must* lie in the same half-plane as the outward normal of the closest facet. This fact is used generate initial guesses for the optimization algorithm out lined below.

C. Algorithm

Remark 4.3: Since the objective function is quasi-concave, there are no equality constraints, and the feasible set for B_N is convex, a unique global maximum can be efficiently found.

The optimization method outlined in Algorithm 1 is used to compute a global maximizing value for B_N to an arbitrary tolerance, ϵ . It is based on the quasi-convex optimization method in [1], which in turn is based on the classic bisection method. Bisection begins with an upper and lower bound of the objective function, which in this case are easily computed. It follows that it computes the maximizing B_N in exactly $\log_2(\|B_N\|_{\max}/\epsilon)$ iterations. $\|B_N\|_{\max} = \sqrt{1 + r_{\max}^2}$ is the largest wrench that a single robot can apply with a unit force if r_{\max} is the distance to the point on the object furthest from the center of mass. Note that the initial guess Q_{bisect} is not actually the bisector, because we found that in practice the global optimum frequently occurs when B_n is normal to the closest facet. The feasibility query: *Find B_N , Subject To*

$$Q(B) \geq Q_{bisect}, \quad \| [B_N^x, B_N^y] \| \leq 1, \quad B_N^z \leq r_{\max},$$

is described in detail [1] for the interested reader. Note that the normal to the closest facet, n^* provides a good initial guess for this problem.

V. EXAMPLES

Test Bed Example: We are interested in the application of a team of autonomous tug boats towing a disabled ship. Figure 3 shows our experimental apparatus which consists of 6 thrusters (each representing tug boat) that can be mounted at different locations and orientations along the hull of a scale

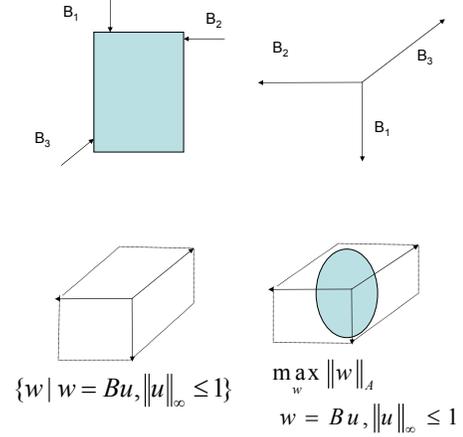


Fig. 2. Graphical interpretation of the wrench polyhedra.

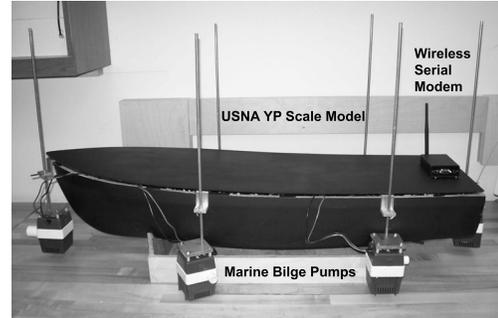


Fig. 3. An experimental vessel with six re-mountable thrusters. The boat is a 1 meter scale replica of a US Navy Training Vessel. The thrusters are made from submersible marine bilge pumps.

model of a Navy training vessel (1 meter long). The thrusters provide forces in the range of $[0, 0.5]$ Newtons.

Figure 4 (left) depicts a simplified rectangular model of the object, originally with with 5 robots in contact. The blue lines terminating in circles indicate the contact location and orientation. For this example, robots are constrained to pull in an outward direction relative to the contact facet. The optimization method was used to compute the optimal configuration of the 6th robot, indicated with a red line terminating in a star. Figure 4 (right) shows the original wrench polyhedra W (shaded gray), along with the optimized wrench polyhedra (dark wire frame). Originally $Q = 0$ (not force closed), and the optimized value is $Q = 0.39$ – demonstrating that the method automatically generates force closed grasps when possible. The example executed in 0.816 seconds. All examples in this paper were solved on a *P4* desktop, using MATLAB, version *R2006a*, and the optimization toolbox. Computation times do not include the time required to render the graphics.

