MCB – *Industrial Robot* Feature Article

The BarrettHand grasper –
programmably flexible part handling and assembly

Abstract

This paper details the design and operation of the BarrettHand BH8-250, an intelligent, highly flexible eight-axis gripper that reconfigures itself in real time to conform securely to a wide variety of part shapes without tool-change interruptions. The grasper brings enormous value to factory automation because it: reduces the required number and size of robotic workcells (which average US$90,000 each – not including the high cost of footprint) while boosting factory throughput; consolidates the hodgepodge proliferation of customized gripper-jaw shapes onto a common programmable platform; and enables incremental process improvement and accommodates frequent new-product introductions, capabilities deployed instantly via software across international networks of factories.

Introduction

This paper introduces a new approach to material handling, part sorting, and component assembly called “grasping”, in which a single reconfigurable grasper with embedded intelligence replaces an entire bank of unique, fixed-shape grippers and tool changers. To appreciate the motivations that guided the design of Barrett’s grasper, we must explore what is wrong with robotics today, the enormous potential for robotics in the future, and the dead-end legacy of gripper solutions.

For the benefits of a robotic solution to be realized, programmable flexibility is required along the entire length of the robot, from its base, all the way to the target workpiece. A robot arm enables programmable flexibility from the base only up to the

Industrial Robot: An International Journal
Volume 27 – Number 3 – 2000 – pp. 181—188
© MCB University Press – ISSN 0143-991X – PDF made with MCB permission
<http://www.mcbup.com/research_registers/aa.asp>
<http://www.emerald-library.com>

by William T. Townsend
President & CEO
Barrett Technology, Inc.
<http://www.barrett.com/robot/>

toolplate, a few centimeters short of the target workpiece. But these last few centimeters of a robot must adapt to the complexities of securing a new object on each robot cycle, capabilities where embedded intelligence and software excel. Like the weakest link in a serial chain, an inflexible gripper limits the productivity of the entire robot workcell.

Grippers have individually-customized, but fixed jaw shapes. The trial-and-error customization process is design intensive, generally drives cost and schedule, and is difficult to scope in advance. In general, each anticipated variation in shape, orientation, and robot approach angle requires another custom-but-fixed gripper, a place to store the additional gripper, and a mechanism to exchange grippers. An unanticipated variation or incremental improvement is simply not allowable.

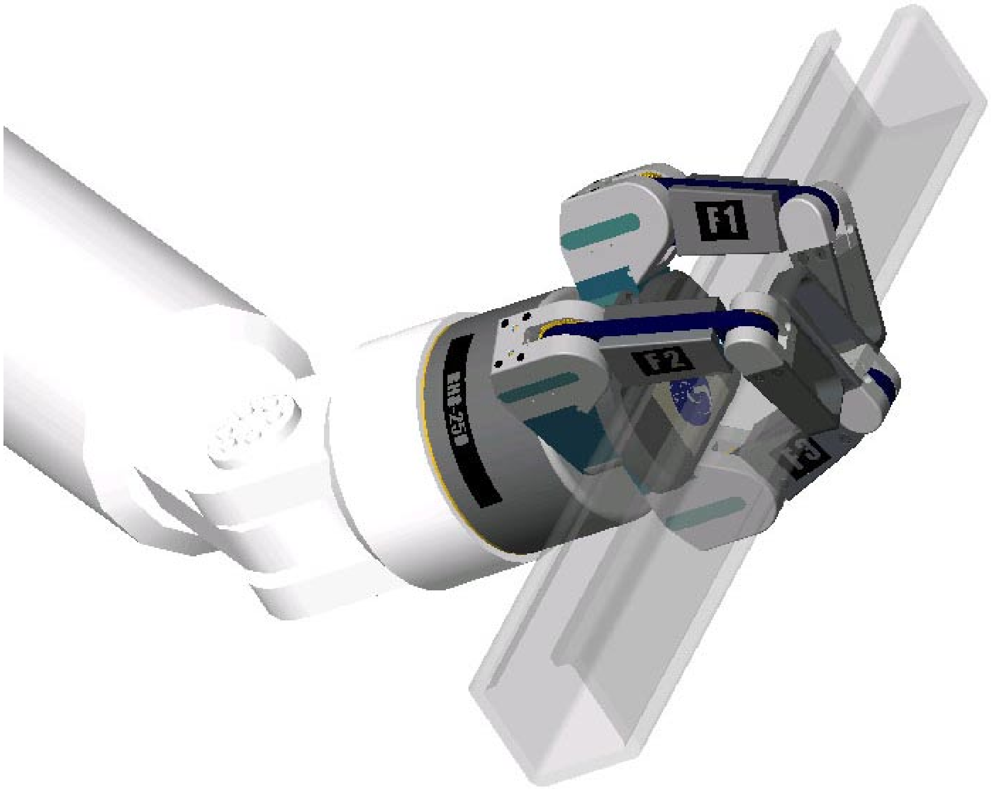

Figure 1 Graspers automatically conform to any part shape in any orientation

By contrast, the mechanical structure of Barrett’s patented grasper, illustrated in Figure 1, is automatically reconfigurable and highly programmable, matching the functionality of virtually any gripper shape or fixture function in less than a second without pausing the workcell throughput to exchange grippers.

For tasks requiring a high degree of flexibility such as handling variably shaped payloads presented in multiple orientations, a grasper is more secure, quicker to install, and more cost effective than an entire bank of custom-machined grippers with tool changers and storage racks.

