Proceedings of DETC2001
ASME Design Automation Conference
September 9-12, 2001

DAC-21128

PATH VERIFICATION FOR UNSTRUCTURED ENVIRONMENTS AND MEDICAL APPLICATIONS

Leo J. Stocco
Integrated Surgical Systems
1850 Research Park Drive
Davis, CA, 95616
(530)792-2610, (530)792-2690, LStocco@robodoc.com

ABSTRACT

In some robot applications, workspace limitations can interfere with the desired path. In unstructured environments such as robot assisted surgery, checking for this must be done quickly to minimize procedural delays. Here, an algorithm is proposed that computes the distance between a few points on the desired path and the nearest workspace boundary to determine if the path is admissible (i.e. inside the dextrous workspace). Execution time is dependent on the proximity of the path to the reachable workspace limit and is upper bounded by a configurable workspace deadband value. The algorithm is applied to the ROBODOC® Surgical Assistant with good results.

Keywords: path, workspace, admissible, ROBODOC

NOMENCLATURE

${}^0P$	a motion path represented in frame 0
${}^0T_1$	a transformation from frame 1 into frame 0
L	total path length
W	reachable workspace
x	a position in the workspace
d_0	distance from position 0 to the workspace boundary
$\dot{q}$	deadband between reachable and dextrous workspaces
a_0	length of link 0
q_0	actuator 0

INTRODUCTION

In some robot applications, the robot must not start its task unless it can be completed without interruption. This is the case with automotive painting as it is with certain types of welding [1]. For these applications, the admissibility of the path must be verified before proceeding. Most proposals to address the admissible path problem either find an admissible path between two points [2-4] or modify a desired path to bring it within workspace constraints. Path modification has been proposed for both task-space paths [5-7] and joint-space paths [1] when the manipulator is redundant. This paper addresses the admissible path problem as it applies to the ROBODOC Surgical Assistant. The ROBODOC is a non-redundant milling robot so the desired path can not be modified without altering the result. To ensure the accuracy and continuity of an implant cavity prepared by the ROBODOC, the cut path must be navigable from start to finish without encountering any workspace limits. The path should also be free of obstacles to the entire robot arm. Although this could be checked by an algorithm such as that proposed in [8], it is not possible to identify all obstacles in the proximity of the robot .

An algorithm is proposed here that determines the admissibility of a path by computing the distance between a number of incremental locations along the path and the nearest workspace boundary. The algorithm is efficient in practice and guarantees that any path that is found to be admissible is continuously admissible. Section 2 of this paper motivates the need for a fast path verification algorithm in robot assisted arthroplasty procedures, Section 3 describes the proposed algorithm, Section 4 quantifies its performance and Section 5 describes its implementation in the ROBODOC System with conclusions presented in Section 6.

ROBOT ASSISTED ARTHROPLASTY

Robot assisted arthroplasty procedures include total hip arthroplasty (THA) and total knee arthroplasty (TKA). In THA, a round cup is implanted into the acetabulum (pelvis) and an elongated hip stem prosthesis is implanted into the femur (thigh bone). Using a robot to mill the hip stem cavity rather than cutting it manually with a reamer (hand drill) and broach (rasp) reduces bone trauma, avoids splintering and guarantees that geometric and placement accuracy will be within a set tolerance. Close correspondence between the geometry of the implant and cavity ensures uniform bone stress and enables the use of cementless implants which rely on bone ingrowth for fixation.

In TKA, the distal femur and proximal tibia (calf bone) are resurfaced and replaced by prostheses. During the preplanning stage, the orientations of the prostheses are chosen precisely since proper joint alignment is critical to the longevity of the new joint. In conventional TKA procedures, a collection of jigs (saw guides) are used to manually prepare the bone surfaces in accordance with the preoperative plan. When a robot prepares the bone surfaces, the correspondence between the preoperative plan and the surgical outcome is guaranteed to be within a fixed tolerance. The same can be said for the geometric accuracy of the implant cavity as well. Typical cementless hip stem and knee implants are shown in Figure 1. Note the porous surfaces to accommodate bone ingrowth.

