Multisensor Robotic System for Autonomous Space Maintenance and Repair

M. A. Abidi, W. L. Green, T. Chandra, and J. Spears

Department of Electrical and Computer Engineering
University of Tennessee
Knoxville, TN 37996-2100

ABSTRACT

Space applications can be greatly enhanced by the use of robotics and automation in activities such as orbital inspection and maintenance. Partial autonomy of space systems will enhance safety, reliability, productivity, adaptability, and reduction of overall cost. At the core of a robotics system is the ability to acquire, integrate, and interpret multisensory data to generate appropriate actions to perform a given task. In this paper, we show the feasibility of realistic autonomous space manipulation tasks using multisensory information. This is shown through two experiments involving a Fluid Interchange System and a Module Interchange System. In both cases, autonomous location of the mating element, autonomous location of a guiding light target, mating, and demating of the system are performed. Using the vision, force/torque, proximity, and touch sensors, the Fluid Interchange System and the Module Interchange System experiments were accomplished autonomously and successfully.

1 INTRODUCTION

Robots have become a major element of today’s industrial world. They have been beneficial in replacing humans not only in tasks at which robots are more efficient, but also those which humans find undesirable because they are strenuous, boring, difficult, hazardous, or simply more expensive or unachievable for humans. Robotics is being used currently in hazardous environments such as those encountered in chemical, nuclear, military, underwater, and space applications. A prominent example may be found in space applications such as on-orbit laboratories, servicing and repair facilities for spacecraft, and manufacturing and assembling space laboratories [1,2,3].

Space applications can be greatly enhanced by the use of robotics and automation. Activities such as collision-free navigation, rendezvous and docking, satellite retrieval and servicing, orbital inspection, maintenance, refurbishment and repair, and navigation of remotely piloted vehicles now require human assistance, often in environments that are hostile. At the present time, because of the "youth age" of the disciplines of artificial intelligence, robotics, and automation, these space activities are performed usually in manual mode, sometimes in teleoperated mode, but seldom in autonomous mode. This philosophy increases the level of involvement of astronauts raising further their exposure to danger. Autonomous operation should be the norm in hazardous environments. Total or partial autonomy of space systems will enhance safety, reliability, productivity, adaptability, and reduce the overall cost of space missions [4,5,6,7,8]. Each of these factors alone justifies automation.

1.1 Objective

The intent here is to demonstrate the feasibility of realistic autonomous robotic manipulations using multisensory information cues. The environment is variably structured. Only the relative position of the manipulator/end-effector with respect to the workpieces being manipulated is unknown when data is being sensed. This does not preclude the relative motion of objects once an experiment is started. This task is performed using integrated information from vision, force/torque, proximity, and touch sensing. The cooperative use of sensors is illustrated through the following two experiments which were suggested by NASA.

Fluid Interchange System (FIS): The purpose of this experiment is to autonomously mate and demate a fluid interchange system using multisensory data. A mock-up of the FIS has been designed and built. It is composed of a mating nozzle element, a receptacle, and a number of light targets.

Module Interchange System (MIS): The purpose of this experiment is to autonomously identify, remove, and replace a faulty or spent module (from orbiter) by a new module (from service). A mock-up of the MIS system was designed and built. It is composed of a mating module element, a mounting rack, and light targets. Here again, the system is manipulated using cues from the vision, force/torque, proximity, and touch sensing.
1.2 Experimental Set-Up

A mock-up of the FIS has been designed and built [Fig. 1]. The mock-up of the FIS is composed of an alignment/locking mechanism which consists of a nozzle and a receptacle mounted in a module rack. This module holds also the 4-light guiding target used by the vision system. The nozzle and receptacle are essentially a large male and female BNC-like connector. The nozzle is cylindrical and has 2 stainless steel pins located 180° apart, perpendicular to its outer surface which extend 0.50 inches. It has a flat, padded, parallelepipedic end that serves as an attachment point to the end-effector of the robot. The nozzle is 4.75 inches long. It consists of a cylindrical portion, 2.25 inches in height. Its inner and outer diameters are 1.25 and 1.75 inches, respectively. The attachment handle is 2.50 × 1.75 × 1.25 inches and is centered on the opposite end of the cylindrical portion from the nozzle opening.