For uninterrupted operation, just one or two spare graspers can serve as emergency backups for several workcells, whereas one or two spare grippers are required for each gripper variation – potentially dozens per workcell. And, it’s catastrophic if both gripper backups fail in a gripper system, since it may be days before replacements can be identified, custom shaped from scratch, shipped, and physically replaced to bring the affected line back into operation. By contrast, since graspers are physically identical, they are always available in unlimited quantity, with all customization provided instantly in software.

Gripper legacy

Most of today’s robotic part handling and assembling is done with grippers. If surface conditions allow, vacuum suction and electromagnets can also be used, for example in handling automobile windshields and body panels. As part sizes begin to exceed the order of 100gms, a gripper’s jaws are custom shaped to ensure a secure hold. As the durable mainstay of handling and assembly, these tools have changed little since the beginning of robotics three decades ago.

Grippers, which act as simple pincers, have two or three unarticulated fingers, called “jaws”, which either pivot or remain parallel during open/close motions as illustrated in Figure 2. Well organized catalogs are available from manufacturers that guide the integrator or customer in matching various gripper components (except naturally for the custom jaw shape) to the task and part parameters.

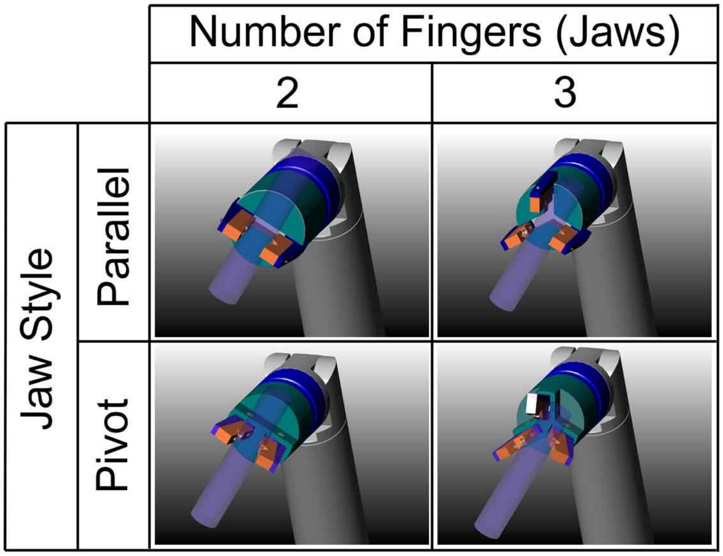

Figure 2 Gripper variations are limited

Payload sizes range from grams for tiny pneumatic grippers to 100+ kilograms for massive hydraulic grippers. The power source is typically pneumatic or hydraulic with simple on/off valve control switching between full-open and full-close states. The jaws typically move 1cm from full-open to full-close. These hands have two or three fingers, called "jaws". The part of the jaw that contacts the target part is made of a removable and machinably soft steel or aluminum, called a "soft jaw".

Based on the unique circumstances, an expert tool designer determines the custom shapes to be machined into the rectangular soft-jaw pieces. Once machined to shape, the soft-jaw sets are attached to their respective gripper bodies and tested. This process can take any number of iterations and adjustments until the system works properly. Tool designers repeat the entire process each time a new shape is introduced.

As consumers demand a wider variety of product choices and ever more frequent product introductions, the need for flexible automation has never been greater. However, rather than make grippers more versatile, the robotics industry over the past few years has followed the example of the automatic tool exchange technique used to exchange CNC-mill cutting tools.

But applying the tool-changer model to serial-link robots is proving expensive and ineffective. Unlike the standardized off-the-shelf cutting tools used by milling machines, a robot tool designer must customize the shape of every set of gripper jaws — a time-consuming, expensive, and difficult-to-scope task. Although grippers may seem cheap at only US$500 each, the labor-intensive effort to shape the soft jaws may cost several times that. If you multiply that cost times a dozen grippers as in the example above and throw in a toolchanger and tool-storage rack for an additional US$10,000, the real cost of the "few-hundred-dollar" gripper solution balloons to US$20,000 to US$60,000.

To aggravate matters, unknowns in the customization process confound accurate cost projections. So the customer must commit a purchase order to the initial installation fee on a time and materials basis without guarantee of success or a cost ceiling. While priced at US$30,000, intelligent graspers are not cheap. However, one can "customize" and validate the process in software in a matter of hours at the factory in a single day. If the system does not meet performance targets, then only a day's labor is wasted. If the system succeeds, then there are not any hidden expenses following the original purchase order.

Beyond cost, the physical weight of tool changer mechanisms, located at the extreme outer end of a serial-link robotic arm, limits the useful payload and dynamic response of the entire system. The additional length of the toolchanger increases the critical distance between the wrist center and payload center, degrading kinematic flexibility, dynamic response, and safety.

Description of the BarrettHand

Flexibility and durability in a compact package

The flexibility of the BarrettHand is based on the articulation of the eight joint axes identified in Figure 3. Only four brushless DC servomotors, shown in Figure 4, are needed to control all eight joints, augmented by intelligent mechanical coupling. The resulting 1.18kg grasper is completely self-contained with only an 8mm diameter umbilical cable supplying DC power and establishing a two-way serial communication link to the main robot controller of the workcell. The grasper's communications electronics, five microprocessors, sensors, signal processing electronics, electronic commutation, current amplifiers, and brushless servomotors are all packed neatly inside the palm body of the grasper.