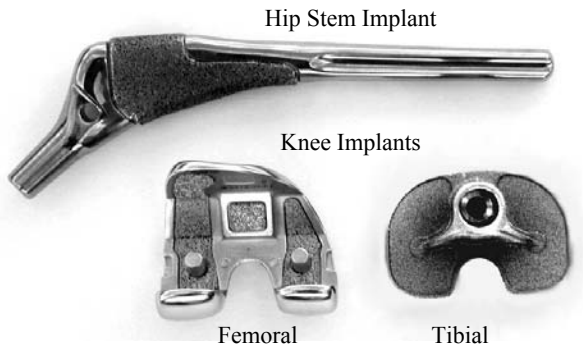

Figure 1. Typical Hip and Knee Implants

The ROBODOC System is comprised of a preoperative planning station (ORTHODOC®) and a robotic cutting mill [9]. The surgeon supplies a CT scan of the patient's leg to the ORTHODOC System and then uses it to select an implant and position it in six dimensions with respect to the CT data. The ORTHODOC generates a surgical plan which is transferred to the ROBODOC cutting robot.

In surgery, the femur and/or tibia (TKA) are exposed, fixated to the robot and known landmarks on the CT data are identified on the bone. Positions on the bone and CT are fitted to one another to derive a transformation between the two. After this registration of the bone, robot and CT data, the robot proceeds to cut the implant cavity.

Implant placement accuracy is specified to be within $\pm 1mm$ and dimensional accuracy (i.e. deviation from the desired implant shape) is specified to be within $\pm 0.2mm$ for each cross-sectional boundary [10]. Since the dimensional specification is more stringent than the placement specification, the bone can not be moved once cutting starts or the dimensional specification will not be satisfied. Therefore, the admissibility of the path must be verified prior to the start of cutting to ensure that workspace limitations will not prevent the cut path from being completed. Although the ROBODOC has a large motion range, it often requires a long cutter sleeve to mill a stem implant cavity such as that shown in Figure 1. Re-orienting a long cutter sleeve about the cutter tip requires large arm motions which can easily violate workspace boundaries.

The desired cutting trajectory ${}^{I}P$ supplied by ORTHODOC is specified in implant coordinates, a generic reference frame used to represent all implant cut paths. A transformation from implant coordinates to CT coordinates ${}^{CT}T_I$ is also supplied by ORTHODOC and accompanies ${}^{I}P$ on the surgical plan. Once the bone position is identified in surgery, a transformation from CT coordinates to robot coordinates ${}^{R}T_{CT}$ is computed by ROBODOC and used with ${}^{CT}T_I$ by (2) to represent the cutting trajectory ${}^{I}P$ in robot coordinates ${}^{R}P$.

$$ {}^{R}P = {}^{R}T_{CT}\,{}^{CT}T_I\,{}^{I}P \tag{2} $$

Since ${}^{R}T_{CT}$ is not available until after the position of the bone has been identified, the admissibility of ${}^{R}P$ can only be verified while the patient is under anesthetic and, in the case of knee arthroplasty, may have a tourniquet restricting blood flow to the leg. Since prolonged exposure to anesthesia and/or tourniquets poses a health risk, it is necessary to minimize this time.

PATH ADMISSIBILITY ALGORITHM

In practice, it may not be sufficient for a path to be reachable since proximity to singular positions may reduce accuracy and controllability below acceptable levels. Therefore, the admissibility criteria proposed here includes a deadband volume inside the workspace boundary which, although reachable, is considered inadmissible. This ensures that any admitted trajectory is a minimum distance $\underline{d}$ from the reachable workspace boundary and is, therefore, inside the dextrous workspace of the manipulator.

The proposed path verification algorithm is shown in (3) through (11) where i is the looping index, L is the total path length, x_i is a position belonging to path P, $X(l_i,P)$ is a function that returns the position x_i corresponding to a length l_i along path P, $D(x_i,W)$ is a function that returns the distance d_i between position x_i and the nearest workspace boundary, $\underline{d}$ is the deadband width (i.e. the minimum distance between the path

and the reachable workspace boundary) and W is the set of all positions inside the reachable workspace of the manipulator. Note that any singular position that exists inside the reachable workspace should be treated the same as a reachable workspace boundary by $D(x_i,W)$.

$$i = 0, l_i = 0 \quad (3)$$

Repeat (4)

$$x_i = X(l_i, P) \quad (5)$$

$$d_i = D(x_i, W) \quad (6)$$

$$i = i + 1 \quad (7)$$

$$l_i = min(l_{i-1} + d_{i-1}, L) \quad (8)$$

$$\text{Until } (d_{i-1} < 2\underline{d}) \text{ or } (l_{i-1} = L) \quad (9)$$

$$\text{If } (d_{i-1} < 2\underline{d}) \quad P \notin W \quad (10)$$

$$\text{Else} \quad P \in W \quad (11)$$

Consider the example elbow manipulator in Figure 2 with link lengths a_0 and a_1 where $a_0 > a_1$ and with joints q_0 and q_1 which have unlimited motion range. The manipulator, its workspace and an example polyline path P are shown in Figure 2. The reachable workspace is shown by a solid line and the dextrous workspace is shown by a dotted line.

