Reactive Robotics I: Reactive Grasping with a Modified Gripper and Multi-fingered Hands

Marek Teichmann Bud Mishra

September 1998 (Last Modified March 2000)

Abstract

We study the problem of grasping an unknown object with constant cross section using various "reactive" robot hands. In the simplest example, we equip a standard parallel jaw gripper with several light-beam sensors (close to each jaw) and implement a reactive algorithm for grasping polygonal objects with this architecture. Extending these ideas further, we also devise two and three fingered reactive hands for objects with smooth boundary and equip these with distance and angle sensors that are located at the finger tips.

The sensors used are simple, provide only limited and immediate information, but allow us to reactively find a good grasp on an object of unknown geometry and dynamics. When grasped the forces applied will be normal to the object boundary. Furthermore, in all cases, the object is not disturbed as the grasping points are being sought.

Various approaches to dextrous manipulation can be categorized into two groups: *object-priority approaches* and *finger-priority approaches*. This classification was first proposed by Fearing. In the first case, one assumes an accurate model of the object to be grasped and from such a model, an off-line geometric algorithm determines a set of grip points, where the fingers are then placed. As already well-known, this approach has found extensive usage in [MSS87, PSS$^+$93] for example. Once the grip points have been determined, the geometry of the object is deemed irrelevant and the grasp is determined and maintained by only controlling the magnitudes of the forces at the grip points. This approach has provided a clear and deep understanding of stable grasps, how their existence depends on the nature of contact and the physical complexity of grasping, etc. Nevertheless, this approach has proven less useful in practice, as obtaining an accurate model of the object might not be feasible, or the exact location of the object might not be available. On the other hand, the main success in dextrous manipulation seems to have come from two directions: (1) telemanipulation, where a human in the loop uses much more sensory information than what is assumed theoretically necessary, and (2) simple parallel jaw grippers, where grasping algorithms can be made immune to lack of any sensory information, thus suggesting a need for an extensive study of the "robot algorithms" for grasping under the finger-priority model. For example Goldberg [GF93, Gol91] constructed a parallel jaw gripper with the property that friction in the transversal direction between a jaw of the gripper and the grasped object is removed (or greatly reduced.) A similar "sensorless" strategy has been used by Rao [RG94] to determine the shape of the object being grasped, by making a series of grasp actions using the above device and measuring the final distance between the gripper jaws.

Here, we suggest another approach, where the sensory information is selectively used to determine movement of the robot so that ultimately a grasp is achieved independent of the quality of the sensory information or the reliability of the actuator movements. Furthermore, unlike Goldberg's algorithm, our algorithms and devices are non-disturbing in the sense that it does not cause any movement of the objects except for the movements desired and can grasp objects that are kinematically constrained.

Our "algorithms" are rather different from what is usually understood by algorithms: the robots (i.e. actuators and sensors) themselves are active components of the algorithm, performing "computations" much like an analog computer in conjuncts with a digital computer. For this reason, the "algorithm design" also implies a careful choice of the participating actuators and sensors and their placement. Furthermore, the algorithm does not separate in any meaningful way phases of "digital planning" interrupted by "analog actuation and sensing". Thus our reactive algorithms are much smoother in operation, and significantly robust to error in sensing. The need for such "robot algorithms," incorporating controllability and observability issues, was very clearly articulated by Latombe [Lat95].

We shall describe applications of these ideas to the design of a reactive parallel-jaw gripper and reactive multi-fingered hands. These devices are rather simple and can be built with off-the-shelf components. For instance, the reactive parallel jaw gripper consists of a standard parallel jaw gripper and two infrared emitting diodes (LEDs) and two infrared detectors attached to each jaw as depicted in Figure 1.

The rest of the paper is organized as follows: In section 1, we describe what we mean by a reactive algorithm and what advantages it holds over the traditional approaches. Section 2 and 3 describe the theoretical underpinnings and a practical implementation of a parallel jaw reactive gripper, respectively. Section 4 and 5 describe the three and two fingered reactive planar hands, respectively. The final section, concludes with some questions and reflections on the future of reactive robotics.

1 What is a Reactive Algorithm?

By a reactive algorithm, we mean a simple algorithmic scheme where a robot's sensors determine immediately the actions of the actuators. However, note that the actuators themselves may interact with the sensors (e.g., by moving them or occluding them, etc.) to close a feedback loop and thus causing further goal-driven as well as corrective actions. The application described here is original, although the underlying concept is not and can be traced back to the work of Norbert Wiener as described in his classic book "Cybernetics" [Wie65].

In the simplest picture, a set of sensors determine a binary vector, where each bit of the vector may represent whether a sensor detects a signal or not ("on/off sensor"), whether the value detected by one signal is higher than another or a predetermined threshold, etc. Our designs rely only on sensors that can solve such simple geometric tasks as whether a line intersects an object ("probes"), whether an object is within a small predetermined distance ("sniffers"), whether a line is normal to an object at a given point ("whiskers"), etc. Each possible value of the binary vector determines the movement of a single actuator immediately. The necessary actions for a given value of the vector are expressed in terms of a simple table with one entry per vector value.

Other than these simple computing schemes the "algorithm" has no other memory or computing abilities. However, in practice we allow ourselves to "cheat" by doing rather simple computations (e.g., arcsine of an angle) or by remembering a few bits. In theory, while a given task could be achieved by a "purely reactive" algorithm, the resulting algorithm may cause inefficiency or awkwardness purely for the sake of being faithful to a theory.

2 Reactive Parallel Jaw Grasping

In this section we consider a standard parallel jaw gripper and attach to each jaw two infrared light emitting diodes (IR LEDs) and two infrared detectors as depicted in Figure 1. This allows us to have in fact 4 light beams which we can test for being interrupted or not by alternately illuminating the IR LEDs. The general scheme will be to close the gripper while constantly checking the status of the sensors, and performing actions according to that status in a table driven manner.

