Proceedings of the 1992 IEEE
International Conference on Robotics and Automation
Nice, France - May 1992

Force Sensing and Control for a Surgical Robot

Peter Kazanzides (1), Joel Zuhars (1), Brent Mittelstadt (1), and Russell H. Taylor (2)

(1) Integrated Surgical Systems, 829 West Stadium Lane, Sacramento, CA 95834 USA
(2) IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598

Abstract

This paper describes the use of force feedback in a surgical robot system (ROBODOC). The application initially being addressed is total hip replacement (THR) surgery, where the robot must prepare a cavity in the femur for an artificial implant. In this system, force feedback is used to provide safety, tactile search capabilities, and an improved man/machine interface. Output of the force sensor is monitored by a Safety Processor, which initiates corrective action if any of several application-defined thresholds are exceeded. The robot is able to locate objects using "guarded moves" and force control ("ball-in-cone" strategy). In addition, the force control algorithm provides an intuitive man/machine interface which allows the surgeon to guide the robot by leading its tool to the desired location. Finally, an application of force control currently under development is described, where the force feedback is used to modify the cutter feedrate (force controlled velocity).

1.0 Introduction

1.1 Surgical application

The application initially being addressed is total hip replacement surgery, where the robot must prepare a cavity in a patient's femur for an artificial implant. The steps in the procedure relevant to the use of robotics are as follows (for a more detailed description, see [4]):

1) Three titanium locator pins are implanted in the patient's femur to form a frame of reference, and a Computed Tomography (CT) scan is taken.

2) The CT data is read by the ORTHODOC pre-surgical planning system [2], and a 3-D reconstruction of the bone is created. Using the computer, the surgeon manipulates implant models in order to determine optimum selection and placement. The ORTHODOC system outputs a data file that specifies, among other things, the implant selection, the position of the implant in CT coordinates, and the position of the three locator pins in CT coordinates.

3) The ROBODOC system accepts the data file from ORTHODOC. The robot is draped (for sterility) and the patient's femur is attached to the robot base via a femoral fixator. A ball probe is installed, the surgeon guides the robot to the approximate location of each locator pin, and the robot performs a tactile search to accurately determine the position of each pin in robot coordinates. The CT-robot transformation is computed from the pin positions in CT coordinates and robot coordinates, and used to transform the surgeon's desired implant position into robot coordinates.

4) A cutting tool is installed and the robot cuts the implant cavity at the desired position in the femur.

The first generation system has performed 26 surgeries, all successful, on dogs requiring artificial hip implants. A system for human use is currently being developed.

1.2 System architecture

The surgical robot [4][5][6] is a 5-axis SCARA robot consisting of two revolute axes (theta-1, theta-2) in the X-Y plane, a prismatic axis (Z), and two wrist rotations (roll, pitch). A six axis force sensor (3 forces and 3 moments) is attached to the end of the wrist, and a surgical tool is attached below the force sensor. The robot controller is based on an IBM Industrial PC with 3 Axis Control Cards (ACC's) for servo control (up to 2 axes per ACC). A Safety Processor is added to the PC bus to independently monitor a redundant encoder on each robot axis (the redundant encoders are on the output shafts, as opposed to the primary encoders which are on the motor shafts). The Safety Processor also interfaces with the force sensor via an A/D subsystem that converts the force sensor analog outputs to digital values. The robot controller obtains the digital force data from the Safety Processor dual-port memory.

1.3 Force control algorithm

The force control described in this paper is based on damping control [7], where the reference velocity is derived from the force error (see Figure 1). The reference