The receptacle is mounted on a rectangular plate, 6.75 × 8.75 inches, which is in turn mounted on a module, 1.25 × 1.25 × 7.50 inches. The entire receptacle module system is 12.75 inches deep. It is mounted in a standard Nuclear Instrumentation Module system (NIM) rack. The receptacle has a flared rim and an inner and outer diameters of 2.00 and 2.90 inches, respectively. At a depth of 1.40 inches, a spring-loaded pressure plate is mounted; its travel distance is 0.25 inches. The receptacle has also 2 V-notch located 180° apart in its rim leading into 2 grooves that lock the nozzle into position once inserted.

The receptacle holds also the 4-light guiding target which will be used by the visual system to locate the position and orientation of the receptacle. The positions of the light target, with respect to the lower left-hand corner of the receptacle, are in inches: (0.25, 7.50), (0.25, 1.20), (0.50, 1.20), and (7.50, 0.50). The first dimension is the distance across the rack and the second dimension is the height.

The rack is 8.50 × 20.00 × 17.25 inches. The inner dimensions are 7.75 × 17.00 × 10.60 inches. Modules are guided into the rack by 12 sets of equally spaced (0.35 inches) strips placed along the inner top and bottom surfaces of the rack.

The initial 2-D position and orientation of the nozzle are unknown. Its height, however, is known. The 3-D position and orientation of the receptacle holding the target are unknown. The only constraint in this experiment is that the FIS (nozzle, receptacle, and FIS target) be positioned in the field of view of the camera and be within reach of the end-effector. The world coordinate system [Fig. 2] is defined with respect to the robotic workstation table and is divided into 2 areas: the mating element area and the target area. This means that in both experiments, the robotic system is expected to find the mating element anywhere in the mating area and the locate the light target anywhere in the target area.

![Figure 1: The Fluid Interchange System (FIS) with nozzle, receptacle, and light targets.](image1.png)

![Figure 2: World coordinate system with respect to the robotic workstation.](image2.png)

All the dimensions mentioned above along with the information corresponding to the relative positions of the objects being manipulated in this experiment are stored in a database easily updatable by the operator and directly accessible by each sensor module during the running of the operation.

1.3 Description of the Experiments: FIS and MIS

The experiment is performed using a Cincinnati Milacron T3 - 726 six-degree-of-freedom industrial robot. Several sensors have been added to this arm at the end-effector level. They are vision, force/torque, proximity, touch, range, and sound [Fig. 3]. (The range and sound sensors are not used in this experiment because they both use ambient air as a medium of transduction, hence not appropriate for space applications.) The control of the robot motion as well as the positioning, activation of the sensors, and data acquisition are performed using Fortran and C languages running on a VAX 11/785-VMS.

The experiment can be summarized as follows. A mating element appears anywhere in the mating area. Its 3-dimensional pose is known except for its 2-dimensional position. The mating element is within the field of view of a camera rigidly mounted on the arm and is within reach of the robot gripper. The light targets are within the field of view of the camera when the robot is in a predetermined, yet arbitrary, position. The paradigm described in the previous section is followed here for the execution of the experiment. The robot is moved by the human operator such that the light target is visible, then the robot is put in autonomous mode. The robot determines the approximate position of the mating element using the visual sensor. This measurement is later refined once the robot knows the approximate position of the target. Afterwards, the robot position itself high above the table so that the mating area is within the field of view of the camera and uses its vision sensor to determine (approximately) the 2-dimensional position of the mating element. This measurement is later refined by moving the robot arm closer to the mating element. The robot then picks up the mating after checking for the presence of the mating element using first its proximity sensor then its touch sensor.

SPIE Vol. 1006 Space Station Automation IV (1988) / 105
vision system. At the same time, we will mention some similar existing systems that were reported in the literature.

2.1 2-D position and Orientation of Mating Elements

Here, we describe a method for determining the arbitrary 2-D position and orientation of the mating elements of the FIS and MIS using a vision sensor. Many of the vision-based systems used for path planning for mobile robots, automated inspection, and mechanical assembly use similar techniques. A summary of these techniques may be found in [9,10,11]. In this system, several preprocessing steps are done to the image depicting the mating elements. First, the grey-scale image, \( I \), is transformed to a binary image, \( I' \), using thresholding [1]. The characteristic function for this image, \( f(x, y) \), is equal to “0” for image points on the background and “1” for points on the object.