These algorithms can be used as low level primitives for grasping applications, for example in automated manufacturing. With minor modifications, the algorithms can also be used for determining the shape of (the convex hull of) a planar part, again without disturbing the part. In contrast, Rao's scheme takes a multitude of gripping actions for obtaining information about the shape, and even then, it is not possible to

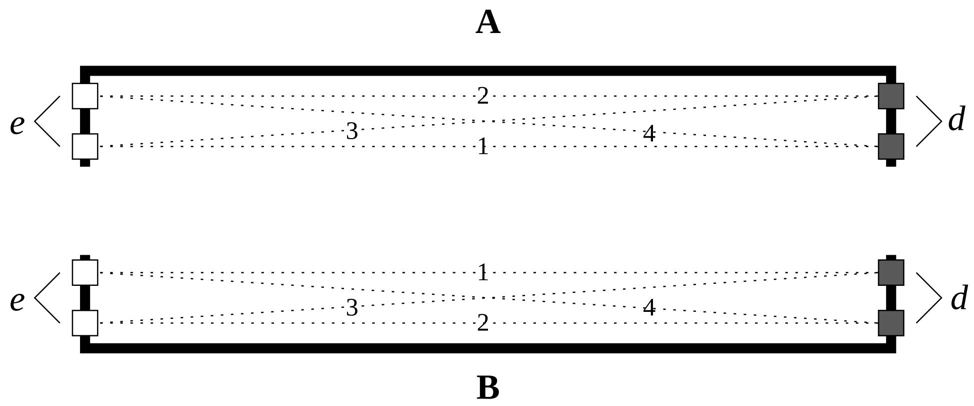

Figure 1: Parallel jaw gripper with sensors. *Legend*: **A**=Jaw A, **B**=Jaw B, **e**= (Emitter) IR (infrared) LED and **d**= (Detector) IR phototransistor. The controller of this reactive gripper determines each primitive motion of the gripper and its components (e.g., jaws **A** and **B**) based on the states of the infrared beams (dotted lines labeled 1, 2, 3 and 4) between the emitters and the detectors. The geometry of the object and the state of the gripper determine which infrared beams are interrupted and based on this information, the reactive controller produces a controlled movement of the gripper. Eventually, gripper detects a state in which closing the parallel jaws yields a stable grasp of the object.

reconstruct the shape in some cases. This is due to the fact that the object is moved during each grasping action.

2.1 Related Literature

Our work shares some similarities with Goldberg's gripper [Gol91, Gol93] and is motivated by the same paradigm as Canny and Goldberg's "RISC" (Reduced Intricacies in Sensing and Control) robotics [CG94].

In [Gol91, Gol93], Ken Goldberg has modified the basic gripper to reduce the friction between the gripper and the object being grasped. This friction occurs when the gripper is being closed on an object, and this object is forced to rotate. Several results exist on finding grasps for such a parallel jaw gripper, when the object is known [RG94, RG92]. Recent results consider closing the gripper on an unknown object, and measuring the jaw distance for various orientations [Rao93]. This scheme permits the determination of the object, but also subjects it to many motions. Another method involves grasping in random orientations to distinguish planar parts [Gol91, Gol93]. For some related designs with simple light-beam sensors, also see [Ake96, AM98].

In the Computational Geometry literature, there are several results on probing: Probing using finger probes [CY87], and line and other probes [DEY86], for instance, with the objective of constructing a model of the probed object. We are mainly interested in line probes: a line is swept (either in parallel or rotating) until the object is hit. The probe returns the position of the line where it hit the object first. Also the method of "rotating calipers" [Tou83] where a pair of parallel supporting lines rotates around a polygon has many algorithmic applications in geometry, and is closely related to the algorithm described here.

A reactive grasping algorithm for a special type of multi-fingered hand is given in [MLH93], but it uses a gripper mounted camera and a vision system. Like ours, their algorithm controls the position of the gripper interactively according to the current sensory data, but the algorithms are inherently more complex.

In this paper, we add probing capabilities to a basic parallel jaw gripper and to two finger and three finger hands, and combine probing techniques with grasping using light beam sensors and proximity sensors. The goal is to produce algorithms for obtaining grasps with as little knowledge as possible about the object, and without touching the object. First, we present grasping algorithms for the extended parallel jaw gripper equipped with light beam sensors as described below.

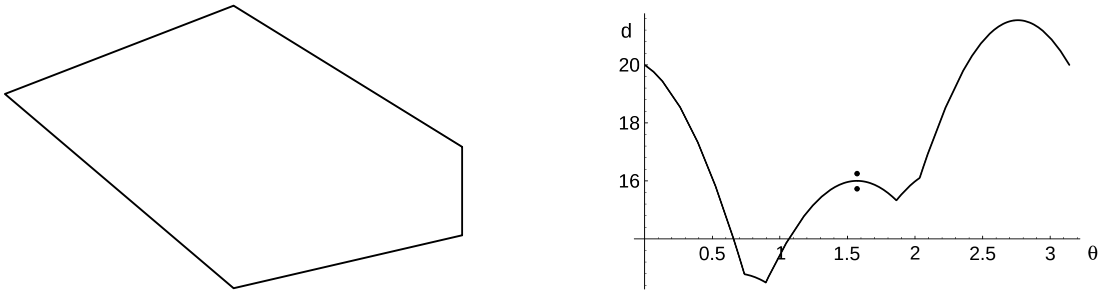

Figure 2: A polygon and its diameter function.

2.2 The Parallel Jaw Algorithm

2.2.1 The Diameter Function

Consider a fixed polygonal object P, and define the function $d(\theta)$ to be the diameter of P in the direction θ. This is defined as the length of the projection of P (a line segment) on a line that forms an angle of θ with the x-axis. Antipodal pairs of points correspond to local maxima, and local minima correspond to different stable grasps. Figure 2 shows a polygon and its diameter function.

The current configuration of the gripper consists of

- The gripper orientation,
- The inter-jaw distance, and
- The position of (the center of) the gripper.