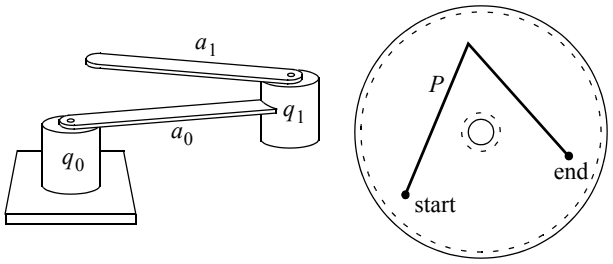

Figure 2. Elbow Manipulator and Workspace

The admissibility of path P is verified as follows. The distance between the start of P and the workspace edge is computed and the path is followed by that distance. This new position is guaranteed to be inside the workspace regardless of the shape or direction of the path. The distance between the new position to the workspace edge is computed and the path is followed repeatedly until either the end of the path is reached or a position is found to lie outside the dextrous workspace. If the end of path is reached, the entire path is guaranteed to be inside the dextrous workspace.

A more detailed example is shown in Figure 3. The algorithm starts with a looping index and path distance of zero (3). The distance from the workspace edge d_0 (6) and a new path distance ($l_1 = d_0$) (8) are computed. Since $d_0 > 2\underline{d}$ and $l_0 < L$ (9), the process is repeated (i.e. x_1 (5), d_1 (6) and $l_2 = d_0 + d_1$ (8) are computed). This repeats until $i = 9$ at which point $l_7 + d_7 > L$ so $l_8 = L$ (9) (see also (12) and (13)) and the algorithm returns success.

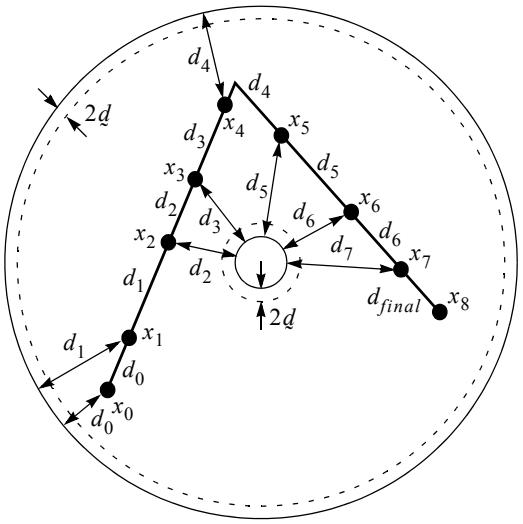

Figure 3. Path Verification Example

$$l_n = \sum_{m=0}^{n-1} d_m, n \in \{0 \ldots 7\} \quad (12)$$

$$L = \left(\sum_{n=0}^{6} d_n \right) + d_{final} \quad (13)$$

Note that the deadband value $\underline{d}$ bounds the worst-case performance of the algorithm since it is half the minimum increment to l_i (8). In other words, the algorithm progresses a minimum distance of $2\underline{d}$ along P with each loop iteration. Note that in the degenerate case ($\underline{d} = 0$), the algorithm can become trapped as P approaches a workspace boundary (i.e. as $d_i \to 0$). Also note that intermediate portions of a verified path may penetrate the $2\underline{d}$ deadband by up to half its width. Therefore, all points on an admitted path are guaranteed to be a minimum distance of $\underline{d}$ from the workspace boundary. Consider the pre-defined path P in Figure 4 which is a worst-case example of how close a path can come to the reachble workspace boundary without violating the admissibility criteria. A path that travels directly toward the workspace boundary at x_{n+1} must reverse direction no closer than $\underline{d}$ to the reachable workspace boundary for the next point x_{n+2} to be admissible.