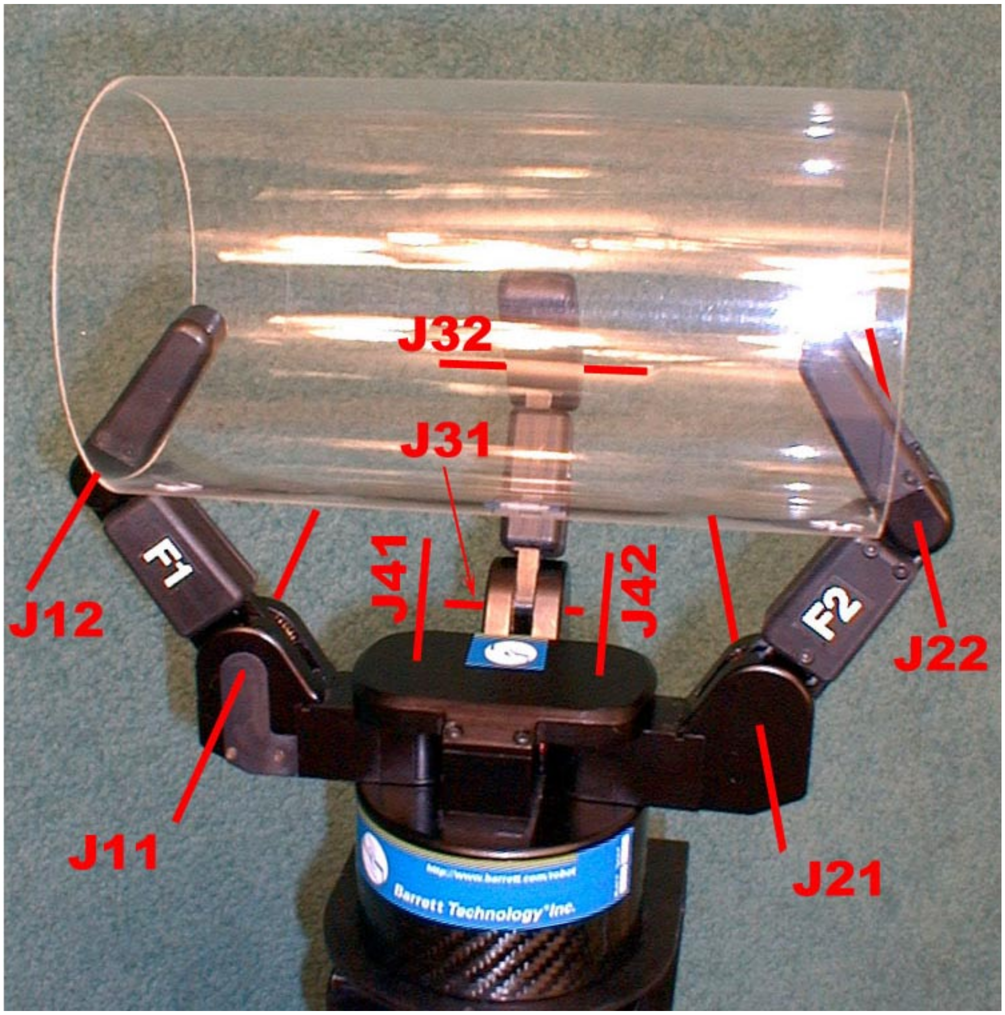

Figure 3 Eight axes of the BarrettHand

The BarrettHand has three articulated fingers and a palm as illustrated in Figure 5 which act in concert to trap the target object firmly and securely within a grasp consisting of seven coordinated contact vectors — one from the palm plate and one from each link of each finger.