We choose a point which represents the position of a 2-D object. In this case, the center of area of the object was selected [2]. To calculate the center of area, \( (\bar{x}, \bar{y}) \), we find the area, \( A \), and the first moments \( M_x \) and \( M_y \) of the object using the following relationships.

\[
A = \int \int f(x, y) \, dx \, dy, \quad (1)
\]

\[
M_x = \int \int xf(x, y) \, dx \, dy, \quad (2)
\]

\[
M_y = \int \int yf(x, y) \, dx \, dy. \quad (3)
\]

The coordinates of the center of area \( \bar{x} \) and \( \bar{y} \) are computed as follows

\[
\bar{x} = M_x / A \quad (4)
\]

and

\[
\bar{y} = M_y / A. \quad (5)
\]

For an \( N \times N \) discrete image,

\[
\bar{x} = \frac{\sum_{x=1}^{N} \sum_{y=1}^{N} xf(x, y)}{\sum_{x=1}^{N} \sum_{y=1}^{N} f(x, y)}, \quad (6)
\]

\[
\bar{y} = \frac{\sum_{x=1}^{N} \sum_{y=1}^{N} yf(x, y)}{\sum_{x=1}^{N} \sum_{y=1}^{N} f(x, y)}. \quad (7)
\]

The usual practice to determine the 2-D orientation of an object is to compute the axis of least second moment or moment of inertia [2]. To accomplish this, we find the line for which the integral of the squared distance to points in the object is minimum, i.e., we minimize \( E \) w.r.t. \( \theta \):

\[
E = \int \int r^2(x, y)f(x, y) \, dx \, dy, \quad (8)
\]

where \( r(x, y) \) is the minimum distance from the point \( (x, y) \) to the line being sought. To represent a line, we use the two parameters \( \rho \) and \( \theta \) (\( \rho \) is the perpendicular distance from the origin to the line; \( \theta \) is the angle between the \( x \)-axis and the line, measured counter-clockwise) [Fig. 4]. The minimization of \( E \) yields the following orientation angle.

\[
\theta = .5 \sin^{-1} \left( \frac{2M_{xy}}{\sqrt{4M_{xx}^2 + (M_{xx} - M_{yy})^2}} \right). \quad (9)
\]
where \( c_{a1}, c_{a2}, c_{a3}, \) and \( c_{a4} \) represent the elements of the homogeneous coordinates. In this paper, we use the pinhole camera model for our vision sensor [Fig. 5]. The image coordinates \((x, y)\) and the world coordinates \((X, Y, Z)\) are related as follows [1,17,18]:

\[
x = \frac{\lambda x^c}{\lambda - z^c},
\]

\[
y = \frac{\lambda y^c}{\lambda - z^c},
\]

\[
W^c = AW,
\]

where

\[
W = (X, Y, Z, 1)^t,
\]

\[
W^c = (X^c, Y^c, Z^c, 1)^t,
\]

\[
A = \begin{bmatrix} a_1 & a_2 & a_3 & a_{10} \\ a_4 & a_5 & a_6 & a_{11} \\ a_7 & a_8 & a_9 & a_{12} \\ a_{13} & a_{14} & a_{15} & a_{16} \end{bmatrix}.
\]

The intermediate parameters \(X^c, Y^c,\) and \(Z^c\) represent positions in the camera coordinate system, and the elements \(a_1\) through \(a_{12}\) represent the matrix transformation \(A\) from world coordinates to camera coordinates. The parameter \(\lambda\) represents the focal length of the camera. By the very nature of the problem, the parameters \(a_1\) through \(a_9\) characterize the effect of rotation only, whereas \(a_{10}, a_{11},\) and \(a_{12}\) characterize the effect of translation.
here is to compute the elements of the matrix transformation, \( a_1 \) through \( a_{12} \), using the image of the target points as well as their relative position in order to determine the exact 3-dimensional position of those targets with respect to a coordinate system fixed relative to the camera.