The position of each pair of parallel beams can be represented by a point in the graph of the diameter function. Rotations of the gripper correspond to moving the point parallel to the θ axis (horizontally), and opening and closing of the gripper to moving vertically. The four pairs of beams can be represented by four points in a '+' (cross) configuration. This last representation is not fully precise since the translational component of the beam positions is not modeled, but is nevertheless useful for our analysis.

2.2.2 The Algorithm

We assume that the initial orientation of the gripper is at angle 0 (with the x-axis, say), and that the gripper rotates around its center of symmetry.

Algorithm 1

State	A_2	A_1	B_1	B_2	*Action*
0	.	.	.	.	close the gripper
1	.	x	.	.	move up
2	.	.	x	.	move down
3	.	x	x	.	rotate in current direction
4	x	x	x	.	move up (parallel to jaw before previous rotation)
5	.	x	x	x	move down (parallel to jaw before previous rotation)
6	x	x	.	.	move up
7	.	.	x	x	move down
8	x	x	x	x	On first entry: reverse rotation direction, back-up Otherwise: stop

Let A_1, A_2, A_3, A_4 be the state of the light beams on jaw A, and B_1, B_2, B_3, B_4 be the same for jaw B. In the following, *up*-ward translation means translation perpendicular to the gripper jaws and towards jaw A, *down* is towards jaw B. This also defines naturally *left* and *right* translation. Also we have a *current direction*, which is clockwise initially, and the initial position is such that the object is somewhere between the gripper jaws.

We need another primitive: *Back-up* will move the gripper to the position it was before performing the previous rotation.

The algorithm checks the current state and performs an action according to the table above, where an 'x' indicates that the corresponding beam is interrupted. The action is performed until the state changes. Since each action is performed in a tight control loop, the resulting motions are very small. We assume that initially the object being grasped is somewhere between the parallel jaws.

The algorithm essentially attempts to keep the top of the object boundary between beams 1 and 2 of one jaw, and the bottom between the corresponding beams of the other jaw (state 3), and rotates the gripper while the diameter decreases.

For example, at the beginning we are in state 0 and the gripper closes until either state 1 or 2 is reached. Then a small translation occurs and immediately we are back in state 0. An initial direction of rotation is chosen arbitrarily, which might be reversed in state 8.

Let D_{12} be the sum of the distances between beams 1 and 2, for each jaw. Let θ be a local minimum of the diameter function d, and consider the plane containing the graph of the diameter function. Define a *valley of height h around θ* as the set of points

(x, y) that are above the graph of d satisfying:

$$\max(d(x), d(\theta)) \leq y \leq d(\theta) + h$$

$$\max_{x<\theta}\{d(x) \leq d(\theta)\} \leq x \leq \min_{x>\theta}\{d(x) \leq d(\theta)\}.$$

The constraint on x says that the valley stops after d dips below $d(\theta)$. See Figure 3. Then we have the following

Theorem 2.1 *Consider a polygonal object with constant horizontal cross-section and with a diameter significantly smaller than the length of the jaws. Assume infinitesimally small sensor resolution (D_{12}). For such an object, Algorithm 1 will terminate with the reactive gripper in a small neighborhood of the closest local minimum (c.l.m.) orientation. The c.l.m. orientation is defined as a local minimum orientation θ of the diameter function $d(\theta)$, that is closest to the initial orientation, and in the direction of initial decreasing slope.*

Furthermore, in the limit as $D_{12} \to 0$, the final gripper orientation converges to c.l.m. orientation.

Proof. At each step, we either close the gripper or translate/rotate so that the gripper can close further. We can think of this algorithm as tracking the diameter curve using two vertically positioned test point, at a distance of D_{12} apart. The bottom point corresponds to the inner pair of beams, the top point to the outer pair, and we want to keep the curve between the two points (state 3). When the bottom point gets above the curve we are in state 0, and we close (go down in the diameter function graph). When the top point moves below the curve we are in state 8. States 1,2,4,5,6,7 take care of translating the gripper. This is not modeled by the diameter function.

To show that the algorithm converges to a local minimum, we show that at any step we make progress. In state 0 we advance down in the diameter function, and in state 3 we advance in the chosen direction. This direction can change only once, when the initial direction causes an increase in the diameter graph. The other states, except state 8, reduce the number of interrupted beams by translating the gripper in an appropriate direction and hence eventually allow us to get to state 0 or 3. States 4 and 5 need some explanation. These are reached after a rotation. (We assume that the gripper closes with sufficient precision so that beam 1 is always interrupted before beam 2 and we detect both events separately.) It can be seen that if we translate parallel to the jaws as they were before the previous rotation, we bring the object closer to the center of the gripper.

State 0 and 3 may increase the number of interrupted beams but we make progress by closing the gripper or rotating further, respectively. When state 8 is reached for

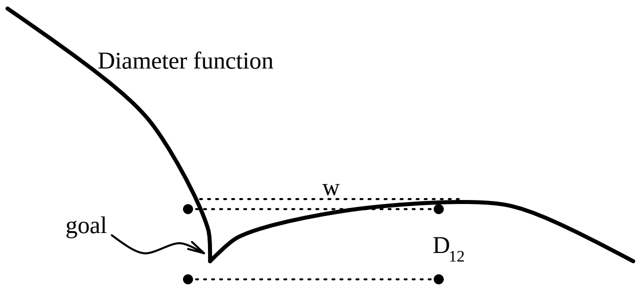

Figure 3: Angular error due to distance between beams.

the first time, it might be because we have chosen a wrong initial direction, and the diameter increases during that rotation. This case is the only one where we rotate with all 4 beams interrupted. State 8 can only be entered from states 3, 4 and 5. If this is the first entry to state 8, we simply retrace our steps (back-up) to the state just before this state was entered. We can now perform rotations again, but in the other direction. Step 8 thus reverses the search direction and continues. Upon second entry to state 8 we are certain of having crossed a local minimum, and the diameter function increases by at least D_{12}, and we are done. Note that the terminating orientation is not necessarily the one corresponding to the closest local minimum of $d(\theta)$, but some orientation in the valley of height D_{12} containing the c.l.m. orientation. □

This algorithm will miss any "small" local minima whose height is smaller than D_{12}. This is the limit of the *resolution* of the gripper. As we decrease D_{12} to zero, the algorithm misses smaller and smaller local minima, and ultimately tracks the diameter curve with total precision. This is physically impossible, so we accept a small error.