Large Scale Example: Figure 5 can be interpreted in an identical manner, except that it involves 15 robots initially.

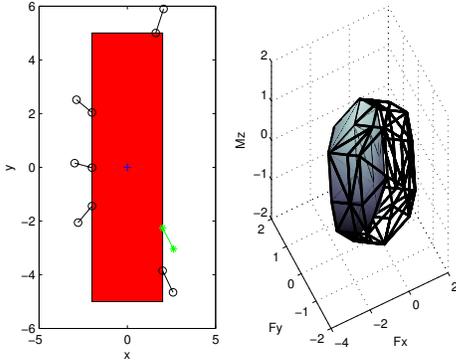


Fig. 4. Grasp synthesis example with 6 robots. (Left) A group of 5 robots (blue/circle line segments) grasps a rectangular object. A sixth robot computes its optimal contact point (green/ * line segment). (Right) The original wrench polyhedra (shaded gray) and the optimized wrench polyhedra (wire-frame).

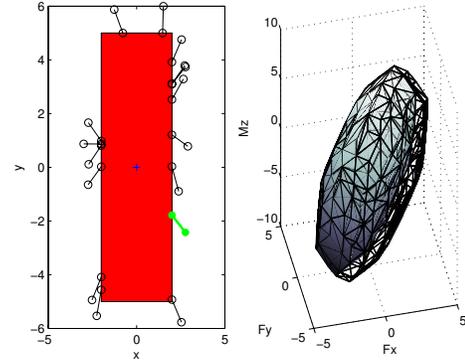


Fig. 5. (Left) A group of 15 robots (blue/circle line segments) grasps a rectangular object. A 16th robot computes its optimal contact point (green/* line segment). (Right) The original wrench polyhedra (shaded gray) and the optimized wrench polyhedra (wire-frame).

Originally $Q = 2.26$, the optimized value is $Q = 2.67$. The example executed in 2.72 seconds.

Figure 6 plots the computation time required to compute the optimum for 10 randomly generated configurations for each swarm size from 5 to 25 robots. The thick red line shows the median computation time (sec) vs. swarm size; the upper dashed blue line the worst-case computation time; and the lower dashed blue line the best-case time. Overall computation time is $O(N^2)$, as expected since the number of algorithm iterations is independent of swarm size, while the time per iteration is dominated by the evaluation of the objective function which is an “all pairs” computation. In our opinion, this is expected in a swarm cooperation scenario requiring global knowledge. Curve fits of the form $T_{comp} = CN^2$ were computed; the constants were $C_{min} = 0.0059$, $C_{med} = 0.0104$, and $C_{max} = 0.1841$ implying that, typically the algorithm has positive scalability attributes. The best-case to median times occur when the optimum value is at or near the initial guess (B_N is normal to the closest facet of W); the worst case times appear to be relatively isolated outliers.

VI. CONCLUSIONS

In this paper we consider the problem of synthesizing grasps for a swarm of mobile robots looking to cooperatively transport polygonal objects in the plane. Specifically we address the question of where a new robot, joining the group, should establish contact with the object in order to maximally improve the manipulation capabilities of the swarm. The contributions of this paper are as follows.

- We cast the grasp synthesis problem to reflect the distributed nature of the swarm’s actuation capabilities.
- We showed that the objective is quasi-concave, meaning a global maximum can be computed in a finite number of iterations.

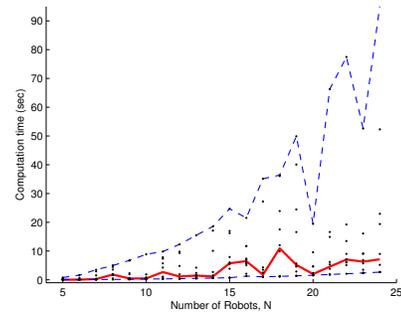


Fig. 6. Best-case, worst-case and median computation time vs. swarm size.

- We presented a numerical solution algorithm and show it automatically incorporates the force closure criteria.
- Regarding the scalability of the method, the best case run time is nearly constant, while the median runtime scales as $O(N^2)$.

