On-Line Compensation of Mobile Robot Docking Errors

KOSTIA MANDEL, MEMBER, IEEE, AND NEIL A. DUFFIE

Abstract-One of the main problems of an unconstrained mobile robot (that is not limited to rails or tracks), that limits its industrial applications, is its docking accuracy. Because robot programs are executed relative to the robot base, inaccurate positioning of the mobile robot during docking must be accounted for in order to improve the accuracy of the mobile robot. In this paper, a method to compensate for the docking inaccuracy of mobile robots is proposed. The method is based on modifying the task of the robot arm according to the docking error-the offset between the desired and actual docking locations of the mobile robot. The docking error is sensed by a sensor mounted on the robot arm: it can be either a vision system or a touch trigger probe. The algorithms for calculating the spatial docking error for each sensor and how the robot's task is modified accordingly are presented. The need for a method that will allow to reach results even in the presence of perturbed data, something that currently applied methods cannot guarantee, is discussed and one is presented. The calculation of the spatial offset between the actual and the desired locations of the mobile robot using vision system was implemented and results of some experiments are presented and discussed.

I. INTRODUCTION

ONE OF THE shortcomings of FMS today is the limitations of the available flexible material-handling equipment. An industrial robot that will have the capability to move around rigidly mounted on an industrial-quality, remotely controlled vehicle (like an AGV) seems to be the answer to most of those shortcomings. A major problems facing such a system is the accuracy of the AGV in docking (arriving to a station), which is currently in the order of 1 in. As robot programs are executed relative to the robot base, inaccurate positioning of the mobile robot base must be improved by more than an order of magnitude in order to be considered for industrial use.

A. Mobile Robot Overview

Before a world task (a general list of operations) can be performed by a robot, two conditions must be fulfilled: 1) the world task has to be described as a robot task-a robot specific list of end effector locations, velocities, and duties defined with respect to the robot base; 2) the robot task has to be transformed into a robot program-a set of commands issued to the robot's joint actuators.

The relationship between robot task and robot program is defined by the robot's kinematic equations: the inverse solution of the kinematic equations transforms any robot task to the corresponding robot program. The relationship between a world task and a robot task is defined by the offset between the world frame (in which the world task is specified) and the robot frame (in which the robot task is specified). For a stationary robot this relationship is fixed, as the robot location is fixed; while for an AGV-mounted robot the relationship between a world task and a robot task are not fixed due to the positioning inaccuracies of the AGV in docking. Each time the mobile robot will arrive to a certain station, its location will be perturbed about the designated location.

Two approaches can be taken in order to solve the docking problem. In the first approach, the docking problem can be solved by reducing the mobile robot's problem to a stationary one., An external system (either mechanical fixture that will use force, or a three-dimensional servo positioning device that will use fine controlled motion) will position the mobile robot at the designated location in every docking. According to the second approach, the robot task should be modified according to the docking error. The offset between the designated location of the robot, for which the robot task was specified, and its actual location will be evaluated in each docking. The offset (expressed as a transformation matrix) will then be used to modify the original robot task such that it will perform the desired world task from the new location, using the modified robot task.

In this paper a solution that uses the second approach to solve the docking problem of a mobile robot is presented. This approach was chosen due to its flexibility, expendability, and low implementation cost.

B. Solution Overview

In general, the solution method can be categorized as referencing or calibration of two frames. The offset between the actual location of a mobile robot (frame S in Fig. 1) and the designed robot location (frame Q in Fig. 1) cannot be measured as the Q frame is imaginary. Thus the transformation between the Q frame and the S frame cannot be derived directly. Each frame, however, can be related to a fixed reference frame attached to the station being served (frame F in Fig. 1). The offset between the Q and the F frames can be derived either from physical measurement or from an assumed offset. Then the original robot task can be transformed from the Q to the F frame. In subsequent dockings, when the mobile robot will be in location S, the offset between the fixed frame F and the actual frame S will be sensed and the appropriate

Manuscript received June 9, 1986; revised June 30, 1987. This paper was presented in part at the 7th International Conference on Robotics and Factories of the Future, San Diego, CA, July 28-31, 1987.

K. Mandel is with AT&T Bell Laboratories, Columbus, OH 43213.

N. A. Duffie is with the Mechanical Engineering Department, University of Wisconsin-Madison, Madison, W1 53706.

IEEE Log Number 8718841.