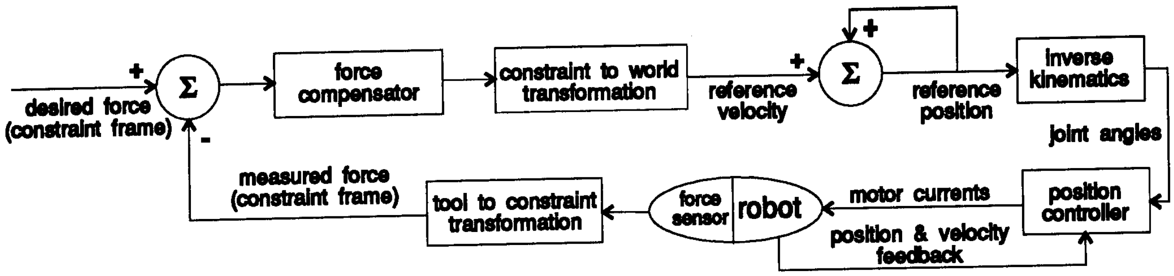

Figure 1: Force controller

velocity is added to the current position (i.e., discrete integration of the velocity) to provide a reference position to an inner position servo. The coordinate transforms in Figure 1 allow the desired force and the force compensator parameters to be specified in a frame natural to the task (i.e., the constraint frame). In the current implementation, either the robot world frame or tool frame can be selected as the constraint frame. Generalization to arbitrarily-specified constraint frames is straightforward, but has not been necessary in our application.

The position servo provided with the robot (on the ACC 's) was used for a number of practical reasons -- it works well, it includes many safety features, and we do not have the capability to modify it anyway. In addition, several researchers have shown that a high-frequency inner position loop improves force control stability [1][3].

We are, however, able to add new "verbs" to the real-time loop (on the Industrial PC) that generates setpoints for the position servos every 18.2 msec (55 Hz). Thus, the force controller was implemented as a new verb, called *fcomply*, in the real-time loop.

In the simplest form of damping control, the *force compensator* in Figure 1 is just a diagonal matrix of gains. Some obvious practical enhancements are to include a deadband about the origin (to compensate for noise in the force feedback), and to saturate the velocity at an upper

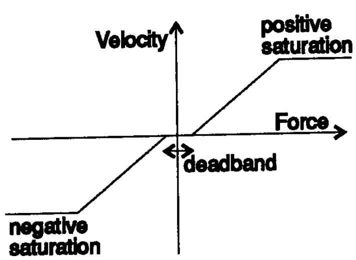

Figure 2: Compensator with deadband and saturation

limit (to reduce impact forces). With these modifications, the force compensator can be represented by the force-velocity profile shown in Figure 2. The initial force control implementations, which worked rather well for object location and man/machine interface functions, utilized essentially this type of compensator. Some improvements to the compensation, such as the introduction of nonlinear gains (besides the deadband and saturation nonlinearities) are presented in a later section.

2.0 Safety monitoring

The safety processor monitors the force feedback and compares it against two application-specified thresholds, called the *force pause threshold* and the *force stop threshold*. If the force pause threshold is exceeded, the safety processor informs the robot controller, via a software flag, that all robot motion should be stopped, the cutter should be turned off, and the system should enter an error recovery mode. The recovery procedure depends on the specific situation, but can generally be classified as one of the following actions:

1) The error condition is corrected and the robot continues. An example of this is when the robot hits an obstacle, such as a soft tissue retractor, while cutting the bone. Once the obstacle has been removed, the cutting procedure can be resumed. Another example is when the robot is cutting extremely hard cortical bone, which occasionally causes the threshold to be exceeded. Usually, the robot motion can simply be resumed.

2) The robot is moved and the error condition is corrected; the robot must then return to the point at which motion was interrupted before resuming. This situation can occur for either of the examples cited above if the surgeon deems it necessary to withdraw the robot from the bone for further inspection. For example, if the cutter was damaged by contact with a metallic object, such as a soft tissue retractor, it is necessary to change the cutter before resuming.

3) The robot is moved and the error condition is corrected; the robot then continues to the destination from

its current position. A typical example of this is when the robot encounters an obstacle while moving outside the bone (i.e., not during the cutting procedure). If the obstacle cannot easily be moved, the robot can be guided around it, or the robot's orientation can be changed so that the obstacle is avoided. The latter option is possible when the motion involves only a change in the tool tip position, so that orientation of the tool does not need to be specified. For example, there is a large range of possible tool orientations when using the ball probe to find the locator pins. Thus, if an obstacle, such as the femoral fixator, is encountered during the pin finding procedure, the problem can be corrected by re-orienting the tool.