Future work will focus on two objectives. First, we plan to complete a full experimental demonstration of the autonomous tug boat swarm, preliminary efforts are illustrated in Figure 7. Second, the author feels that one of the most unrealistic assumptions in the paper is that Robot N knows the exact pose of the remaining robots in the swarm – primarily because it requires full information sharing between the robots, adversely affecting scalability, but also because it is unlikely that each robot would have an accurate estimate of its own position to share. Given a set valued estimate $\hat{B}_1, \dots, \hat{B}_{N-1}$ of the poses, the author intends to pursue a min-max strategy to compute the N^{th} robot’s optimal contact point under the worst-case configurations of the remaining robots.

APPENDIX: PROOF THAT Q IS QUASI-CONCAVE

First note that $\max(0, \frac{B_j \times B_k}{\|B_j \times B_k\|} \cdot B_i)$ is quasi-concave in each B vector; and the objective function eq. IV.1 consists of the minimum of the sum of a

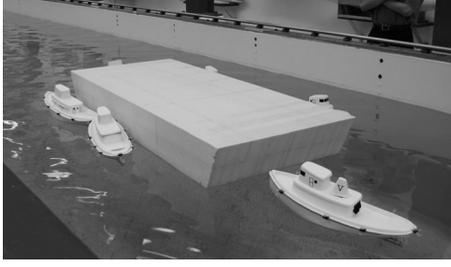


Fig. 7. Experimental test bed. A group of 6 unmanned tugboats (0.5 meters long) and a scale model flat bottomed barge (2 meters long). The tugs have articulated magnetic attachment devices used to grab the barge.

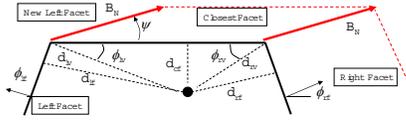


Fig. 8. A cross section of the new wrench polytope.

set of quasi concave functions. Unfortunately, in general, the sum of quasi-concave functions is not quasi-concave – making an algebraic proof difficult using that formulation. However the facet distances can be rewritten without using a repeated summation. Note a function is quasi-concave iff it is quasi-concave when restricted to any line segment intersecting its domain. To that end, consider the 2-D cross section of the polytope $W \subset \mathbb{R}^3$ in Fig. 8. We look at the individual d_{ij} 's and show they are quasi-concave in $\theta = \text{dir}(B_N)$ and $\|B_N\|$. Also note, the addition of B_N to the list of generators can change the distance to any facet if $B_N \cdot n_{ij} > 0$ but it suffices to consider only the four cases shown in the picture. Let CF denote the closest facet to the origin and RF , LF and RV , LV the facets and vertices to the immediate right and left along the cross section. A quantity with a prime, denotes the new value after taking the Minkowski sum with B_N . θ is measured CCW with respect to CF . The addition of B_N changes the following quantities:

$$d'_{cf} = d_{cf} + \|B_N\| \max[0, \cos(\theta - \pi/2)] \quad (\text{VI.1})$$

$$d'_{(r,l)f} = d_{(r,l)f} + \|B_N\| \max[0, \cos(\theta - \phi_{(r,l)f})], \quad (\text{VI.2})$$

which are all quasi-concave in both the magnitude and direction, θ . In addition, it also adds a new facet on either the left or right side

$$d'_{n(r,l)f} = d_{(r,l)v} \sin(\theta + \phi_{(r,l)v}). \quad (\text{VI.3})$$

The new right facet only appears when $0 < \theta \leq \phi_{r,f} - \pi/2$; the left when $\phi_{l,f} + \pi/2 < \theta < \pi$. \sin is concave when restricted to this range. Therefore, all of these functions are quasi-concave, as seen in Fig. 9, and the min of a set of quasi-concave functions is quasi-concave.