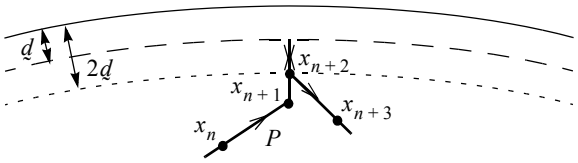

Figure 4. Dextrous Workspace Boundary

In general, the greatest obstacle to implementing the path admissibility algorithm is obtaining a function $D(x_i,W)$ which

computes the distance between a point on the path and the nearest workspace boundary. Methods have been proposed for identifying the workspace boundaries of certain parallel [6] and serial [7] manipulators and an analytic solution is easily obtained for simple kinematic arrangements such as Figure 2. For other, more difficult cases, $D(x_i,W)$ may have to be computed numerically.

The algorithm is easily extended to most 6-DOF manipulators that combines a Cartesian arm with a spherical wrist by performing the algorithm once for the wrist centre to test if its path is inside the workspace of the arm, and a second time for the end-effector to test if its path is inside the workspace of the wrist. For the arm, x_i and l_i are Cartesian values that are defined at the wrist centre. For the wrist, x_i and l_i are angular values that are defined at the end-effector with respect to the wrist centre. Note that the position function $X(l_i,P)$ (5) of the arm must account for the effect of orientation changes on the position of the wrist centre.

The algorithm can not be applied to manipulators with configuration dependent singular positions unless those singular positions can be identified in task-space. In the latter case, the singular positions can be avoided by treating them as inadmissible positions regardless of the robot configuration.

PERFORMANCE

$X(l_i,P)$ and $D(x_i,W)$ are evaluated once per loop so, as long as they can be computed analytically, computation time is linear with the number of loop iterations. The lower $\underset{\sim}{i}$ and upper $\tilde{i}$ bound on the number of loop iterations are shown in (14) and (15) respectively. The lower bound occurs when the distance between the path's start position and the nearest workspace boundary is greater than the path length. In this case, the loop is performed once with x_0 equal to the path's start position and a second time with x_1 equal to the path's end position. The upper bound occurs when the entire path is exactly $2\underset{\sim}{d}$ away from the workspace boundary. In this case, the path is navigated in $2\underset{\sim}{d}$ increments.

$$\underset{\sim}{i} = 2 \tag{14}$$

$$\tilde{i} = ceil(L/2\underset{\sim}{d}) + 1 \tag{15}$$

A test trajectory (zigzag), shown on the left in Figure 5, is placed at four different locations in the workspace of the manipulator from Figure 2. At location 1 the path is near the outer edge of the dextrous workspace. At location 2 the path is near the inner edge of the dextrous workspace. At location 3 the path is far from all workspace boundaries and at location 4 the path is near both dextrous workspace boundaries. With $r_{min} = 100$, $r_{max} = 1000$, $\underset{\sim}{d} = 25$, and $L = 17501$, the number of iterations required by each of the paths are shown in Table 1.

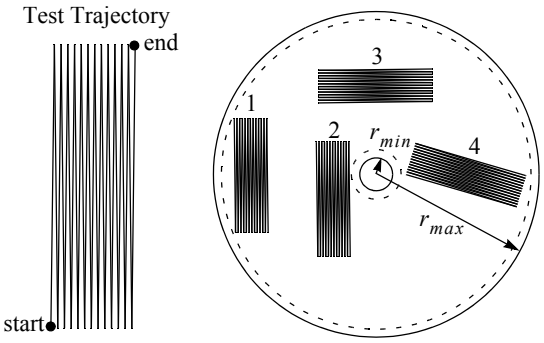

Figure 5. Algorithm Performance Test

Table 1. Number of Computations per Path

	Path 1	Path 2	Path 3	Path 4
Maximum Iterations	352	352	352	352
Required Iterations	97	83	47	65
Percent of Maximum	27.6%	23.6%	13.4%	18.5%

The algorithm executed an average of 5 times faster than the worst-case scenario and was fastest when the path was furthest from the workspace boundary. Since efficiency depends on the path's proximity to the reachable workspace boundary, the choice of $\underset{\sim}{d}$ trades off the execution time with the stringency of the passing criteria. A larger $\underset{\sim}{d}$ value results in larger minimum steps sizes but also shrinks the dextrous workspace making admissibility less likely.

Note that all of the example paths considered were admissible. When a path is inadmissible, the algorithm terminates once the workspace violation is identified so the ineration count will be even less.