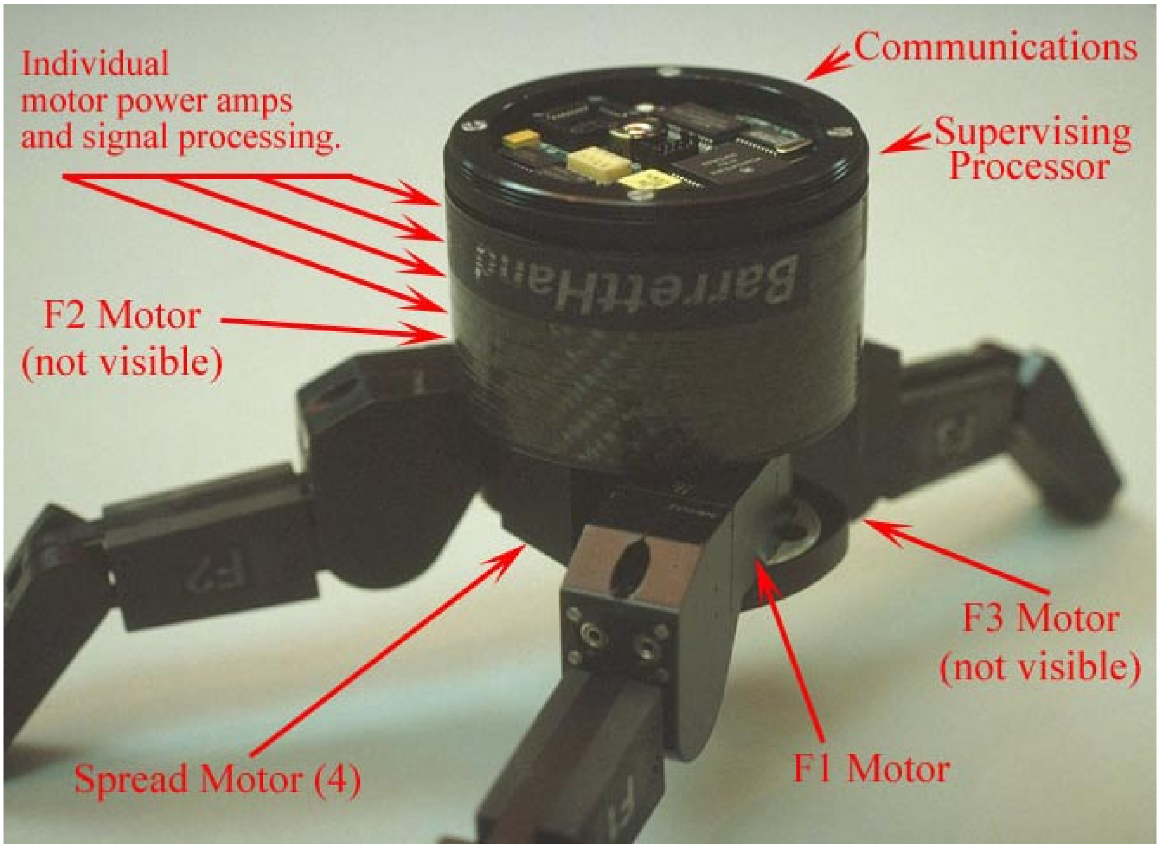

Figure 4 Motor locations in the BarrettHand

Each of the BarrettHand's three fingers is independently controlled by one of three servomotors as shown in Figure 6. Except for the spread action of fingers Fl and F2, which is driven by the fourth and last servomotor, the three fingers, Fl, F2, and F3, have inner and outer articulated links with identical mechanical structure.

Each of the three finger motors must drive two joint axes. The torque is channeled to these joints through a patented, TorqueSwitch mechanism (Figure 7), whose function is optimized for maximum grasp security. When a fingertip, not the inner link, makes first contact with an object as illustrated in Figure 8, it simply reaches its required torque, locks both joints, switches off motor currents, and awaits further instructions from the microprocessors inside the hand or a command arriving across the communications link.

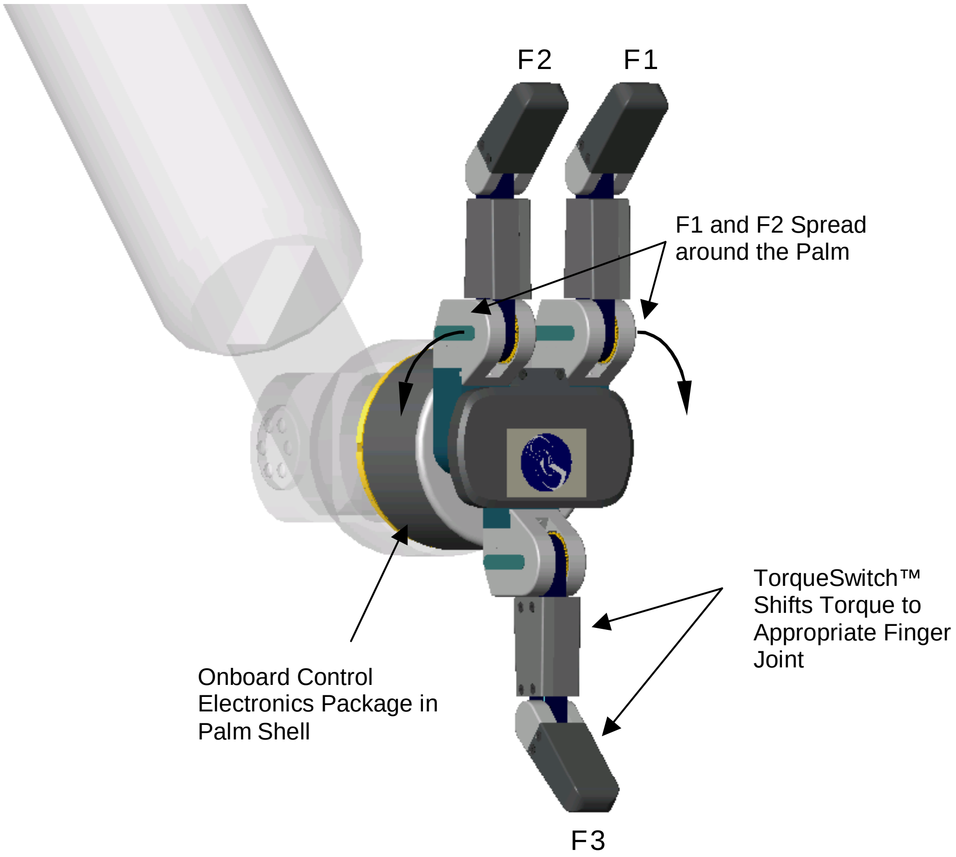
F2
F1
F1 and F2 Spread around the Palm
TorqueSwitch™ Shifts Torque to Appropriate Finger Joint
Onboard Control Electronics Package in Palm Shell
F3

Figure 5 Three articulated fingers of the BarrettHand spread and conform to various shapes

But when the inner link, as illustrated in Figure 9, makes first contact with an object for a secure grasp, the TorqueSwitch, reaches a preset threshold torque, locks that joint against the object with a shallow-pitch worm, and redirects all torque to the fingertip to make a second, enclosing contact against the object within milliseconds of the first contact. The sequence of contacts is so rapid that you cannot visualize the process without the aid of high-speed photography. After the grasper releases the object, it sets the TorqueSwitch threshold torque for each finger in anticipation of the next grasp by opening each finger against its mechanical stop with a controlled torque. The higher the opening torque, the higher the subsequent threshold torque. In this way, the grasper can accommodate a wide range of objects from delicate, to compliant, to heavy.