For any world point \( W_i \), Eq. 17 may be rewritten in tensor notation as

\[
W_{ii}^a = A_{ij} W_{ij}.
\]  

(21)

Substituting in Eq. 15 and 16, we obtain

\[
\lambda x_i - x_i A_{3j} W_{ij} = \lambda A_{1j} W_{ij},
\]  

(22)

\[
\lambda y_i - y_i A_{3j} W_{ij} = \lambda A_{2j} W_{ij},
\]  

(23)

for each \( i \) and \( j = 1, \ldots, 4 \). Each point provides a pair of linear equations involving the unknown parameters \( a_1 \) through \( a_{12} \); therefore a minimum of 6 points is required to uniquely define the matrix \( A \). However, we have shown [16] that 4 is the minimum number of points required to completely compute \( A \). In the following, for the ease of mathematical manipulation, we use three points to compute a multiple solution for the matrix \( A \). Later, we add a fourth point to disambiguate this solution.

(A) The 3-Point Solution:

The matrix \( A \) describes the movement(s) that a point \((X, Y, Z, 1)\) has to undergo to map into \((c_{x1}, c_{x2}, c_{x3}, c_{x4})\). Here we investigate the use of the movement properties to reduce the number of points needed for computing \( A \). When the translation vector is null, the coefficients \( a_{10}, a_{11}, \) and \( a_{12} \) are zero, and \( a_1 \) through \( a_6 \) are functions of the rotational movement only. For a 3-degree-of-freedom movement, the system can be described by a rotation around the \( Z \)-axis with a parameter \( \theta \), a rotation around the \( Y \)-axis with a parameter \( \beta \), and a rotation around the \( X \)-axis with a parameter \( \alpha \). This can be expressed as follows.

\[
R_\beta = \begin{bmatrix}
\cos \beta & 0 & -\sin \beta & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]  

(24)

\[
R_\beta = \begin{bmatrix}
\cos \alpha & -\sin \alpha & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix},
\]  

(25)

\[
R_\alpha = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha & 0 \\
0 & -\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]  

(26)

\[
R_T = R_\alpha \cdot R_\beta \cdot R_\theta.
\]  

(27)

The transformation, \( R_T \), denotes the total movement. Equations 24 through 26 introduce implicitly the idea that this transformation preserves distance. Hence for \( R_T \),

\[
R_T = \begin{bmatrix}
a_1 & a_2 & a_3 & 0 \\
a_4 & a_5 & a_6 & 0 \\
a_7 & a_8 & a_9 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
\]  

(28)

the following holds true:

\[
a_1^2 + a_2^2 + a_3^2 = 1,
\]

(29)

\[
a_4^2 + a_5^2 + a_6^2 = 1,
\]

(30)

\[
a_7^2 + a_8^2 + a_9^2 = 1,
\]

(31)

\[
a_1^2 + a_2^2 + a_3^2 = 1,
\]

(32)

\[
a_1^2 + a_2^2 + a_3^2 = 1,
\]

(33)

\[
a_3^2 + a_5^2 + a_6^2 = 1,
\]

(34)

and

\[
a_1 a_2 + a_3 a_6 + a_7 a_8 = 0,
\]

(35)

\[
a_1 a_3 + a_4 a_6 + a_5 a_8 = 0,
\]

(36)

\[
a_2 a_3 + a_5 a_6 + a_8 a_9 = 0.
\]

(37)

In the case where the translation vector is not zero, the global movement can be described by a matrix similar to \( R_T \) which includes the coefficients \( a_{10}, a_{11}, \) and \( a_{12} \) characterizing the translation. Obviously, Eqs. 29 through 37 still hold true:

\[
A = \begin{bmatrix}
a_1 & a_2 & a_3 & a_{10} \\
a_4 & a_5 & a_6 & a_{11} \\
a_7 & a_8 & a_9 & a_{12} \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]

(38)

To complete the set of equations to a state where the 12 unknowns are uniquely defined, we shall write 6 independent equations. Since each image point gives 2 equations, 3 well-distributed points are sufficient. If the three target points are taken such that they form a right triangle, then these equations yield a closed form solution for \( A \).