This error implies however that we do not have any maximum angle error guarantee for any small but fixed value of D_{12}, since we can overshoot by a large angle if the diameter function increases slowly, see Figure 3. For example for an oval-like polygonal shape, if the difference between the minimum and the maximum diameter is too small, we might miss the minimum.

2.2.3 Improving accuracy

We previously had an alternative algorithm which would allow us to improve accuracy. It was based on a binary search for the local minimum. Another algorithm used the crossed beams to control the rotation direction. This work has been reported in [TM94b]. However in the course of experimentation, we have found that it is possible to obtain a "virtual" pair of parallel beams using a *single* LED–detector pair. Furthermore, these virtual beams can be made as close to each other as desired, subject to the precision of the analog to digital converters which are used with the IR detectors to obtain the strength of the IR radiation coming from the LED. This step is implemented in practice by having various thresholds for the received intensity (see the section on implementation.) Later we discuss our implementation of this algorithm using this virtual pair of light beams. In this new scheme, accuracy as described does not pose a problem.

2.2.4 Oscillations

It is conceivable that at some point due to inaccuracies or time delays, the algorithm would oscillate between a pair of "opposite" states (e.g., 1 and 2, or 4 and 5). This could also happen if the beam distance is exactly the diameter of the polygon in the current orientation. This can be avoided by checking if no gripper closing has occurred between those states and then forcing the algorithm to go to state 0 (close).

3 An Implementation

3.1 The Setup

To verify the practicality of the algorithm for reactive grasping developed above, we have constructed a prototype of the parallel jaw gripper with sensors and implemented the reactive algorithm. The gripper was mounted in the chuck of a robotic milling machine, MOSAIC, developed at New York University [Ash90]. A picture of the gripper alone and mounted on MOSAIC can be seen in Figure 4.

The MOSAIC machine is a three-axis, knee-type milling machine with pulse-width modulation motor drivers. It has a horizontal table which can move horizontally in two (xy) directions, and a spindle which can move up or down (in the z direction), and rotate around a vertical axis. Its motion controllers are installed in a VME chassis housing a 68020 CPU running the `VxWorks` real-time operating system. The controllers can also servo the machine tool spindle. This is the special feature of MOSAIC which allows us to use it instead of a robot arm. The only other difference is that it is the table which moves, instead of the arm.

We augmented the existing software to contain an implementation of the algorithm of Section 2.2. The motion primitives which we used were only absolute and relative moves, for the table and for the spindle, and an *abort* command which stopped the currently executing move command.

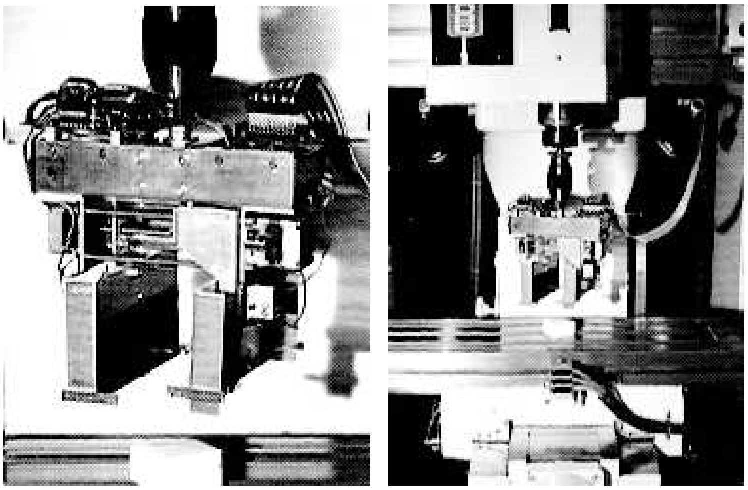

Figure 4: Reactive parallel jaw gripper.

3.1.1 The Reactive Gripper

Our gripper consists of three main components. A Motorola MC68332BCC single board micro-controller, a custom interface board between the micro-controller and the gripper hardware itself. The micro-controller is connected through a serial line to the VME computer.

The gripper itself consists of two stepper-motor driven platforms or 'jaws' which can move *independently*. Let us assume that the motion occurs in the xy-plane. This is in contrast with most existing grippers, and can be used to advantage to reduce the number of interactions between the gripper micro-controller and the robot. It also allows more precise control over jaw position relative to the object to be grasped. Figure 5 illustrates the design and shows some of the dimensions.

On each jaw a pair of infrared LED's is mounted on one side, and a pair of infrared photo-transistors is mounted on the other side, as depicted in Figure 6.

This setup allows us to implement four light-beams, two parallel and two crossed, as depicted in Figure 1. This implementation allows us to turn on each LED in sequence and read the level of infrared radiation arriving at the photo-transistors using an analog to digital (A/D) converter. In the current implementation, we do not use the crossed beams. A patent for this gripper and the algorithm is held by NYU [TM96].