REFERENCES

- [1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [2] M. Egerstedt, T. Balch, F.Dellaert, F. Delmotte, and Z. Khan. What are the ants doing? Vision-based tracking and reconstruction of control programs. In *IEEE International Conference on Robotics and Automation*, pages 4182 – 4187, 2005.
- [3] J.M. Esposito and T.W. Dunbar. Maintaining wireless connectivity constraints for swarms in the presence of obstacles. In *IEEE Conference on Robotics and Automation*, pages 946–952, 2006.
- [4] M. Feemster, J. Esposito, and J. Nicholson. Manipulation of large objects by swarms of autonomous marine vehicles. In *IEEE Southeastern Symposium on Systems Theory*, pages 255–259, 2006.
- [5] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE International Conference on Robotics and Automation*, pages 2290–2295, 1992.

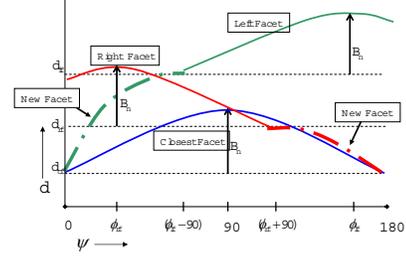


Fig. 9. Distance to planar facet extensions vs. the orientation, θ , of B_N .

- [6] G.Liu, J.Xu, and Z.Li. On quality functions for grasp synthesis, fixture planning, and coordinated manipulation. *IEEE Transactions on Automation Science and Engineering*, 1(2), 2004.
- [7] A.J. Ijspeert, A.Martinoli, A. Billard, and L.M. Gambardella. Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2):149–171, 2001.
- [8] D. Kirkpatrick, B. Mishra, and C.K. Yap. Quantitative Steinitz theorem with applications to multifingered grasping. *Discrete and computational geometry*, 7(3), 1992.
- [9] R. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1-2):85–101, 2000.
- [10] L.Han, J.C.Trinkle, and Z.X.Li. Grasp analysis as a linear matrix inequality problem. *IEEE Transactions on Robotics and Automation*, 17(6):663–674, 2000.
- [11] G.F. Liu and Z.X. Li. Real-time grasping force optimization for multifingered manipulation: Theory and experiments. *IEEE/ASME Transactions on Mechatronics*, 9(1), 2004.
- [12] K.M. Lynch. Locally controllable manipulation by stable pushing. *IEEE Transactions on Robotics and Automation*, 15(2):318–327, 1999.
- [13] B. Mirtich and J. Canny. Easily computable optimum grasps in 2-D and 3-D. In *IEEE International Conference on Robotics and Automation*, pages 739–747, 1994.
- [14] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [15] R. Olfati-Saber and R. Murray. Flocking with obstacle avoidance: cooperation with limited communication in mobile networks. In *IEEE Conference on Decision and Control*, 2003.
- [16] C.A. Parker, H. Zhang, and R. Kube. Blind bulldozing: Multiple robot nest construction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2010–2015.
- [17] G. Pereira, V. Kumar, and M. Campos. Decentralized algorithms for multi-robot manipulation via caging. *International Journal of Robotics Research*, 2004.
- [18] D. Rus. Coordinated manipulation of objects in the plane. *Algorithmica*, 19(1):129–147, 1997.
- [19] E. Smith, M.G. Feemster, and J.M. Esposito. Swarm manipulation of an unactuated surface vessel. In *IEEE Southeastern Symposium on Systems Theory*, pages 16 – 20, 2007.
- [20] E. Smith, M.G.Feemster, and J.M. Esposito. Cooperative manipulation on teh water using a swarm of autonomous tugboats. In *IEEE International Conference on Robotics and Automation*, 2008.
- [21] A. Sudsang and J. Ponce. A new approach to motion planning for disc shaped objects in the plane. In *IEEE Conference on Robotics and Automation*, pages 1068–1075.
- [22] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents, Part I: Fixed topology. In *IEEE Conference on Decision and Control*, 2003.
- [23] Edward O. Wilson. Ants the civilized insect. *National Geographic*, 210(2):136–149, Aug 2006.
- [24] X.Y. Zhu and J. Wang. Synthesis of force-closure grasps on 3-D objects based on the Q-distance. *IEEE Transactions on Robotics and Automation*, 19(4):669–679, 2003.