In the initial system, force pause threshold checking was disabled during the user recovery procedure because otherwise it would not have been possible to move the robot (if it still exerted a force greater than the threshold). In the current system, the threshold checking will be temporarily disabled while a small automatic motion is performed to reduce the force below the limit. The threshold checking will then be re-instated before initiation of the user recovery procedure.

The force stop threshold is defined as a safeguard against failures in the recovery procedure for the force pause threshold (for example, if the motion to reduce the force actually causes an increase in the applied force). The force stop threshold is higher than the force pause threshold, and if it is exceeded at any time, the Safety Processor directly powers off the robot (i.e., trips the motor power relay). The Safety Processor continues to monitor the force and only allows the robot to be powered on when the force falls below the stop threshold (the computer can never turn on robot power -- the power on button must be manually activated).

3.0 Tactile search methods

There are several situations in which the robot must accurately locate an object in its environment. For example, the robot must find the three locator pins that were inserted into the patient's femur before the CT scan was taken. The pre-surgical planning software locates these pins in CT coordinates, so by finding the pins in robot world coordinates, the robot system can determine the transformation from CT coordinates to robot world coordinates. The accuracy with which the robot locates these pins directly affects the accuracy of the CT-robot transformation matrix.

Another application of tactile search is for calibration of the robot end-effector, which is necessary for the robot to accurately position and orient the tool tip (whether it be a cutter or ball probe). The calibration procedure currently being used requires the robot to locate the center of a fixed post from a number of orientations. The data is then fit to a kinematic model of the robot wrist and tool.

Although there are many different sensing modalities (optical proximity sensing, electrical continuity, etc.) that can be used to satisfy some of the above requirements, it was felt that force feedback provided a good solution in all cases. This is especially true in a surgical setting, where the presence of blood or other fluids may defeat the alternative methods.

3.1 Finding the point of contact

The most basic tactile search strategy developed was one to locate the "point of contact" with a surface in a given direction. Here, "point of contact" is defined as the transition between zero and non-zero measured force. This point is determined as follows:

1) The robot performs a "guarded move" [8] in the specified direction. In a "guarded move", the robot moves along the desired path until a specified force threshold is encountered. This is similar to the force pause threshold described above (Safety Monitoring), except that the threshold is lower and exceeding it is not considered an error condition. The speed of the "guarded move" and the force threshold are selected to yield a small applied force, F0, when the motion is terminated.

2) The robot moves a small displacement away from the surface and then reads another force value, F1. Using F0 and F1, the system adjusts the step-size (displacement) so that a "small number" of force readings can be expected before surface contact is lost. This "small number" (currently 5) is selected to balance the time required for data collection and computation with the accuracy of the result. The adaptive adjustment of the step size compensates for the fact that end-effector stiffness varies significantly with the search direction and the orientation of the tool.

3) A linear regression is performed on the position and force values collected above, and the zero-crossing of the best fit line is used as the "point of contact".

3.2 Finding the top center point of a cylinder

The method for finding the point of contact can be used to find the top center point of a cylinder. Specifically, the robot must locate three points on the top of the cylinder, which define a plane, and three points on the edge of the cylinder which, when projected to the plane, define a circle. Computing the center of this circle yields the position of the top center point of the cylinder.

Initially, this technique was used to find the locator pins in the femur. Although it worked very well in the laboratory, the exposure required to perform the method (especially finding the three edge points for the distal pins) was difficult to achieve in the operating room, so an alternate strategy ("ball-in-cone") was adopted.

3.3 Ball-in-cone strategy

The "ball-in-cone" strategy was developed as a practical method for finding the locator pins in a surgical setting. The distal pins (located near the knee) are surrounded by soft tissue, and since the surgeon desires to make as small an incision as possible, often the only part visible is the hex socket used for pin insertion.

The basic idea behind the "ball-in-cone" strategy is that a ball dropped into a cone will settle in a predictable, and repeatable, location. In our application, the "ball" is the robot's ball probe and the "cone" is the hex socket of the pin (actually, the hex socket only provides an approximation of a cross-section of a cone, but that is all that is really required). The diameter of the ball probe (approx 3 mm) is a little larger than the diameter of the hex socket (approx 2.5 mm), so the robot can find the pin by seating the ball probe in the hex socket. This is similar to the classical robotics problem of "peg-in-hole", although it is somewhat easier because orientation of the ball probe is not important.