The finger articulations, not available on conventional grippers, allow each digit to conform uniquely and securely to the shape of the object surface with two independent contact points per finger. The position, velocity, acceleration, and even torque can all be processor controlled over the full range of 17,500 encoder positions. At maximum velocity and acceleration settings, each finger can travel full range in either direction in less than one second. The maximum force that can be actively produced is 2kg, measured at the tip of each finger. Once the grasp is secure, the links automatically lock in place allowing the motor currents to be switched off to conserve power until commanded to readjust or release their grasp.

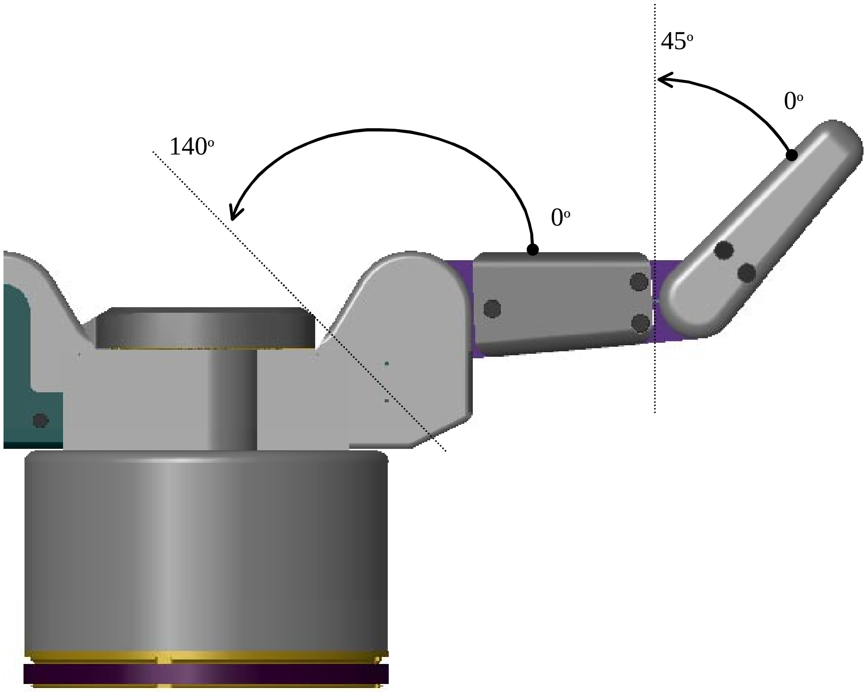

Figure 6 The inner and outer joints close in a 3-to-4 ratio with respect to the robot toolplate until the inner link strikes an obstacle, activating the TorqueSwitch

While the inner and outer finger-link motions curl anthropomorphically, the spread motion of Figure 10 is distinctly non-anthropomorphic. The spread motion is closest in function to a primate's opposable (thumb) finger, but instead of one opposable finger, the BarrettHand has twin, symmetrically opposable fingers centered on parallel joint axes rotating 180 degrees around the entire palm to form a limitless variety of gripper-shapes and fixture functions.

The spread can be controlled to any of [3,000] positions over its full range in either direction within 1/2 second. Unlike the mechanically lockable finger-curl motions, the spread motion is fully backdrivable, allowing its servos to provide active stiffness control in addition to control over position, velocity, acceleration, and torque. By allowing the spread motion to be compliant while the fingers close around an object, the grasper seeks maximum grasp stability as the spread accommodates its position, permitting the fingers to find their lowest energy states in the most concave surface features.

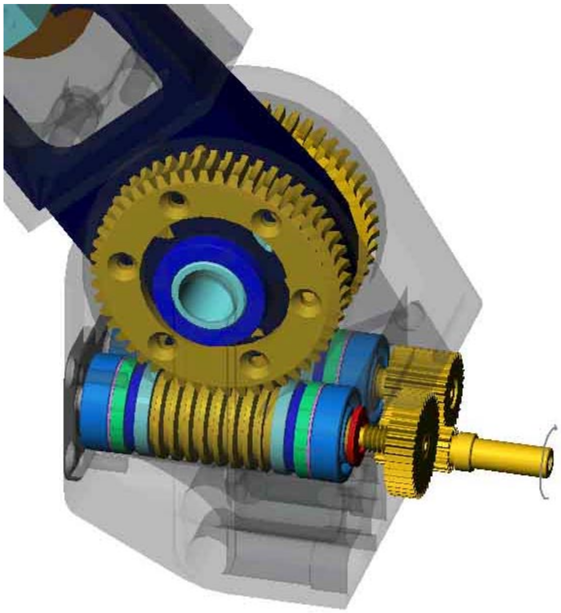

Figure 7 The TorqueSwitch mechanism

Electronic and mechanical optimisation

Intelligent, dexterous control is key to the success of any programmable robot, whether it is an arm, automatically guided vehicle, or dexterous hand. While robotic intelligence is usually associated with processor-driven motor control, many biological systems, including human hands, integrate some degree of specialized reflex control independent of explicit motor-control signals from the brain. In fact, the BarrettHand combines reflexive mechanical intelligence and programmable microprocessor intelligence for a high degree of practical dexterity in real-world applications.