0882-4967/87/1200-0591$01.00

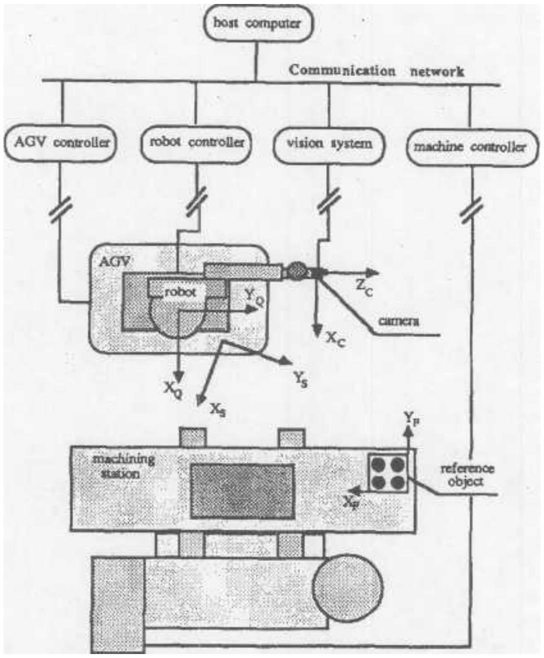

Fig. 1. The experimental system with the physical locations of the frames used.

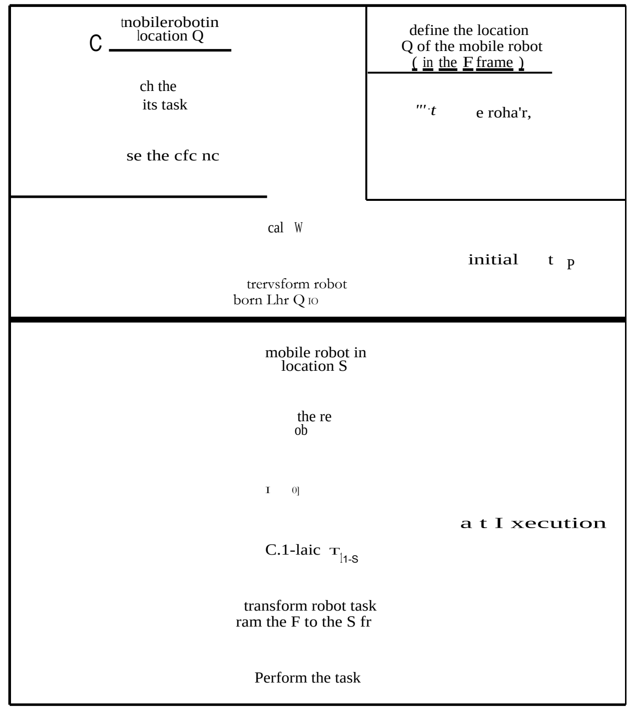

Fig. 2. Transformation of a robot task from the imaginary to the actual robot location.

transformation will be derived. That transformation will be used to transform the robot task, then specified with respect to the F frame, to the S frame where the robot will actually be. This process is illustrated in Fig.2.

The relationship between the coordinates of points in any two frames (say, Q and F) can be expressed in the form of a homogeneous transformation matrix

$$\begin{bmatrix} X_1 & X_2 & \cdots & X_n \\ Y_1 & Y_2 & \cdots & Y_n \\ Z_1 & Z_2 & \cdots & Z_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}_F = \begin{bmatrix} R & L \\ 0\;0\;0 & 1 \end{bmatrix}_{Q-F} * \begin{bmatrix} X_1 & X_2 & \cdots & X_n \\ Y_1 & Y_2 & \cdots & Y_n \\ Z_1 & Z_2 & \cdots & Z_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}_Q \tag{1}$$

or in matrix notation

$$P^F = T^{Q-F} * P^Q. \tag{2}$$

Two methods can be employed to calculate the transformation matrix between two frames from *theoretical* position, in both frames, of a reference object [1], [2]. In the first method, the transformation matrix is decomposed into three homogeneous rotation matrices and a homogeneous translation matrix. A sequence of pure rotations is assumed and geometrical relationships are used to find the rotation angles corresponding to the assumed rotation sequence. The rotation angles are substituted back into the rotation matrices that, when multiplied in the assumed sequence, define the rotation portion of the transformation matrix (R in (1)). The translation vector is then calculated by solving the three linear equations that relate the coordinates of one point in both frames. In the second approach, the twelve unknown entries of the transformation matrix are calculated directly, from the known position of four reference points, by solving the twelve linear equations (1) is composed of ($n = 4$). Because a real-time solution is desired, the direct evaluation approach was chosen for the calculation of the transformation matrix. It requires the solution of only linear equations while the assumed rotation sequence approach requires the solution of six nonlinear equations. The derivation of the theoretical transformation matrix using two different data acquisition methods is described in Section II of this paper.