This method involves several preprocessing steps which allow one to make some simplifying assumptions about the problem geometry and also indicate the adjustment to be applied to the solution to compensate for these steps. Throughout this paper, we use the vector \( W_i \) to represent the augmented vector \((X_i, Y_i, Z_i, 1)\) where \((X_i, Y_i, Z_i)\) are the \((X, Y, Z)\) coordinates of point \( P_i \). The 3 target points will be denoted by \( P_1, P_2, \) and \( P_3 \). Any superscript on \( W \) indicates a corresponding superscript on \( X, Y, \) and \( Z \), and refers to the coordinates of the point using a system other than the fixed world system. These transformations are shown in Fig. 6 and are summarized as follows herein.

![Figure 6: Transformation of the world coordinate system into the ideal position, transformation of the camera coordinate system into the standard position, and relationship between the two transformations.](image)

\( W_i \): Original World Coordinates

\( W_i^\circ \): Ideal Position Coordinates

\( W_\cdot^\circ \): Standard Position Coordinates

\( W_i^\circ \): Original Camera Coordinates
The first step transforms a general problem into an intermediate coordinate system in which the three target points are in what we shall define as standard position. Standard position is characterized by \( P_0 \) at the origin, \( P_1 \) on the positive \( X \)-axis, and \( P_2 \) in the first quadrant of the \( XY \)-plane (or on the positive \( Y \)-axis). This transformation will be denoted by \( T \); its effect on the recovery of the target position is examined in Appendix 3 of [3].

The second preprocessing step must be performed on each image to be processed. It involves another transformation, which describes how the image would look to a camera position which has its lens center in the same location as the real camera but is located directly at the origin of a given \( \alpha \) system and has its \( x \)-axis in the same plane as the \( X \)-axis of the \( \alpha \) system. We will refer to this "fictitious" camera as being in ideal position and will call the image seen by such a camera an ideal image. Such an image is characterized by the image of \( P_0 \) at the origin and the image of \( P_1 \) on the positive \( x \)-axis. This transformation will be denoted by \( T_i \); its effect on the recovery of the target position is examined in Appendix 2 of [3].

The transformation relating the coordinates of a given point from the ideal position to the standard position is described by matrix \( C \). The computation of \( C \) and the combined effect of \( T \) and \( T_i \) is given in Appendix 3 of [3]. (The transformation \( B \) is computed as an intermediary step in the computation of \( C \).)

For implementation purposes, the three previous operations are summarized as follows.

**STEP 1**: Compute the transformation matrix \( T \).
**STEP 2**: Compute the transformation matrix \( T_i \).
**STEP 3**: Compute the transformation matrix \( C \).
**STEP 4**: Compute the transformation matrix \( A \) using \( A = T_i^{-1}CT \).

(B) Adding a Fourth Point

Sometimes, solving for the matrix \( C \) causes the solution for \( A \) to be multiple. To disambiguate this solution, we add a fourth point. Applying the same procedure to 4 coplanar points, \( P_0, P_1, P_2, \) and \( P_3 \), yields a system of linear equations in \( C \). Using the preprocessing steps given above, we have \( x_0, y_0, z_0, x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_0', y_0', z_0', x_1', y_1', z_1', x_2', y_2', z_2', x_3', y_3', z_3' \), all zero. Therefore,

\[
C_{14} = C_{24} = C_{21} = 0. \quad (39)
\]

The 6 unknowns which are not zero are related by 5 equations:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
\lambda x_1' & 0 & 0 & x_1 x_1' & 0 & x_1' \\
\lambda y_1' & 0 & 0 & x_1' & x_1 & x_1' \\
\lambda z_1' & 0 & 0 & x_1' & x_1 & x_1' \\
0 & 0 & 0 & x_1' & y_1' & y_1' \\
0 & 0 & 0 & y_1' & x_1' & y_1' \\
0 & 0 & 0 & y_1' & x_1' & y_1' \\
\end{bmatrix}
\begin{bmatrix}
C_{11} \\
C_{12} \\
C_{13} \\
C_{31} \\
C_{32} \\
C_{33} \\
\end{bmatrix}
= \begin{bmatrix}
C_{11} \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix} \quad (40)
\]

or

\[
MC' = C_0. \quad (41)
\]

Note that the first row of Eq. 40 does not carry any information; it was added to make the matrix, \( M \), a square one. Since we have 6 unknowns and 5 equations, we need to add one more equation. Using the facts that \((C_{11}, C_{21}, C_{31})'\) is orthonormal and \(C_{21} = 0\), we may write

\[
C_{11}^2 + C_{21}^2 = 1. \quad (42)
\]