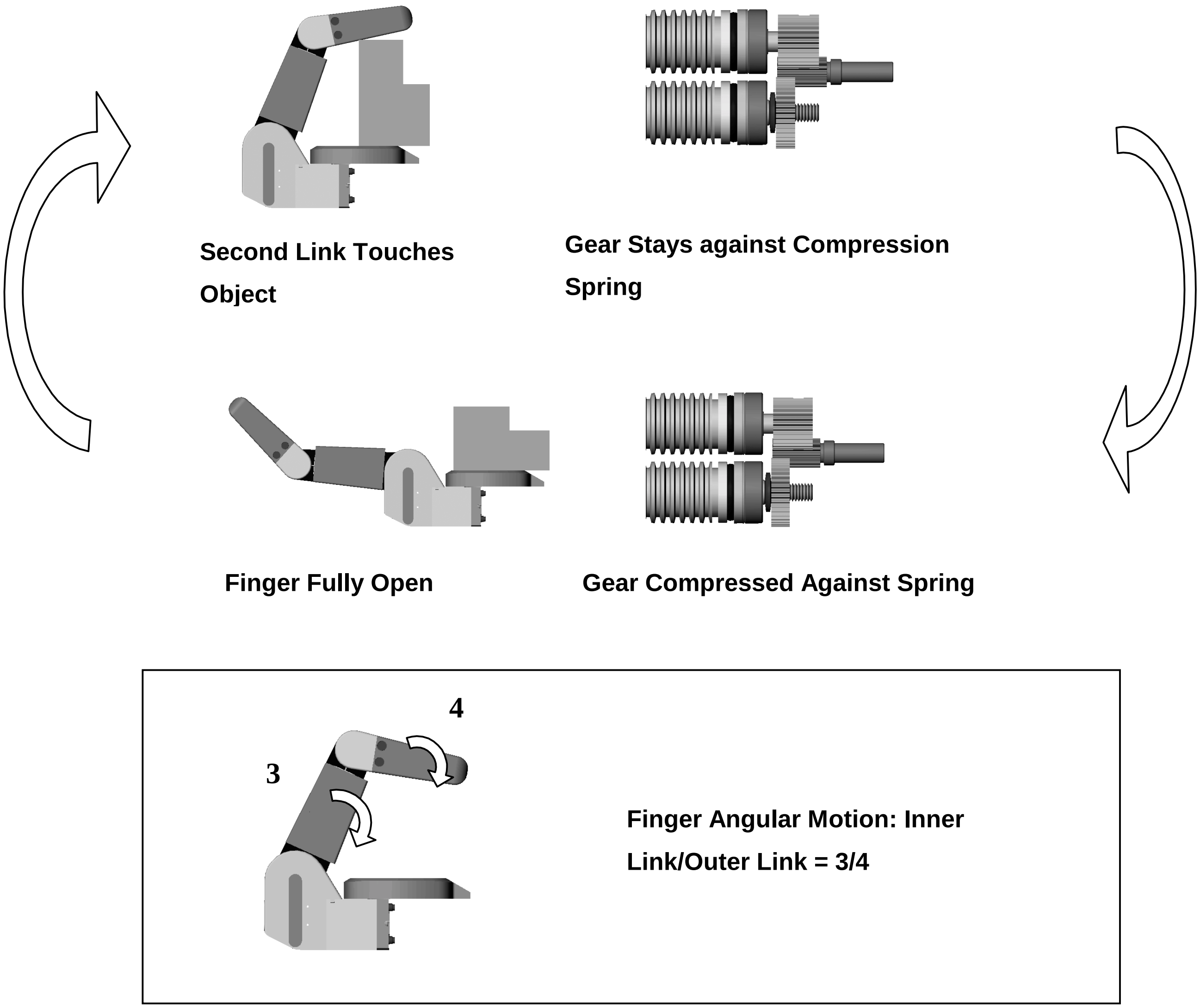
Second Link Touches Object
Gear Stays against Compression Spring
Finger Fully Open
Gear Compressed Against Spring
4
3
Finger Angular Motion: Inner Link/Outer Link = 3/4

Figure 8 TorqueSwitch behavior for fingertip gripping

By strict mathematical definition, dexterity requires independent, intelligent motor control over each and every articulated joint axis. For a robot to be dexterous, at least *n* independent servomotors, and sometimes as many as *n* + 1 or 2*n*, are required to drive *n* joint axes. Unfortunately, servomotors constitute the bulkiest, costliest, and most complex components of any dexterous robotic hand. So, while the strict definition of dexterity may be mathematically elegant, it leads to impractical designs for any real application.