In theory, the available data are absolutely accurate at all times, in reality they are not. Errors in measured position data have significant impact on the calculation of the transformation matrix. When the derivation of a transformation matrix from measured data is attempted, two problems have to be addressed: 1) how to make sure that the calculated transformation matrix is a rigid body, homogeneous transformation matrix and not simply a coefficient matrix that solves the linear set of equations; 2) how to make sure that the calculated transformation matrix is the most stable one and will be less affected by perturbation in the measured position data than any other transformation matrix. The perturbation analysis is discussed in Section III of this paper.

Some results of the experiments conducted to demonstrate the validity and measure the accuracy of the proposed transformation matrix derivation method are presented in Section IV.

II. THEORETICAL TRANSFORMATION MATRIX DERIVATION

In order to derive the transformation matrix, the locations of several specific points in both the mobile frame and the fixed frame are required. Two sensors were chosen to measure those locations: touch trigger probe and vision system. Each sensor has different characteristics thus requiring a different data processing algorithm for calculating the theoretical transformation matrix.

A. Touch Trigger Probe Application

Because the actual location of the mobile robot's base in the fixed frame F (see Fig. 1) is undefined, so is the location of its end effector. If the location of the end effector in the fixed frame could be determined, due to the known relationship between the end effector and the robot frame, the location of the mobile robot's frame (Q or S) would be defined as well.

In order to define the location of the end effector in the F frame, a cube is attached to the station the mobile robot is serving. The location of the cube with respect to the station is defined and the fixed frame is attached to a cube, as shown in Fig. 3. The touch trigger probe is mounted as an end effector, and the robot is programmed to reach with the tip of the probe's stylus the cube's center (the determination of the cube's size is discussed in [6]). The touch trigger probe is used to generate an interrupt to the robot controller when the tip of its stylus touches the cube's surface. The interrupt causes the controller to stop moving the arm and to record the position of the end effector (the tip of the touch trigger probe's stylus) in the robot frame. This position is sent to the host computer where it is combined with position data of other reference points to derive the transformation matrix between the fixed and the mobile frames using (1). Equation (1) could be solved directly for the transformation matrix T_{Q-F} if the reference point matrices P_i were known in both frames, which is not the case here. Because of. the AGV's poor repeatability it is impossible to tell ahead of time where the stylus will contact the reference cube-a reference point.

It is enough to know the coordinates of three nonlinear, *specific,* points on rigid body in order too define the location of the body in space. If the three sets of coordinates do not correspond with specific points, the body's location is undefined. In the case here, the exact locations of the reference points on the cube are not known, thus there can be many cubes (frames) that will fit any three reference points. The minimum number of points needed to uniquely define a cube in space is six: three on one surface, two on an adjoining surface, and one on a surface perpendicular to the first two, as illustrated in Fig. 3.

Due to the special positioning of the F frame and the reference points on the cube, it is known on which face of the cube each reference point lies, or which coordinate of that point is zero. Using this information, (1) becomes (3)

$$\begin{bmatrix} X_1 & X_2 & X_3 & X_4 & & \\ 0 & 0 & 0 & Y_4 & & \\ Z_1 & Z_2 & Z_3 & 0 & 0 & Z_6 \\ 1 & 1 & 1 & & & 1 \end{bmatrix}_F = \begin{bmatrix} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}_{Q-F} * \begin{bmatrix} X_1 & X_2 & X_3 & & & X_6 \\ Y_1 & Y_2 & Y_3 & & & Y_6 \\ Z_1 & Z_2 & Z_3 & & & Z_6 \\ 1 & 1 & 1 & & & 1 \end{bmatrix}_Q \quad (3)$$

It can be observed that in (3) there are only six independent linear equations. The other six equations needed to solve for the transformation matrix T_{Q-F} are generated from the orthogonality conditions of the rotation part of the transformation matrix. Those six equations are nonlinear, which complicates the solution. Further development was done in order to eliminate the necessity to solve six nonlinear equations.