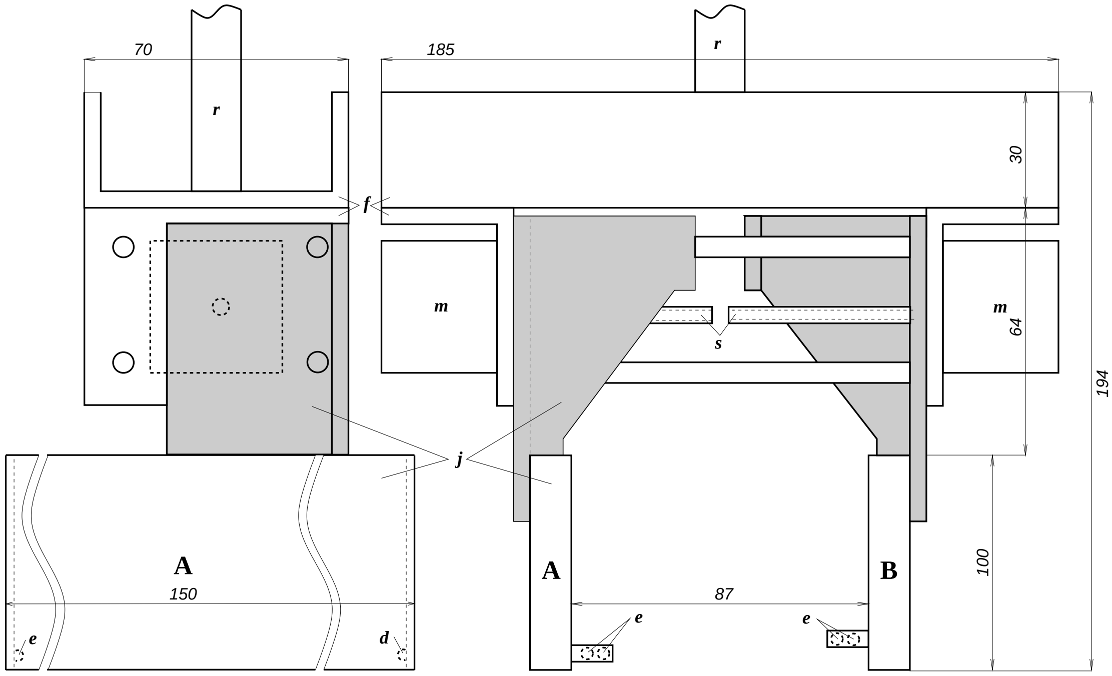

Figure 5: The reactive gripper. All dimensions are in mms. *Legend*: **A**=Jaw A, **B**=Jaw B, **e**= (Emitter) IR (infrared) LED, **d**= (Detector) IR phototransistor, **f**= Gripper frame, **j**= Jaws, **m**= Stepper motor, **s**= Stepper motor screw, and **r**= Spindle. The side view is shown on the left and the front view on the right. The parallel jaws **A** and **B** slide linearly on the stepper motor screw **s** and independently controlled by the stepper motors (labeled **m**); the entire assembly rotates about the spindle **r**. The emitter **e** and detector **d** are attached to the jaws and the state of the infrared beam (either interrupted or uninterrupted) emitted by **e** determines the motion of the actuators (stepper motor **m**, and spindle **r**). Together they provide the necessary reactive controller for this simple parallel jaw gripper that can grasp an unknown object of constant cross section. (Also see Figure 1 for more details.)

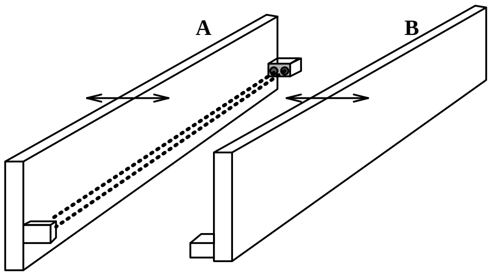

Figure 6: Light beam setup.

3.2 The Experiment

In this experiment, the distance between the two light beams on each jaw was quite large (0.8mm). The main difficulty encountered was a non-trivial amount of jitter in the spindle axis, and was primarily due to the fact that the controller for this axis is optimized for torque control and not position control. The jitter caused the light beams to constantly change states, which is a problem only in state 8. Averaging successive readings and incorporating hysteresis solved the problem. We kept a log file which contained, for each state transition, the x, y coordinates (in inches in the coordinate system of the machine tool), as well as the orientation of the gripper (in revolutions), and a number indicating the width of the opening between the jaws (in the number of step-motor steps). Figure 7 shows graphs corresponding to the above parameters, as a function of the transition number, for one run of the algorithm. The first figure shows the initial position of the object with respect to the gripper. The initial direction of rotation was clockwise. We can see that there is an initial approach where the gripper (actually the table) is only translating (state 1)—this corresponds to the straight horizontal segment in the position graph. This phase is followed by a set of rotations with some translations, and then state 8 is entered for the first time and a reversal occurs, at transition number 159. Finally the gripper quickly reaches the other side of a valley in the diameter graph when the second transition (number 215) to state 8 occurs. We ran a number of similarly successful experiments on differently shaped objects.

4 The Case of Three Fingers.

Next, we consider objects of constant horizontal cross-section, as in Section 2.2, but we assume in addition that the object is convex and that its boundary is smooth. We present a reactive algorithm for a three finger hand using a different set of sensors than