Because of the nature of Eq. 41, all the unknown terms can be written as a function of \( C_{11} \). If we denote \((-M^{-1})_{11}\) by \( r \) we can write

\[
C_{11} = \pm \sqrt{\frac{1}{1 + r^2}}. \quad (43)
\]

Without inverting \( M \), we can compute \( r \) by combining the equations given in the preceding system:

\[
r = \frac{\lambda x_1'B_3 - x_1'B_1}{x_1'(x_1'B_3 - B_2)} = -\frac{C_{31}}{C_{11}}, \quad (44)
\]

where

\[
B_1 = \lambda y_1'y_3'(y_1'' - y_3')(X_2''Y_3'' - X_3''Y_2''),
B_2 = y_1'y_3'(y_1'' - y_3')(X_2''Y_3'' - X_3''Y_2''),
B_3 = y_1'y_3'(y_1'' - y_3')(X_2''Y_3'' - X_3''Y_2''). \quad (45)
\]

Although Eq. 43 suggests that \( C_{11} \) can take 2 distinct values, the truth is that this cannot happen because

\[
\text{if } r > \frac{\lambda}{x_1}, \quad C_{11} > 0,
\]
\[
\text{if } r < \frac{\lambda}{x_1}, \quad C_{11} < 0.
\]

The case where \( C_{11} = 0 \) leads to an infinite value of \( r \) or \( P_1, P_2, \) and \( P_3 \) being collinear, which contradicts our first assumption about the relative position of these points. Once \( C_{11} \) is computed the remaining parameters are readily computed. The matrix \( A \) is computed using the relationship \( A = T_i^{-1}CT \). The coordinates of \( P_0, P_1, P_2, \) and \( P_3 \) are each determined by inverting \( A \) for each corresponding target.

3 Force/Torque, Proximity, and Touch Sensing

3.1 Force/Torque Sensing

Force and torque sensing generally refer to the measurement of the global forces and torques applied on a workpiece by a robot manipulator. The force/torque sensor used in this work is the Lord FT-30/100 [Figs. 3 and 7]. The sensor is designed for wrist mounting. It uses 8 piezoresistive strain gauges that provide real-time feedback of the location, magnitude, and the orientation of forces and moments.

The system is equipped with a processor that resolves the eight raw pieces of data into six cartesian forces and torques at a rate of 100 Hz. The converted force and torque measurements can be referenced with respect to any element in the arm including to the sensor itself. This allows the cancellation of the effect of the load of the sensor, end-effector or tool, or workpiece which would, otherwise, degrade the quality of measurements. This system has a maximum force and torque capacity of 30 lb and 100 in-lb, respectively. The system resolution is about 1 oz and 1 in-oz for force and torque, respectively. A one-chip Z8 microcomputer has been added to this system for communication between the sensor and the main processor for sensor biasing, threshold setting, and interface to other sensors [19,1].
3.2 Proximity Sensing

Proximity sensing usually involves detection of a disturbance brought to a monitored signal by the presence/absence of an object within a specific volume around the sensing element. As opposed to range sensors which provide the actual distance between the targeted object and the sensor, proximity sensors provide only a binary output indicating if this distance is below/above a predetermined threshold distance. Most of the proximity sensors are either inductive, magnetic, capacitive, ultrasonic, or optical.

This variety in sensing modality yielded an equal variety in characteristics. Some sensors are capable of detecting the presence of objects from great distances averaging several feet (ultrasonic and optical); others are capable of sensing only within a few millimeters (inductive and magnetic). Capacitive, ultrasonic, and optical sensors are capable of detecting most material though with significantly varying sensitivity –highly dependent on surface reflection properties and orientation–, whereas inductive and magnetic sensors are sensitive only to metals or more specifically ferrous metals.

The proximity sensor used in this experiment is optical; it is made of two solid-state infrared light emitters coupled with two solid-state light receivers arranged in the configuration shown in Fig. 3. It is designed to detect the presence of an object between the parallel jaws of the end-effector when the beam is interrupted. Conversely, it detects the presence of an object in front of the end-effector when a return beam is sensed. In this experiment, the use of the proximity sensor is limited to collision avoidance during the manipulation phase [19,1].