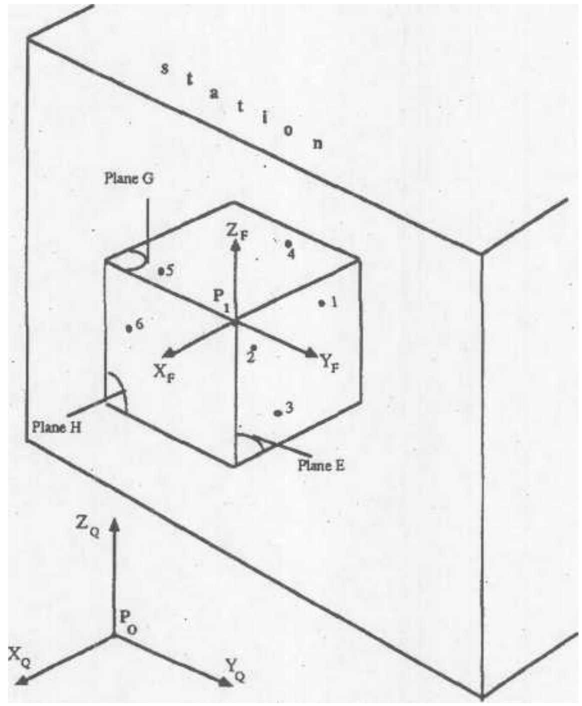

Fig. 3. The reference cube and the F frame in a station.

Using vector algebra, a vector N_1 perpendicular to plane E can be defined in terms of the coordinates of the three contact points on that plane. Similarly, a vector N_2 can be defined for plane G and N_3 for plane H. The equations of the three vectors define the equations of the three planes. The solutions of the linear set of equations formed by the three plane equations are the coordinates of the intersection point of the three planes, in the Q frame. Those coordinates are also the components of the translation vector $(P_1)_Q$ in the T_{F-Q} matrix.

The equations of the F frame axes can be described in terms of the point $(P_1)_Q$ and the three vectors N_1, N_2, and N_3.

$$(X_F)_Q. \quad \frac{X-(X_{P1})_Q}{A_3} = \frac{Y-(Y_{P1})_Q}{B_3} = \frac{Z-(Z_{P1})_Q}{C_3} = K_1$$

$$(Y_F)_Q \quad \frac{X-(X_{P1})_Q}{A_1} = \frac{Y-(Y_{P1})_Q}{B_1} = \frac{Z-(Z_{P1})_Q}{C_1} = K_2$$

$$(Z_F)_Q. \quad \frac{X-(X_{P1})_Q}{A_2} = \frac{Y-(Y_{P1})_Q}{B} = \frac{Z-(Z_{P1})_Q}{C} = K_3 \quad (4)$$

where A_i, B_i, and C_i are the respective coefficients of the i, j, and k components of the ith vector. Arbitrary points E, G, and H are selected along the X_F, Y_F, and Z_F axes by assigning arbitrary, values to K_1, K_2, and K_3 in (4). Each one of those points defines two vectors: one from P_0 and one from P_1, as illustrated in Fig. 4.

The following relationship can be observed from Fig. 4:

$$(E_Q)_Q - (P_1)_Q = (E_F)_Q$$
$$(G_Q)_Q - (P_1)_Q = (G_F)_Q$$
$$(H_Q)_Q - (P_1)_Q = (H_F)_Q. \tag{5}$$

The physical meaning of subtracting the vector $(P_1)_Q$ is a parallel axes translation of the F frame to the Q frame, after which the translation vector in T_{Q-F} becomes zero. Under these conditions, and for $n = 3$, (1) becomes (6).

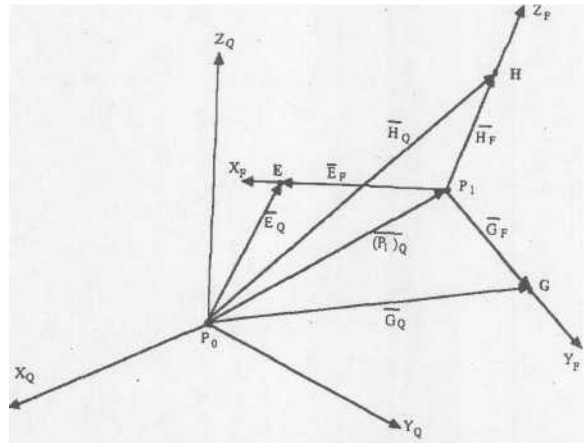

Fig. 4. The vectors E, G, and H.