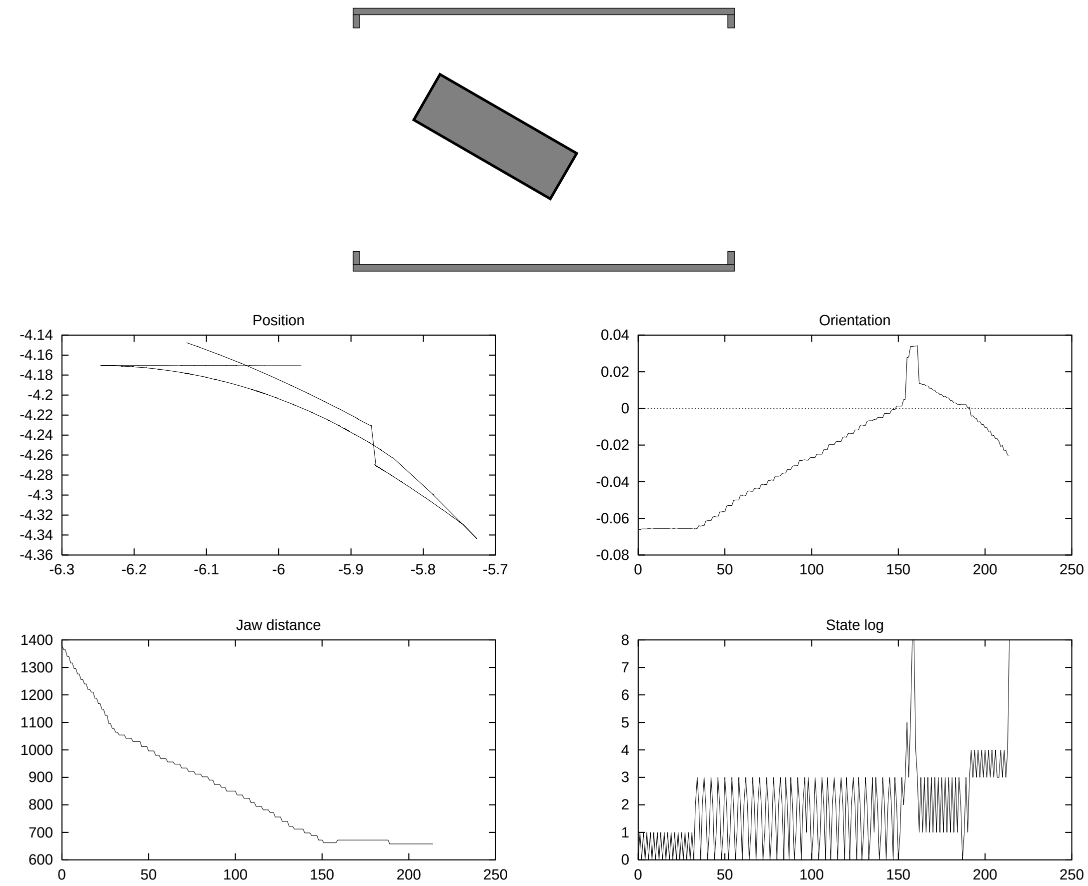

Figure 7: One run of the reactive algorithm. In the diagrams, position is given as $(x,\ y)$-coordinates in inches as the gripper moves. The other diagrams show how the orientation of the gripper (in revolutions), jaw-distance (motor steps) and state of the reactive controller as a function of time.

above. Again, we use the local geometry of the object to find a set of grip points and we also solve the local motion planning problem of getting the fingers to their grasping positions. Our algorithm is as before non-disturbing.

Mirtich and Canny [BM94] presented algorithms for finding two and three finger grasps for known objects, optimizing various criteria. It is easy to adapt our ideas to produce reactive algorithms for generating their planar grasps reactively. Independently, Erdmann [Erd94] has devised a closely related algorithm for two finger grasps which uses sensors similar to ours; although, the motivations for our work differ considerably from those of Erdmann, there is an interesting convergence of final design methodologies.

Here, we consider a gripper that consists of three fingers whose endpoints move in a plane. The fingers can move to arbitrary positions within certain bounds, but their order around the triangle they form remains the same. We search for the gripping points by following the object boundary in a way that is dictated by the boundary until some geometric conditions are satisfied, which will then guarantee a grasp.

Each finger is equipped with simple sensors, as described in the next section, which allow the fingers to follow the contour of the object to be grasped, and locally determine the angle of the object boundary it is close to.

4.1 Sensors

Each finger is equipped with the following sensors:

1. An omni-directional distance sensor, which returns the distance of an object in any (planar) direction. Its range can be very small, only what is required for tracking the boundary of an object that lies within close proximity to the finger.

2. An angle sensor. If the finger is close to an object as defined by the previous sensor, we require the sensor to return the angle of the object boundary at the closest point.

These sensors can be implemented by using for example an array of distance sensors similar to the optical sensor of Okada [Oka82] or the sensor of [BSB93]. Preliminary experiments indicate that it is possible to use a pair of simple IR reflective sensors, placed a small distance apart, and measure the difference between reflected light intensity.

4.2 Primitives

In order to describe our algorithm at a high level we assume that the following primitive operation can be called upon. It consists of following the object boundary by a finger. This primitive has already been implemented on various robotic platforms, see for example [Jen92].

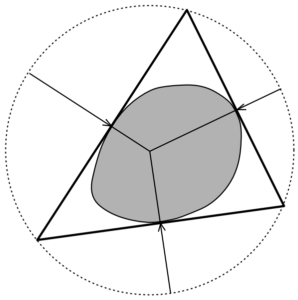

Figure 8: Forces applied at edge midpoints meet.

4.3 The Three Finger Reactive Algorithm

Our algorithm for finding a three fingered grasp is based on the paradigm of finding a locally minimal area triangle that encloses the object. A geometric algorithm for convex polygons can be found in [OAMB86].

Theorem 4.1 (Klee [Kle86]) *If T has a locally minimum area among triangles containing a convex body B, then the midpoint of each side of T touches B.*

It can also be shown [OAMB86] that if the midpoint of an edge e of a triangle does not touch the object, then e can be perturbed such that its midpoint after the perturbation lies inside the original triangle (i.e., the one preceding the perturbation), and that this perturbation reduces the triangle area. This is precisely how the algorithm operates. In essence we minimize a potential function defined by the triangle area, whereas in Section 2.2, we minimize the distance between the two jaws of the parallel jaw gripper.

Given such a triangle, we note that the lines through its edge midpoints and perpendicular to their corresponding edges are concurrent at the point which is the center of the circle circumscribing the triangle. See Figure 8. If we place the three fingers at the edge midpoints, we get a planar force closure grasp that does not rely on friction. The finger forces will be chosen such that they sum to 0. Planar torque closure is obtained, if we allow for friction.

We now describe the algorithm. Let B be the object to be grasped. For simplicity of exposition, when we say a finger is at point p of the boundary, it is in fact in close proximity of the object, and does not touch it. (It is on the boundary of a slightly enlarged object. The point p will be the closest point to the finger on the boundary of the original object.) The tangent to the object at that point is still well defined, by convexity. Note that given three such finger positions, the sensors at the fingers indicate the object tangents, so we know the triangle they form.