3.3 Touch Sensing

Sensory feedback is of great importance in many intelligent robotics applications requiring recognition, acquisition, and handling of complex workpieces. A touch sensor can provide a robot with valuable information that can be crucial in each of these stages. Touch sensors presently used in industry are often simple and crude devices with limited capability. As with other sensing modalities, the state-of-the-art in touch sensing is still very primitive. Binary touch sensors, the simplest type, indicate only the presence/absence of an object. This is sometimes referred to as simple contact sensing or simple touch. Despite its simplicity, binary sensing provides valuable information [1,19].

In this work, 2 microswitches are mounted inside the end-effector textured compliant surfaces. They are used primarily for inspecting the presence/absence of a workpiece between the end-effector jaws. The manipulation experiment is halted whenever the actual reading is different from that expected [Fig. 3].

4 Experimental Results

In this section, we describe the autonomous mating and demating of the Fluid Interchange System (FIS) and the Module Interchange System (MIS) using the concepts presented in the previous two sections.

4.1 The FIS: Set-up and Experiment

The set-up relative to this experiment was shown in Section 1 [Fig. 1]. The FIS experiment involves determining the 2-D position and orientation of the nozzle as well as the 3-D position and orientation of the light target, then performing mating and demating of the interchange system.

First, the robot moves the camera to a predefined position high above the mating elements area [Fig. 8] then...
turns on its own lights to determine the 2-D position of the nozzle. An image is acquired, histogram–equalized, and thresholded to extract the object (nozzle) from the background (table) [Fig. 9(a) and (b)]. The image acquired here is 128 × 128 in size. Using Eqs. 4 and 5, the center of area of the nozzle is computed. In the experiment profiled here, it is found to be: \((i_1, j_1) = (72, 55)\). The parameters \(i_1\) and \(j_1\) are the row and column pixel coordinates, respectively. Using the camera geometry, the corresponding coordinates with respect to the world coordinate system are: \((X_1, Y_1, Z_1) = (14.02, 24.14, 4.75)\). The \(Z_1\) coordinate, which is the height of the nozzle, is known. The dimensions are measured in inches with respect to the world coordinate system. Usually this measurement is rough, and thus cannot be relied upon to pick up the nozzle.

Second, the robot moves the camera closer, directly above the nozzle [Fig. 10], and acquires another image of the nozzle to determine its position and orientation more accurately. This image is similarly processed [Fig. 9(c) and (d)] and the pixel coordinates of the center of area are computed. In this case, they are: \((i_2, j_2) = (76, 63)\).

Using the camera geometry, the corresponding coordinates with respect to the world coordinate system are: \((X_2, Y_2, Z_2) = (13.23, 24.18, 4.75)\). The \(Z_2\) coordinate, which is the height of the nozzle, is known. Using the scheme presented in Section 2, the orientation is calculated to be 47.1° with respect to the X-axis. Once these values are known, the robot then picks up the nozzle [Fig. 11] and proceeds to locate the receptacle. After the nozzle is picked up, the touch sensor is activated to check if the nozzle is properly held between the parallel jaw of the end-effector. If the outcome is negative, the robot makes another attempt to pick up the nozzle. To locate the receptacle, the robot moves the camera to a predefined position so that the target area is within the field of view of the camera [Fig. 12]. It turns on the 4 indicator lights on the rack containing the receptacle, acquires an image of the scene [Fig. 13(a)], and then acquires another image [Fig. 13(b)] after it turns off the FIS target. The two images are subtracted [Fig. 13(c)] and the resulting patterns are thinned [1] to find the position of the 4 lights (target) in the image. Their respective pixel coordinates are:

<table>
<thead>
<tr>
<th>Point</th>
<th>i</th>
<th>j</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P_0)</td>
<td>46</td>
<td>28</td>
</tr>
<tr>
<td>(P_1)</td>
<td>82</td>
<td>29</td>
</tr>
<tr>
<td>(P_2)</td>
<td>81</td>
<td>63</td>
</tr>
<tr>
<td>(P_3)</td>
<td>45</td>
<td>61</td>
</tr>
</tbody>
</table>