$$\begin{bmatrix} \bar{E}_F & \bar{G}_F & \bar{H}_F \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \bar{R} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix}_{Q-F} \begin{bmatrix} \bar{E}_F & \bar{G}_F & \bar{H}_F \\ 1 & 1 & 1 \end{bmatrix}_Q \tag{6}$$

In (6), the vectors in the right-hand side point matrix are known from solving (5), while the vectors in the left-hand side point matrix are partially known-each vector has two zero components. Since multiplication by an orthogonal matrix does not change the length of a vector, the corresponding vectors on both sides of (6) are equal in length. After normalizing the vectors on both sides, (6) becomes (7).

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}_F = \begin{bmatrix} T_{11} & T_{12} & T_{13} & 0 \\ T_{21} & T_{22} & T_{23} & 0 \\ T_{31} & T_{32} & T_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{Q-F} * \begin{bmatrix} [\hat{E}_F] & [\hat{G}_F] & [\hat{H}_F] \\ 1 & 1 & 1 \end{bmatrix}_Q \tag{7}$$

Now there are nine linear equations to solve for the nine unknown entries of the rotation matrix. At this point, the vector $(P_1)_Q$ found earlier is combined with the rotation matrix to form the transformation matrix T_{Q-F}. It should be remembered that the vector $(P_1)_Q$ is the translation vector in the transformation matrix T_{F-Q}. The translation vector in T_{Q-F} is defined by the product of $(P_1)_Q$ and the rotation matrix in T_{F-Q}. At that point, a robot task that was specified in the theoretical frame Q, can be transformed to a task specified in the fixed frame F. Later, when the mobile robot will come to serve the station again, it will be in some other location S. The procedure that has been outlined will be repeated to derive the transformation matrix T_{F-S} and from it T_{S-F}. Then the robot task will be transformed to a task in the S frame to be executed by the robot, as illustrated in Fig. 2.

B. Vision System Application

The vision system, as the touch trigger probe, should collect data about the reference points for the calculation of the transformation matrix between the robot frame and the reference frame. When compared with the touch trigger probe application, the vision system application faces three main problems. While the touch trigger probe supplies three-dimensional information about a reference point, the vision system provides only two-dimensional information-which complicates the calculation of the transformation matrix. The second problem is that the information from the touch trigger probe (when mounted as an end effector) is in the robot frame Q or S, while the information from the vision system is in its own frame C. The third problem is the poor accuracy of the vision system when compared with that of the touch trigger probe [3], [4].

The desired depth information can be derived by either using images of the same reference object from two cameras whose relative positions are known, or by using one camera and images of the reference object from two different locations. The first option was dismissed as it requires two cameras at each station the mobile robot will serve, while the second option allows a single camera to be mounted on the mobile robot's arm. The problem of relating the information from the camera to the robot frame can be solved by mounting the camera as an end effector. The transformation from the robot frame to its end effector is continuously available from the robot controller, thus allowing to transform the information from the camera to the robot frame.

The reference object that was chosen for the vision system application was the same cube from the touch trigger probe application. The reference points were the centroids of circles that were drawn on the sides of the reference cube. The location of each reference point', in the fixed frame F, was known.

It is required to find the transformation between the mobile robot frame and the fixed frame, as expressed in (2). In the vision system application, however, the information about the reference points is available in the camera frame C and not in the robot frame. Thus (2) becomes

$$P_F = T_{C-F} * (T_{Q-C} * P_Q) \tag{8}$$

$$P_F = T_{C-F} * P_C. \tag{9}$$

In order to solve for the twelve unknown entries of the transformation matrix T_{C-F}, twelve linearly independent equations are required. The four needed reference points have to be separated between two planes on the reference cube to assure linear independence-three on one plane and one on another, as illustrated later in Fig. 6.

T_{C-F} relates the reference points in the fixed frame F (for which all the coordinates are known) to those points as the camera sees them (for which only the X_c and Y_c coordinates are directly available). Hence (9) is nonlinear due to the product of unknown on its right-hand side-T_{C-F} and the Z coordinates in P_c. The Zc coordinate (depth) for each reference point should be known before the transformation between the camera frame C and the fixed frame F can be calculated.