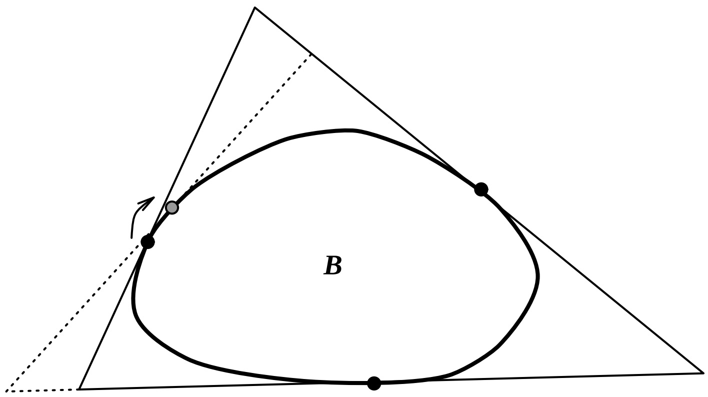

Figure 9: Steps of the algorithm.

- **Phase 1:** In the first phase we find some triangle that contains the object. We assume the object lies somewhere "between" the fingers, such that when we close the fingers, say along three concurrent lines at angles of 120 degrees with each other, each finger comes in proximity with the object.

  Let f_0, f_1, f_2 be the corresponding points that lie on the boundary of B. Let t_0, t_1 and t_2 be the object tangents at those points. If the tangents form a triangle the first phase is done. Otherwise we pick an unbounded edge and "rotate" it by movings its finger along B in the unbounded direction. This procedure must lead to a finite triangle. It is also possible to arrive at a triangle with all but one edge midpoints touching, by keeping the distance of say f_1 and f_2 to t_0 the same.

- **Phase 2:** In the second phase we find a locally minimal area triangle enclosing B. The procedure is simple: since we are looking only for a local minimum in the area, we can use local methods. Each finger divides its triangle edge into two segments. Consider the ratios of the largest segment to the total edge length. This number is between 1/2 and 1, and is 1/2 if the finger is at the edge midpoint. Pick the edge whose finger is proportionally the furthest from the midpoint, i.e. whose ratio is furthest to 1/2. Then move the corresponding finger along the object boundary, in the direction of the largest segment. Clearly, this move will reduce the ratio. Let p_0 be the initial finger position, and p_1 be the position after the move. If the shape is strictly convex at p_0, the move also reduces the area of the triangle, since the p_1 is now located inside the original triangle [OAMB86]. See Figure 9.

  We iterate this step until all fingers are at their edge midpoints. At this point the normal lines at the finger points are concurrent and we found a grasp. Convergence is guaranteed by the fact that the area of the triangle decreases during every move, so an infinite loop does not occur.

4.4 Dealing With Uncertainty

After some point the algorithm will reach the limit of the sensor resolution and will oscillate between various nearby positions close to the local minimum. This can be solved by accepting a small error in the finger positions so that they are close to the edge midpoints. The cases where the algorithm will be the most sensitive to errors will be when the triangle is very skinny. Then a small motion of a finger will have larger consequences on the edge length and its relative position. However, in the presence of some degree of friction, we simply assume that the friction is sufficient to allow for the forces to be concurrent, irrespective of sensor angular error.

This algorithm does not work for non-convex objects since for such objects the midpoint of an edge of a locally minimum area triangle does not necessarily lie on the object boundary. It might still be possible to use the reactive algorithm locally, and we plan to investigate this in a sequel. If we allow the fingers to grasp at object vertices, and if we consider that at a vertex all the possible normals between the normals of the adjacent edges are present, then our algorithm can be used on polygonal objects.

4.5 A Variation

Mirtich and Canny's three finger algorithm finds offline the largest equilateral triangle containing the object and the fingers are placed at the triangle edge contact points. This optimizes resistance to torques normal to the grip plane. A reactive algorithm for finding such three points can be described as follows. Again we start with all three fingers somewhere on the boundary. Fix f_0, and move the other two fingers around the boundary until the tangents form an equilateral triangle. Then rotate all the fingers around the boundary, keeping the angles between their normals 120 degrees, until the normal at f_0 has made a rotation for 120 degrees. Then move back to the position giving the largest area formed by the triangle t_0, t_1, t_2.

5 Two Finger Algorithm

This section describes a simple reactive algorithm for finding a two finger grasp using proximity sensors. The hand model is as follows. The two fingers slide on an axis that is parallel to the grasping plane, and let the midpoint of the two fingers be the center of the gripper. Call the *finger normal* the direction from the finger to the midpoint. Here we assume similar sensors as previously but we do not require angle measurement,

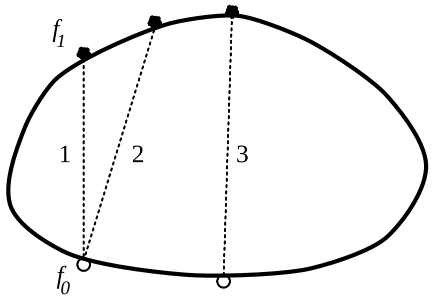

Figure 10: A few steps of the two finger algorithm.

only the information on whether the angle formed by the inward pointing normal to the object at the finger position and the finger normal is 0, positive or negative. Again, we assume that the object has a smooth boundary and is convex.

There must exist a pair of points on the boundary of the object such that the normal lines at both points are in fact the same line. This fact can be seen by considering the pair of supporting lines, one on each side of the object, and rotating this pair until it reaches a local minimum. Then the contact points satisfy the above requirement.

We find such a pair of points as follows (refer to Figure 10.) We assume that initially the object is somewhere between the two fingers. The fingers close until one of the sensors indicate proximity. Call this finger f_0, the other f_1. Then the gripper is rotated until the finger normal of f_0 is normal to the object. Such a rotation will in absolute value be by less than 90 degrees. If during this rotation f_1 becomes close to B, then it starts to track the boundary of B in the direction away from f_0. Otherwise, close the fingers along their axis (translating the center if necessary) until f_1 becomes close to B as indicated by its sensor.