Following is a summary of the "ball-in-cone" strategy:

1) The surgeon guides the robot to two points to teach an approximate pin center and normal (to the head of the pin). The approximate pin center is taught by guiding the robot's ball probe into the center of the hex socket (Figure 3b), and the approximate pin normal is computed from the approximate center point and a point "above" the hex socket (a fairly accurate point can be obtained by inserting a pin extender into the hex socket and teaching a point at the end of the extender, as shown in Figure 3a).

2) The robot enters a force control mode, with a non-zero desired force in the direction of the (approximate) pin normal and a zero desired force in the other two Cartesian directions (see Figure 3c). The non-zero desired force will cause the robot to move the ball probe in the direction of the pin normal until the desired force is

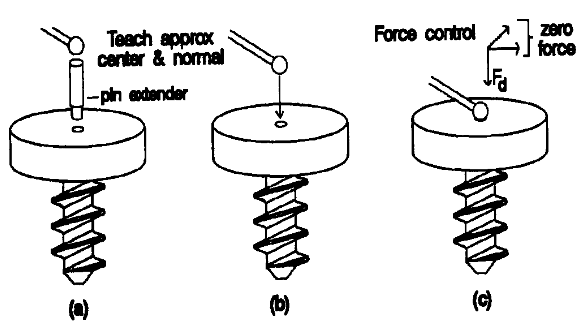

Figure 3: Ball-in-Cone Strategy for Finding Pin

attained. At the same time, if any lateral forces occur, the robot will move to zero them out. This method will work as long as the initial data (approximate center and normal) is reasonable. For example, the center of the ball probe must fall within the hex socket; otherwise, it may settle on any flat spot on the pin surface.

3) The system prompts the surgeon to verify that the probe ball is centered in the socket. This is necessary because of the dependence on the initial data, as discussed above. If the surgeon verifies that the ball is centered, the robot backs away from the pin collecting force and displacement data, so that the "point of contact" can be determined.

4.0 Man/machine interface

The force controller is also used to provide the surgeon with a very intuitive method for guiding the robot. If the robot is put in force control mode, with a desired force of 0, any applied force will cause the robot to move in the direction of the applied force (in an attempt to zero out the force). Thus, the surgeon can guide the robot by grabbing the tool and leading it to the desired location. This is easier than the traditional method of using a teach pendant, which would require knowledge of the robot coordinate system (e.g., which direction is +X, -X, etc).

4.1 Linear gains

Since the underlying force control algorithm is a damping controller, the guiding velocity is proportional to the applied force (i.e., velocity = control_gain * force). The value of the control gain is based on two factors: the desired resolution of motion, and the stability of the force controller. Fortunately, it is not necessary to trade-off between these two factors. When the robot is in free-space, a high gain can be selected since the surgeon will generally want coarse guiding of the robot. Stability in this situation is relatively good since it is dependent on the stiffness of environment, and in the case of guiding by the surgeon, the stiffness is low. When the surgeon approaches an object whose position needs to be "taught" to the robot (such as a locator pin in the femur), a lower gain should be used because the surgeon will want finer motion control. Also, the stiffness of the pin/bone is much higher than the surgeon's hand, so the lower gain is necessary to preserve stability if contact is made. For these reasons, the first implementation of force-controlled guiding allowed the surgeon to select different gain constants from the teach pendant.

4.2 Nonlinear gains

The use of nonlinear gains significantly improves the performance of the force guiding feature by eliminating the need to manually change the control gains based on whether a large or small scale movement of the robot is required. The following exponential gain was found experimentally to be an acceptable choice:

$$\text{Gain} = K * E^{\left(1 - \frac{\text{MaxForce}}{\text{ForceMag(t)}}\right)}$$

ForceMag(t) is the magnitude of the current sensed force, and is not allowed to exceed MaxForce or to be less than a token minimal force (to avoid division by 0). It was determined that a reasonable MaxForce to use was about 8 pounds. This results in a Gain which varies from approximately 0 to K as ForceMag(t) varies from its minimum to maximum values (K is the linear gain of Section 4.1). A deadband of ± 0.05 lbs was selected to reduce the effects of force measurement noise. Using this method, the surgeon can both move the robot comfortably fast during large scale movements and comfortably slow during small scale movements, without changing any control parameters, by pulling heavily or lightly on the tool.