Using the algorithm presented in Section 2, we find the 3-D position and orientation of the receptacle. The resulting position vector in this experiment is:

\[(X_1^0, Y_1^0, Z_1^0, \theta_1, \beta_1, \alpha_1) = (23.49, -19.74, 9.76, 62.4, 94.3, 90.9)\].
The parameters $X^0$, $Y^0$, and $Z^0$ are in inches; $\theta_1$, $\beta_1$, and $\alpha_1$ are in degrees. At this moment, the robot has an approximate idea about the position of the mating rack. To refine its measurement, the robot moves the camera closer to and directly in front of the receptacle [Fig. 14] and then repeats the measurement of the 3-D position and orientation. [Fig. 15(a) through (c)]. The resulting pixel coordinates for the 4 light targets are:

<table>
<thead>
<tr>
<th>Point</th>
<th>i</th>
<th>j</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_0$</td>
<td>29</td>
<td>103</td>
</tr>
<tr>
<td>$P_1$</td>
<td>29</td>
<td>27</td>
</tr>
<tr>
<td>$P_2$</td>
<td>110</td>
<td>28</td>
</tr>
<tr>
<td>$P_3$</td>
<td>110</td>
<td>104</td>
</tr>
</tbody>
</table>

The corresponding position and orientation of the receptacle are calculated as:

$$(X^2, Y^2, Z^2, \theta_2, \beta_2, \alpha_2) = (23.21, -19.46, 9.43, 65.9, 94.3, 88.3)$$

Note the low disparity between the two measurements for $X^0$, $Y^0$, and $Z^0$. By contrast, the correction in the angles is more significant. This was typical of this experiment.

Third, the robot proceeds to insert the nozzle [Fig. 16]. At this time, the proximity sensor is activated during the approach of the end-effector to the mounting element (the 3-dimensional position and orientation of which is known). If the proximity sensor detects the mounting rack at its expected position, it proceeds normally for the insertion phase. Otherwise, the complete operation is halted. The flared rim of the receptacle guides the nozzle into the receptacle during the FIS mating/demating operations by creating forces on the nozzle. As the nozzle is inserted, the stainless steel pins enter the V-notches which further enhance the alignment of the nozzle. Once the nozzle is inserted, the pins reach the bottom of the V-notches. The robot then rotates the nozzle clockwise by 15°. While the nozzle is rotating, the pins enter the grooves in the receptacle, and the nozzle is then released. Under the effect of the forces exerted on it by the spring-loaded pressure plate, the nozzle locks into place and rests in the notches which are located at the end of the grooves. The pressure plate holds the pins locked into the notches, and holds the nozzle in place. The mating of the nozzle into the receptacle is completed [Fig. 17]. The demating operation of the FIS is performed using the same steps to determine the position of the receptacle. The manipulation part is simply the reverse action.

In the future, the FIS will include a fluid-tight connector which will consist of a male connector which is inside the flared rim, a pressure plate assembly, and a female connector (the inner surface of the nozzle). The male connector will include an O-ring which will achieve the fluid-tight connection of the mated FIS. The role of force/torque, proximity, and touch sensors in the MIS experiment is identical to that undertaken during the FIS.

### 4.2 The MIS: Set-up and Experiment

The goal here is to autonomously identify, remove, and replace a faulty or spent module by a new module from a
The MIS experiment: the robot proceeds to insert the nozzle into the receptacle.

The MIS experiment: the robot completes the insertion of the module into the mounting rack.

and demating of the system are performed. We implemented vision-driven techniques that determine the arbitrary 2-dimensional position and orientation of the mating elements as well as the arbitrary 3-dimensional position and orientation of the light targets. The system is also equipped with a force/torque sensor that continuously monitors the six components of force and torque exerted on the end-effector. Both experiments were successively accomplished on mock-ups built for this purpose using vision, force/torque, proximity, and touch, regardless of the initial position of the end-effector, mating element, and light targets. This method is also immune to variations in the ambient light. This is particularly important because of the 90-minute day-night shift in space.

5 Conclusions

In this paper, we described a robotics system that autonomously performs realistic manipulation operations suitable for space applications. This system is capable of acquiring, integrating, and interpreting multisensory data to locate, mate, and demate a Fluid Interchange System and a Module Interchange System. In both experiments, autonomous location of a guiding light target, mating,

References