C. Depth Calculation

Depth information can be calculated, using one camera, if two distinct images of a known reference object are analyzed. In order to eliminate the necessity to process two images, another approach [5] was taken. The idea is that instead of using information from the reference object only, which requires two frames, information about the optical characteristics of the camera can be used instead of one of the images. Using this approach requires only one picture on-line to calculate the depth of any number of reference points larger than two. The algorithm that was developed has two parts. the calibration part and the measurement part. In the calibration part that is done off-line, once per setup, the optical characteristics of the camera (the focal center's coordinates) are determined. In the measurement part, that is done during the docking process, the information about the reference points is extracted and combined with the camera characteristics to calculate the depth of the reference points.

In the calibration part, whose setup is illustrated in Fig. 5, the camera is positioned at a known distance from a surface with four reference points. The camera's picture plane is parallel, centered, and aligned with the reference object. Relating geometrically the information from the vision system with the dimensions of the reference object allows to calculate the coordinates of the focal center in the camera's frame C.

In the measurement part, which is done on-line during the docking, the camera cannot be guaranteed to be parallel to the

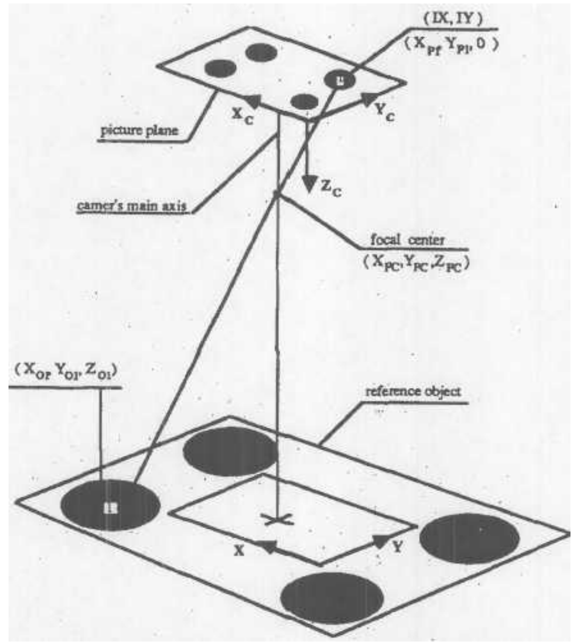

Fig. 5. The setup for the camera's characteristics experiment.

reference object (nonuniform load distribution on the mobile robot, floor conditions, etc.). Due to that, the image of the reference object is distorted and information from it cannot be used directly (see [6] for further discussion). This distortion introduced nonlinearity to the calculation: a set of four nonlinear equations have to be solved in order to calculate the four unknown Z_c coordinates of the reference points.

The calculated Z_c coordinates are substituted back to P_c in (9), which then can be solved for the desired transformation matrix T_{C-F}. Combining T_{C-F} with T_{Q-C}, that is available from the robot controller, in (8), allows to transforms the robot task from the initial robot frame Q to the fixed frame F. Later, when the mobile robot will return to serve the station again and will be in location S (see Fig. 2), the same procedure will be repeated to calculate the transformation T_{C-F}. This transformation will be used to transform the robot task from the fixed frame to the actual robot frame S.

A detailed description of the depth calculating algorithm, as well as of the transformation matrix derivation, can be found in [6].

III. PERTURBATION ANALYSIS

The main drawback of both algorithms outlined in the previous section, as well as many published ones [7]-[9] is the underlying assumption that the data, on which the calculations are based, are absolutely accurate. This, however, is not the case with an actual system where perturbed data resulting from both hardware inaccuracies and software roundoffs are a matter of fact. In this section, a method for calculating the transformation matrix between two frames while taking into account the perturbation in the data is presented.

It can be observed that in both algorithms presented in the previous section, the calculation of the transformation matrix (after all the coordinates of the reference points were calculated with their inaccuracies) is done by matrix inversion. In (9), when the coordinates of *exactly* the same reference points are given in two frames-P_c and P_F, using matrix inversion will result in calculating the rigid body transformation matrix between the two frames-T_{C-F}. This transformation matrix is. the only solution to the linear set of equations in (9). If however, the coordinates of the reference points are perturbed by more than the accuracy of the matrix inversion algorithm employed, a rigid body is no longer described from two frames. The calculated matrix will not be a rigid body transformation matrix but rather some coefficient matrix, with no geometrical meaning, that will solve the set of linear equations.