THE ROBODOC® SURGICAL ASSISTANT

The ROBODOC manipulator is a 3-DOF SCARA arm with a 2-DOF (pitch/yaw) wrist. The Cartesian stage consists of an elbow manipulator with a prismatic vertical motion stage. The kinematic arrangement is simple enough that $D(x_i,W)$ can be computed analytically. A photograph of the ROBODOC manipulator is shown in Figure 6.

The proximity of a point x to the vertical workspace limit d_{vert} is taken from the residual motion range of the prismatic actuator q_2. Proximity of x to the horizontal workspace limit is computed from (16) through (21) where o_0 is the manipulator origin, o_1 and o_2 are the respective origins of joint q_1 when q_0 is at its upper and lower joint limits, r_{min} and r_{max} are the minimum and maximum reach of the arm (as a function of q_1 joint limits), a_0 and a_1 are the manipulator link lengths, α is the angle between the two vectors shown in Figure 7 and d_{inner}, d_{outer}, d_{left} and d_{right} are the computed distances between x and the planar workspace boundaries W_{inner}, W_{outer}, W_{left} and W_{right}.

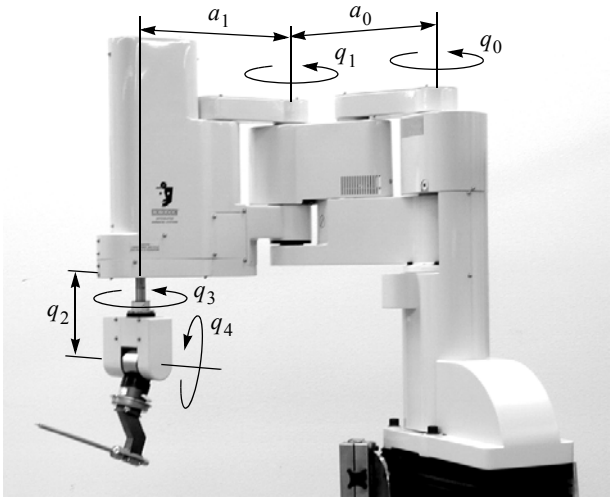

Figure 6. The ROBODOC® Manipulator Arm

In Figure 7, W_{inner} and W_{outer} are drawn using solid lines while W_{left} and W_{right} are drawn using dotted lines. Note that the ROBODOC manipulator has $a_0 = a_1$ and q_1 joint limits that allow the arm to extend completely but not to contract completely. Also note that the workspace shown in Figure 7 corresponds to a particular configuration (right elbow) of the robot. The ROBODOC manipulator supports both right elbow and left elbow configurations but is constrained to a single configuration during any one procedure. A non-zero $\dot{q}$ value in the path verification algorithm also enforces this constraint since the cut path is prevented from visiting the singular position that exists between elbow configurations (i.e. when the arm is completely extended).

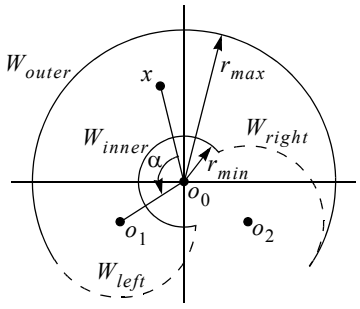

Figure 7. Planar Workspace of a SCARA Arm

$$d_{inner} = \|x - o_0\| - r_{min} \quad (16)$$

$$d_{outer} = r_{max} - \|x - o_0\| \quad (17)$$

$$\text{If } (\alpha > 0) \quad d_{left} = a_1 + \|x - o_1\| \quad (18)$$

$$\text{Else} \quad d_{left} = a_1 - \|x - o_1\| \quad (19)$$

$$d_{right} = \|x - o_2\| - a_1 \quad (20)$$

$$D(x, W) = min(d_{vert}, d_{inner}, d_{outer}, d_{left}, d_{right}) \quad (21)$$

The d_{left} and d_{right} values are the perpendicular distances from W_{left} and W_{right}. As such, (18) will often be greater than the true minimum distance to W_{left}. However, it is not necessary to compute the true minimum distance to W_{left} when $\alpha > 0$ since that value can never be smaller than both d_{inner} and d_{outer} and, therefore, will not affect (21).