4.3 Position and orientation guiding

In the initial implementation, force-controlled guiding was separated into a position mode and an orientation mode. This was done primarily because the pitch axis (an addition to the standard 4-axis SCARA robot) was a stepper motor, and the interface did not allow force control at a low enough level.

The current pitch axis is a servo motor integrated with the rest of the system, so combined position and orientation guiding has been implemented and is available. It is still often desirable, however, to partition guiding into position and orientation modes, so this capability has been preserved. More generally, the individual force compensator gains can be set to select any particular subset of axes (of the constraint frame) for force control. For example, by choosing the tool frame as the constraint frame and setting all gains but the z-axis gain to zero, robot guiding can be restricting to the tool approach vector. This "tool axis guide" mode is especially useful for guiding the robot in and out of the bone cavity during error recovery procedures.

5.0 Force controlled velocity

Force controlled velocity milling is a very desirable feature of a robotic surgical system. Without force controlled velocity, the milling feed rate must be determined empirically by finding the maximum force that can be applied at the drill tip before problems such as tool chatter occur, and then choosing a speed at which this force will not be exceeded even while cutting through the hardest anticipated material. The problem is that the hardest material is encountered over only a very small portion of the milling volume. Velocity needs to be controlled so that the robot slows down while cutting through material that produces high forces and speeds up when cutting forces are lower.

5.1 Replacing time with a function of force, time

For a robot that uses real-time setpoint generation to implement movement, time is incremented linearly and Cartesian setpoints are calculated by substituting the current time into precomputed position equations for each axis and motion phase (e.g., acceleration, constant velocity, and deceleration). The trajectory planner which computes these position equations ensures that for all applicable time, a multi-axis setpoint results which lies on the desired line of motion (currently only straight lines of motion are considered). Based on this, one could imagine that the velocity and acceleration profiles for the planned move could be altered, without affecting the position trajectory, by replacing time with a function of force and time (e.g., "time warping").

A logical choice for the replacement of time is the function $t - D(F(t))$, where $D(F(t))$ is a delay function, and $F(t)$ is the force seen at the tool tip. The rate of increase of $D(F(t))$ should rise with increasing force, thereby slowing the rate of increase of time in the setpoint generation equations. This slowing of time will cause a slowing of velocity and an accompanying variation in acceleration. It was found that no velocity discontinuities (or acceleration spikes) occur at the motion phase change interfaces through use of this substitution, as long as $D(F(0))$ equals zero (initial condition). Also, the desired behavior will result from the use of any equation for $D(F(t))$ whose time derivative varies from ε to 1 as $F(t)$ varies from 0 to a defined maximum force. Here, ε is a small number between 0 and 1. With these constraints, velocity will approach zero as force approaches the maximum force, but the robot will never move backwards along the trajectory.

5.2 Using a force dependent time delay

Several equations were considered as potential choices for the time derivative of $D(F(t))$, all of which satisfied the above constraints. Based on simulation results, the following exponential equation was judged to be likely to give the best performance of those considered:

$$\frac{d}{dt} D(F(t)) = E^{-R\,(\text{MaxForce} - F(t))} \qquad \{F(t) \leq \text{MaxForce}\}$$

A nonlinear equation is preferable to a linear one for several reasons. First, during milling, it is desirable to have the variations in velocity be less sensitive to low magnitude force variations than high magnitude ones. Low magnitude variations should be treated as noise, and ignored as much as possible. Second, it was found that nonlinear equations allow higher maximum velocities for given acceleration/deceleration constraints than linear ones. The exponential equation is preferable to many other nonlinear choices since it can be analyzed mathematically with relative ease.

A maximum rate of decrease must be artificially imposed on F(t), since no limitation on the drag variation of the medium through which the tool tip passes can be made, allowing the observed drag to go from infinite to zero instantaneously. This would cause the tool tip force to decrease very rapidly, resulting in an attempt to increase velocity uncontrollably fast. This problem can be handled by imposing a linear maximum rate of decrease on F(t), or $\frac{d}{dt} F(t) \geq -Q$. Note that no such problem arises when the observed drag goes from zero to infinite instantaneously, since the observed tool tip force will rise slowly from zero in that case.