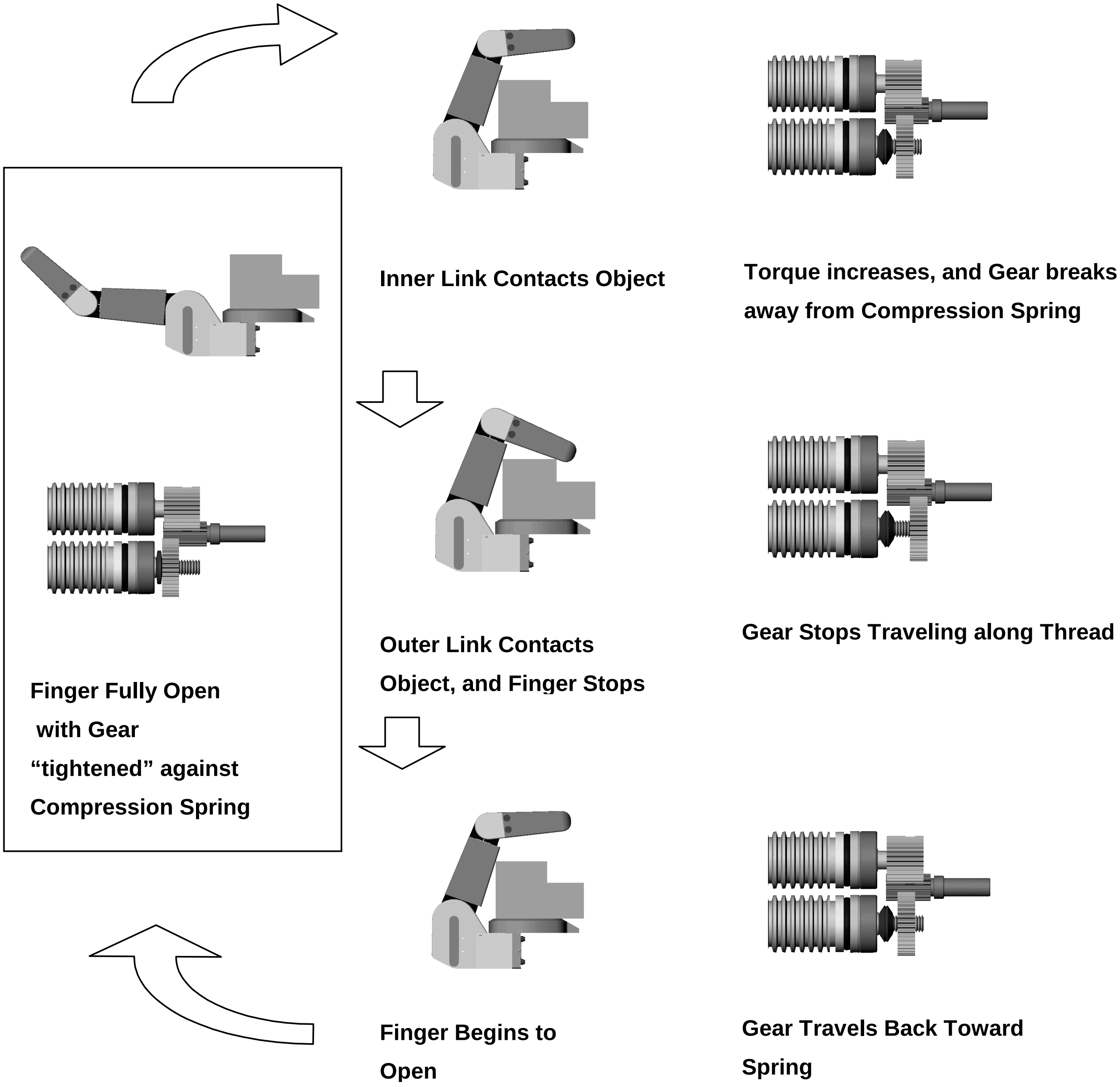
Inner Link Contacts Object
Torque increases, and Gear breaks away from Compression Spring
Finger Fully Open with Gear "tightened" against Compression Spring
Outer Link Contacts Object, and Finger Stops
Gear Stops Traveling along Thread
Finger Begins to Open
Gear Travels Back Toward Spring

Figure 9 TorqueSwitch behavior for secure, finger-enclosed grasping

According to the definition, neither your hand nor the BarrettHand is dexterous. Naturally, their superior versatility challenges the definition itself. If the BarrettHand followed the strict definition for dexterity, it would require between eight and 16 motors, making it far too bulky, complex, and unreliable for any practical application outside the mathematical analysis of hand dexterity. But, by exploiting four intelligent, joint-coupling mechanisms, the almost-dexterous BarrettHand requires only four servomotors.

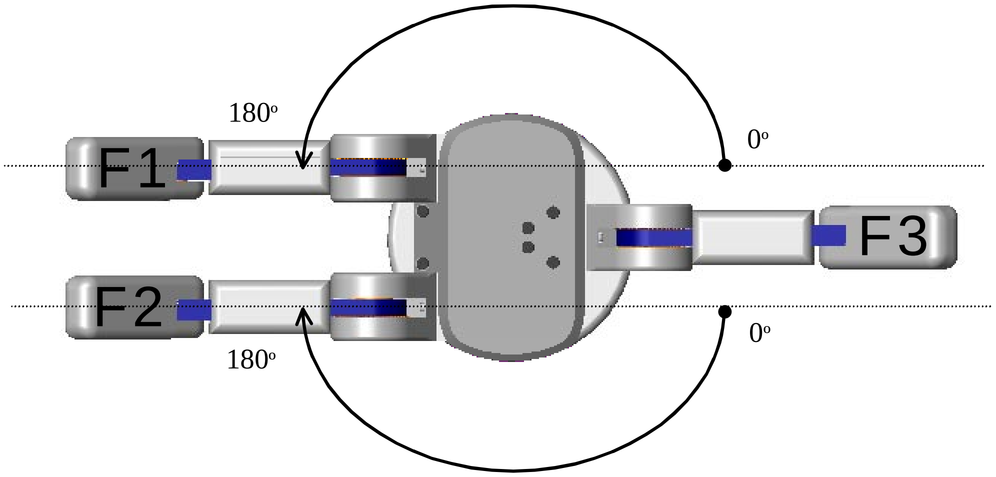

Figure 10 Spread motion of outer two fingers, F1 and F2

In some instances reflex control is even better than deliberate control. Two examples based on your own body illustrate this point. Suppose your hand accidentally touches a dangerously hot surface. It begins retracting itself instantly, relying on local reflex to override any ongoing cognitive commands. Without this reflex behavior, your hand would burn while waiting for the sensations of pain to travel from your hand to your brain via relatively slow nerve fibers and then for your brain, through the same slow nerve fibers, to command your arm, wrist, and finger muscles to retract.