Given a conservative deadband width $\dot{q}$ = 5mm and a rough estimate of the number of iterations one might expect in practice (30% of worst-case), the approximate path length, worst-case and expected numbers of loop iterations for typical, medium sized hip and knee implants such as those shown in Figure 1 are shown in Table 2.

Table 2. Approximate Number of Iterations

	Hip Stem	Knee (Femoral)	Knee (Tibial)
Total Path Length (mm)	5000	7000	3000
Maximum Iterations	500	700	300
Typical (30% Max.)	150	210	90

Given the low mathematical complexity of equations (16) through (21), $D(x_i, W)$ can be computed between 90 and 210 times in just a few seconds with even unsophisticated computer hardware. This is an acceptable delay during an arthroplasty procedure. Note, however, that in practice equations (16) through (21) are complicated by additional computations which take the roll q_3 and pitch q_4 axes into account and which account for deviations from the ideal kinematic model of the arm (i.e. calibration parameters). Consequently, the actual delay is slightly longer, but still acceptable.

If a cut path is found to be inadmissible by the algorithm, the surgeon is notified, the bone is re-positioned and the process is repeated. Because workspace violations are identified for each component of the workspace boundary in equations (16) through (21), guidance is provided as to how the bone should be moved to avoid further delays from subsequent path verification failures.

CONCLUSIONS

Fast computation of path admissibility is necessary in an unstructured environment such as orthopaedic surgery where anesthesia and tourniquet times must be kept to a minimum. A method is proposed that quickly determines path admissibility and, in some cases, can identify the violation of individual workspace constraints. The upper bound on computation time and the minimum distance from singular positions are configurable by a workspace deadband value. In practice, the algorithm performs significantly faster than its worst-case estimate with computation time dropping with increased distance between the desired path and workspace boundary.

The algorithm is applied to the ROBODOC Surgical Assistant to ensure uninterrupted cutting of implant cavities.

Due to the robot's simple kinematic arrangement, the algorithm is implemented analytically and the resulting execution time for a typical implant is well within acceptable limits.

ACKNOWLEDGEMENTS

The author gratefully acknowledges Dr. Peter Kazanzides, Jonathan Lazarus and Steve Cohan for their valuable input during the preparation of this manuscript.

REFERENCES

[1] Abdel-Malek, K., Yeh, H.J., 1998, "Path Trajectory Verification for Robot Manipulators in a Manufacturing Environment", *Proc. Instn. Mech. Engrs.*, V. 211, Part B, pp. 547-556.

[2] Latombe, J-C., 1991, *Robot Motion Planning*, Kluwer Academic Publishers.

[3] Mitchell, J.S.B., 1990, "Algorithmic Approaches to Optimal Route Planning", *Proc. SPIE Int. Soc. Optical Eng.* (Boston, Mass), V. 1388, pp. 248-259.

[4] Sheu, P.C-Y., Xue, Q., 1993, *Intelligent Robotic Planning Systems*, World Scientific Publishing Co.

[5] Brock, O., Khatib, O., 1998, "Executing Motion Plans for Robots with Many Degrees of Freedom in Dynamic Environments", *Proc. IEEE Int. Conf. Rob. Auto.* (Leuven, Belgium), pp. 1-6.

[6] Merlet, J.P., 1994, "Trajectory Verification in the Workspace for Parallel Manipulators", *Int. J. Rob. Res.*, Vol. 13, No. 4, pp. 326-333.

[7] Young, K.Y., Wu, C.H., 1991, "Path Feasibility and Modification Based on Robot Workspace Geometry", *Proc. 30th Conf. on Dec. Ctrl.* (Brighton, England), pp. 1027-1032.

[8] Ming, C.L., Manocha, D., Cohen, J., Gottschalk, S., 1996, "Collision Detection: Algorithms and Applications", *Algorithms for Robotic Motion and Manipulation, 1996 Workshop on the Algorithmic Foundations of Robotics*, pp.129-141.

[9] Kazanzides, P., Mittelstadt, B.D., Musits, B.L., Bargar, W.L., Zuhars, J.F., Williamson, B., Cain, P.W., Carbone, E.J., 1995, "An Integrated System for Cementless Hip Replacement", *IEEE Eng. Med. Bio. Magazine*, Vol. 14, No. 3, pp. 307-313.

[10] Kazanzides, P., 1999, "Robot Assisted Surgery: The ROBODOC Experience", *Proc. ISR 30th Int. Symp. Rob.* (Tokyo, Japan), pp. 281-286.