It becomes obvious that a different method for calculating the transformation matrix is needed. The method should be stable and be able to minimize the influence of errors in the data on the calculated matrix. This means that based on perturbed data, the resulting coefficient matrix should be a rigid body transformation matrix, and that variations in the data will change very little, if at all, the calculated rigid body transformation matrix-minimize S in (10).

$$\|T_{C-F} * P_C - P_F\| \le \delta. \tag{10}$$

The most familiar approach to the problem of determining the coefficients approximating a solution, based on data is the Least Squares (LS) approach [10]. The LS drawback is that the calculated T_{C-F} does not necessarily minimize the error 6. A different method for solving LS problems is the Singular Value Decomposition (SVD) [11]. The SVD is considered to be [12] "... the most reliable method for computing the coefficients of general LS problem ... and the most effective in dealing with errors in the data, roundoff errors, and linear dependence.

Briefly, using the SVD method an m x n matrix A can be decomposed into three special matrices

$$A_{mxn} = U_{mxm} * X_{mxn} * V^T \tag{11}$$

Those three special matrices allow the direct calculation of the pseudoinverse of the matrix A-A^P.

$$A^P = V * E^P * U^T. \tag{12}$$

The pseudoinverse is related to the LS problem in (10) by the fact that the T_{C-F} that minimizes 8 is defined as

$$T_{C-F} = P_F * (P_C)^P. \tag{13}$$

Because of its advantages, the SVD method was chosen as the method to solve for the transformation matrix between two frames based on position data, in both frames, of fixed reference points on a rigid body. A detailed discussion about SVD and its applications can be found in [10]-[12].

IV. EXPERIMENTAL RESULTS

Three experiments were conducted to check the validity and accuracy of the algorithm for calculating the transformation matrix based on data collected by the vision system. The first experiment was the calibration part-the calculation of the camera parameters. The second and third experiments were the measurement part: in the second experiment, the distance from the camera to the reference object was calculated while in the third experiment the transformation matrix between the camera and the reference object was calculated. Sample results from these three experiments (in millimeters) are presented in this section.

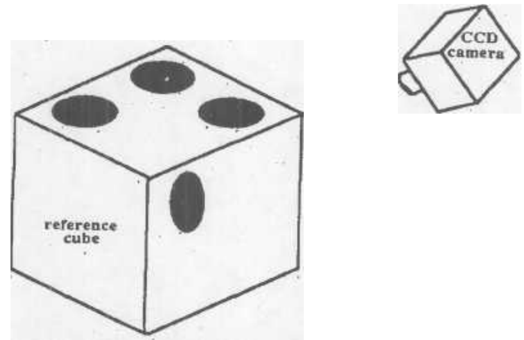

Fig. 6. The reference object and the camera in the measurement part of the 'vision system application.

The reference object that was used in the first two experiments was a sheet of paper on which four circles were drawn. The circles were 80 mm in diameter and their centroids formed a square 100 x 100 mm. For the third experiment, the reference object was the same only one of the circles was displaced, as illustrated in Fig. 6. The vision system used was the General Electric PN-2304 Optomation II.

In the first experiment, the accuracy of the calculated camera parameters *(XFC, YFC,* and Z_{FC}) was checked by using them to calculate the distance from the camera to the reference object (Z_l) and comparing it with the actual measured distance (Z_m). The experiment was conducted nine times at different camera-object distances ranging from 687 to 1067 mm. In each position, the camera's picture plane and the reference object were parallel, centered, and aligned in order to allow relating the reference object to its image for the calculation of the camera parameters. The calculation was done four times based on different combinations of reference points of which only the average is displayed in Table I. In Table I, the three right-most columns display the analysis of the data- Z_{AVG} is the average of the four calculated distances, the percent change is the difference between the measured and the averaged calculated distance, and a is the spread between the four calculated distances. The bottom two rows contain the summary of the experiment. The resulting accuracy of 0.07 percent means that under these conditions the accuracy of the camera parameters calculations is less than the size of a pixel.