As the second example, try to move the outer joint of your index finger without moving the adjacent joint on the same finger. If you are like most people, you cannot move these joints independently because the design of your hand is optimized for grasping. Your muscles and tendons are as streamlined and lightweight as possible without forfeiting functionality.

The design of the BarrettHand recognizes that intelligent control of functional dexterity requires the integration of microprocessor and mechanical intelligence.

Control electronics

Inside its compact palm, the BarrettHand contains its central supervisory microprocessor that coordinates four dedicated motion-control microprocessors and controls I/O via the RS232 line. The control electronics, partially visible in Figure 4 are built on a parallel 70-pin backplane bus. Associated with each motion-control microprocessor are the related sensor electronics, motor commutation electronics, and motor-power current-amplifier electronics for that finger or spread action.

The supervisory microprocessor directs I/O communication via a high-speed, industry-standard RS232 serial communications link to the workcell PC or controller. RS232 allows compatibility with any robot controller while limiting umbilical cable diameter for all power and communications to only 8mm. The openly published grasper communications language (GSL) <http://www.barretttechnology.com/robot/products/hand/D0AA7A.pdf> optimizes communications speed, exploiting the difference between bandwidth and time-of-flight latency for the special case of graspers. It is important to recognize that graspers generally remain inactive during most of the workcell cycle, while the arm is performing its gross motions, and are only active for short bursts at the ends of an arm's trajectories.

While the robotic arm requires high control bandwidth during the entire cycle, the grasper has plenty of time to receive a large amount of setup information as it approaches its target. Then, with precision timing, the workcell controller releases a "trigger" command, such as the ASCII character "C" for close, that begins grasp execution within a couple milliseconds.

Grasper control language (GCL)

The grasper can communicate and accept commands from any robot-workcell controller, PC, Mac, UNIX box, or even a Palmpilot via standard ASCII RS232-C serial communication — the common denominator of communications protocols. Though robust, RS232 has a reputation for slow bandwidth compared to USB or FireWire standards, but its simplicity leads to small latencies for short bursts of data. By streamlining the GCL, we have achieved time of flight to execute and acknowledge a command (from the workcell controller to the grasper and then back again to the workcell controller) of the order of milliseconds. The initial effort to develop a highly optimized grasper language based on such a standard protocol means that the GCL is upwardly compliant with any future industry-standard protocol.

The grasper has two control modes: supervisory and realtime. Supervisory is the normal mode used to control the grasper. It is made up of a simple command structure, designed for optimal performance and minimized learning curve.

Supervisory mode has the following grammatical structure:

Object (prefix) — Verb (command) — Subject (parameters) — Qualifiers (values)

The prefix refers to motors 1 through 4 with the ASCII values for 1, 2, 3, and 4 corresponding to the fingers Fl, F2, F3, and the spread motion. Any number of prefixes may be used in any order. If the prefix is omitted, then the grasper applies the command to all available axes.

As an example, the ASCII character “C” represents the command which drives the associated motor(s) at its individual default (or user defined) velocity and acceleration profile(s) until the motor(s) stops for the default (or user defined) number of milliseconds. As each motor reaches this state its position is locked mechanically in place.

- 1C closes finger Fl.
- 2C closes finger F2.
- 12C closes fingers Fl and F2.
- C is equivalent to 1234C and closes all three fingers and the spread motion.

We also have defined “S” (derived from “spread”) as a shortcut for “4” and “G” (from “grasp”) as a short cut for “123”, so that:

- GC is equivalent to 123C
- SC is equivalent to 4C

There are similar commands for opening fingers, moving any combination of the four axes to an array of positions, incremental opening and closing by default or user-defined distances, reading and setting user-defined parameter values, and reading the (optional) strain gages on the three fingers. The latest version of the BH8-250 firmware has 21 commands and 28 parameter settings, giving it almost unlimited flexibility.

The realtime mode is reserved for advanced uses such as realtime teleoperation control and is frequently accessed through Barrett's user-friendly GUI for PCs running Windows95/98/NT. In realtime mode, the user specifies a tailored packet-structure in supervisory mode. Barrett's PC software gives the user a histogram of 20 successive time-of-flight tests so that the user can refine the packet structure by quantitatively balancing information content with latency.

The GUI accelerates the prototyping of tasks and includes a pictorial of the grasper with sliders for position and rate control. The GUI also has a novel "Generate C++ Code" button which enables anyone to save and later recall successful algorithms without any knowledge of C or C++ programming. But, with C++ programming familiarity, you can also edit the code as desired.

Once realtime mode is initiated, packets are exchanged in full duplex until an ASCII control character is issued to break out of realtime mode and return to supervisory mode. The system has proven effective and robust in a variety of customer applications.

Conclusion

Although the BarrettHand BH8-250 was only introduced commercially in 1999, 30 units have been put into service around the globe at a price of US$30,000 each. The largest concentration of graspers is among automotive manufacturers and suppliers in Japan, including Honda, Yamaha Motorcycles, and NGK (ceramic substrates for catalytic converters). At this time, these manufacturers are only beginning to explore the capabilities of this versatile device, while some customers, such as Fanuc Robotics and the US and Japanese space programs have become repeat customers.