Through stability considerations, the maximum velocity and maximum deceleration can be calculated from the maximum allowable force, the stiffness of the tool, and the force update period. The system can use this relationship to limit the requested velocity and deceleration so that stability is preserved. R and Q can be calculated uniquely from all five system parameters.

5.3 Implementation comments

Using this method, the trajectory planner sets up the move as it normally would. The real-time loop uses t - D(F(t)) in place of t, and computes D(F(t)) each period as the sum of the current and all previous values of d/dt D(F(t)) with D(F(0))=0. The real-time loop also keeps track of successive force readings and imposes the artificial limit on change of force of $d/dt\, F(t) \geq -Q$.

Initial experiments have been performed using this system in a variety of situations, including hitting solid objects and cutting bones. These have shown that the system operates as desired and is very stable in all cases.

6.0 Conclusions

Performance of the surgical robot system is greatly enhanced by the use of force sensing and control. The ability to sense forces at the wrist improves system safety because it detects contact of the tool with unexpected objects. Similarly, force feedback is used to implement "guarded move" strategies, which are useful for locating surfaces whose positions are not precisely known.

The addition of force control, even a fairly simple implementation based on damping control, opened up new possibilities in object location and man/machine interfaces. In particular, a "ball-in-cone" strategy is used to quickly and accurately find the locator pins that establish the coordinate frame for the surgery. The force controller also improves the surgeon-robot interface because it allows the surgeon to guide the robot by dragging the tool tip to the desired location.

Finally, a special form of damping control is presented that allows the robot's velocity along a pre-planned trajectory to be modified in response to the sensed force. This is achieved by a "time warping" technique, which is a practical solution for any system in which the trajectory parameters (e.g., acceleration, velocity, blend times, etc.) cannot easily be modified during the motion. This force controller will allow the surgical robot to automatically adapt the cutter feed rate to different bone densities.

Acknowledgements

The ROBODOC surgical robot system is the product of contributions by many individuals and organizations. In particular, we would like to acknowledge Dr. Hap Paul, Dr. William Bargar, Bill Williamson, Phil Cain, Bela Musits, and the rest of our colleagues at Integrated Surgical Systems, Bill Hanson of IBM Palo Alto Scientific Center, and T.C. Hsia of the University of California at Davis.

References

[1] DeSchutter, J., "Compliant Robot Motion: Task Formulation and Control", *Ph.D. Thesis*, Katholieke Universiteit, Leuven, Belgium, Feb 1986.

[2] Hanson, W. A., Taylor, R.H., Paul, H.A., Williamson, W., *ORTHODOC - An Image Driven Orthopaedic Surgical Planning System*, Proc. 12th IEEE Medicine & Biology Conf., Philadelphia, PA, 1990.

[3] Maples, J.A., and Becker, J.J., "Experiments in Force Control of Robotic Manipulators", *Proc. 1986 IEEE Conf. on Robotics & Automation*, Vol 2, pp 695-702, San Francisco, CA, April 1986.

[4] Paul, H.A., Mittelstadt, B., Bargar, W.L., et al., "A Surgical Robot for Total Hip Replacement", accepted for *IEEE Conf. on Robotics & Automation*, Nice, May 1992.

[5] Taylor, R.H., Paul, H.A., Mittelstadt, B.D. et al, "An Image-directed Robotic System for Hip Replacement Surgery", *Journal of Robotics Soc. of Japan*, Vol 8, No. 5, Oct, 1990.

[6] Taylor, R.H., Paul, H.A., et al, "Taming the Bull: Safety in a Precise Surgical Robot", *Proc 5th ICAR*, Pisa, June 1991.

[7] Whitney, D.E., "Force Feedback Control of Manipulator Fine Motions", *Transactions of the ASME, Journal of Dynamic Systems, Meas., and Control*, pp 91-97, June 1977.

[8] Will, P., and Grossman, D., "An Experimental System for Computer Controlled Mechanical Assembly", *IEEE Transactions on Computers*, Vol C-24, pp 879-888, 1975.