Now to find the grasping points as described above, we "rotate" the gripper by following B's boundary using f_0, keeping its axis normal to B, and f_1 also following B. Then the grasp is found when f_1's normal is also normal to B. This situation must happen by the existence of a local minimum of the distance between supporting lines on each side of the smooth and convex object, as commented earlier. Note that we do not need to rotate by more than 180 degrees.

In this case also, there is a two finger reactive algorithm for finding the optimal grasp in the sense of Mirtich and Canny [BM94]. We can find the pair of points such that the distance between them is maximized by rotating as above by 180 degrees and noting all the grasp positions and the distance between the fingers for each grasp.

6 Concluding Remarks

Here, we have shown how parallel jaw grippers and multi-fingered hands can be augmented with simple sensors so that reactive algorithms implemented on these architectures can provide stable grasps robustly, efficiently and with minimal computing requirements. Our experience with the parallel jaw gripper has demonstrated the feasibility of the reactive schemes. Nevertheless, it remains to be seen how two- and three-fingers reactive algorithms perform in practice, although based on our experience with the reactive gripper device, we are optimistic that these and other reactive robots hold enormous promise. Also many variations are possible. For example, we can replace the proximity and angle sensors by force sensors, but we lose the non-disturbing quality.

It would be interesting to find algorithms that require less sensory information, for example ones that require only the sign of the angles at the fingers instead of the exact angle for the three finger case. This would make the algorithm more robust and simplify the sensors. Generalization of these algorithms to non convex objects would be desirable.

Finally, theorem 4.1 is in fact valid in any dimension, (if we replace midpoint by centroid of a facet). Therefore it seems likely that these techniques can be extended to three dimensional objects and four fingers.

References

[Ake96] S. Akella. Robotic manipulation for parts transfer and orienting: mechanics, planning and shape uncertainty. Robotics Inst. TR: CMU-RI-TR-96-38, Robotics Institute, Carnegie Mellon University, 1996.

[AM98] S. Akella and M.T. Mason. Parts orienting with partial sensor information. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[Ash90] S. Ashley. A mosaic for machine tools. *Mechanical Engineering*, pages 38–43, September 1990.

[BM94] J. Canny B. Mirtich. Easily computable optimum grasps in 2-d and 3-d. In *IEEE International Conference on Robotics and Automation*, pages 739–747, San Diego, CA, May 1994.

[BSB93] A. Bonen, K.C. Smith, and B. Benhabib. Development of a robust electro-optical proximity sensor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 989–990, Yokohama, Japan, July 1993.

[CG94] J. Canny and K. Goldberg. 'RISC' for industrial robotics: Recent results and open problems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1951–1958, 1994.

[CY87] R. Cole and C. K. Yap. Shape from probing. *J. Algorithms*, 8(1):19–38, March 1987.

[DEY86] D. P. Dobkin, H. Edelsbrunner, and C. K. Yap. Probing convex polytopes. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 424–432, 1986.

[Erd94] M. Erdmann. Understanding action and sensing by designing action-based sensors. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994. Workshop on Design of Parts and Devices for Automation.

[GF93] K.Y. Goldberg and M.L. Furst. Low friction gripper, a design modification for the parallel-jaw gripper that improves grasp stability for polyhedral parts. U.S. Patent Number 5,186,515, February 1993.

[Gol91] K.Y. Goldberg. *Stochastic Plans for Robotic Manipulation*. Ph.D. thesis, School Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1991.

[Gol93] K.Y. Goldberg. Orienting Polyogonal Parts without Sensors. *Algorithmica*, 10(2):201–225, 1993.

[Jen92] J. Jennings. Model acquisition for near-sensorless manipulation using mobile robots. Invited talk, December 1992.

[Kle86] V. Klee. Facet centroids and volume minimization. In *Fejes Tòth Festschrift*, 1986.

[Lat95] J.-C. Latombe. Robot algorithms. In *Algorithmic Foundations of Robotics*, A.K. Peters, 1995.

[MLH93] T. G. Murphy, D. M. Lyons, and A. J. Hendriks. Stable grasping with a multi-fingered robot hand: A behavior-based approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 867–874, Yokohama, Japan, July 1993.

[MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(4):541–558, 1987.

[OAMB86] J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin. An optimal algorithm for finding minimal enclosing triangles. *J. Algorithms*, 7:258–269, 1986.

[Oka82] T. Okada. Development of an optical distance sensor for robots. *International Journal of Robotics Research*, 1(4):3–14, 1982.

[PSS+93] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet. On characterizing and computing three and four finger closure grasps of polyhedral objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 821–827, May 1993.

[Rao93] A. S. Rao. *Algorithmic Plans for Robotic Manipulation*. Ph.D. thesis, University of Southern California, Los Angeles, CA, 1993.

[RG92] A. Rao and K. Goldberg. Grasping curved planar parts with a parallel jaw gripper. Technical Report 299, IRIS, August 1992.

[RG94] A. S. Rao and K. Y. Goldberg. Shape from diameter: Recognizing polygonal parts with a parallel-jaw gripper. *IRob*, 13(1):16–37, 1994.

[Tei95] M. Teichmann. *Grasping and Fixturing: a Geometric Study and an Implementation*, Ph. D. Thesis, NYU, 1995.

[TM94a] M. Teichmann and B. Mishra. Reactive Algorithms for Grasping Using a Modified Parallel Jaw Gripper, In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, ICRA 94, 1994.

[TM94b] M. Teichmann and B. Mishra. Reactive algorithms for 2 and 3 finger grasping. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Grenoble, (France), 1994.

[TM96] M. Teichmann and B. Mishra. Reactive robotic gripper. US patent #5,541,485 awarded to NYU, July 1996.

[Tou83] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON '83*, pages A10.02/1–4, 1983.

[Wie65] N. Wiener. *Cybernetics or Control and Communication in the Animal*. MIT Press, June 1965.