In the second experiment, the depth was calculated using the camera parameters found in the first experiment. As the first experiment, this one was done nine times at different distances ranging from 727 to 1003 mm. This time, however, in each position the picture plane and the reference object were not centered neither aligned, only parallel. At each position, the calculation was done four times, based on different combinations of reference points, of which only the average is displayed in Table II whose format is the same as that of Table I. The resulting accuracy of 0.26 percent means that under

TABLE I
RESULTS OF THE CAMERA PARAMETERS AND DEPTH CALCULATION FROM THE CALIBRATION PART

					For ft calculated distance		
Run #	Z_m	X FC				% change	a
	687	5.3155	43053	26.306	687.88	0.1281	6.94
	716	5.3155	4.3025	26.280	715.80	-0.0284	8.90
3	750	53269	43053	26 333	758.01	1.0680	7.73
4	793	5.3269	4.3053	26.322	79168	-01665	900
5	835	5.3412	4.3025	26.331	83540	0.0483	9.09
6	881	5.3355	4.2796	26.363	872.65	-0.9474	1294
7	942	5.3355	4.2796	26.362	933.94	-0.8560	8.39
8	991	5.3241	4.2796	26.662	991.39	00390	11.32
	1067	5.3240	4.2796	26.393	1066.70	-0.0281	6.26
Total average		5.3283	4.2919	26.361		-00743	
Total a		0.0091	0.013	0.1132		0.5543	

TABLE 111
RESULTS OF THE TRANSFORMATION MATRIX CALCULATION FROM MEASURED DATA

Rotation by -30 ° around Z c Zm = 800 mm	
Theoretical matrix T	calculated matrix T + ST
$\begin{matrix} 0.866 & 0.5 & 0.0 & \\ -0.5 & 0.866 & 0.0 & 7 \\ 0.0 & 0.0. & 1.0 & 800.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{matrix}$	$\begin{matrix} 0.868 & 0.482 & 0.0 & -138.675 \\ -0.495 & 0.862 & 0.0 & -36.461 \\ 0.019 & -0.087 & 1.0 & 805.433 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{matrix}$
aram	Err°rmatrix ST
Z1 = 796.37 01 = -0.23 ° Z2 = 794.43 02 = 0.46 0 Z3 = 804.09 03 = -1.18 ° 04 = -0.33' average angular error = -032 0	$\begin{matrix} 0.002 & 0.018 & 0.0 & 7 \\ 0.005 & 0.004 & 0.0 & 7 \\ 0.019 & 0.087 & 0.0 & 5.433 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{matrix}$

TABLE II
RESULTS OF THE DEPTH CALCULATION FROM THE MEASUREMENT PART

		For the calculated distance		
Run #	Z-		change	
	727	728.55	0.2137	7.24
	727	729.79	0.3833	6.72
3	796	79449	-0.1897	12.29
4	796	799.15	0.3957	4.61
	881	87 .96	-1.0291	12.89
	881	880.87	-0.0148	1034
7	926	92750	0.1616	8.05
	926	916.04	-1.0752	16.11
9	1003	991.00	-1.1981	16.42
Total average			-0.2614	
Total a			0.66	

these conditions the accuracy of the depth calculation was on the average about 1.5 mm.

In the third experiment, the transformation matrix between the camera frame and the reference frame was calculated. In this experiment there are two cases: rotation around the Z axis, and any other rotation. Four reference points are necessary to solve for the transformation matrix, of which only three can be on one plane of the reference object (the top face of the cube in Fig. 6). Except for rotation around the Z axis only, all four reference points are seen by the vision system. In that case, only three points are seen and additional information (constraint) is required instead of the fourth reference point. This additional constraint is found in the problem setup. A vector in the Z direction will not change its length if rotated around the Z axis. Hence, for rotation around the Z axis (9) becomes (14).

$$\begin{bmatrix} X01 & X02 & X03 & 0 \\ Y01 & Y02 & Y03 & 0 \\ Z01 & Z02 & Z03 & 100 \\ 1 & 1 & 1 & 0 \end{bmatrix}_C = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & L_X \\ \sin\phi & \cos\phi & 0 & L_y \\ 0 & 0 & 1 & L_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}_{F-C} * \begin{bmatrix} X01 & X02 & X03 & 0 \\ Y01 & Y02 & Y03 & 0 \\ Z01 & Z02 & Z03 & 100 \\ 1 & 1 & 1 & 0 \end{bmatrix}_F \quad (14)$$

The length of the additional vector should have the same order of magnitude as the coordinates of the reference points in the. F frame for the P_F matrix to have small condition number, which is an important factor for the SVD method.

The transformation matrix was calculated eight times under different conditions, four for rotations around Z_C and four for rotations around X_C. It should be emphasized that the traditional approach-matrix inversion-did not generate the transformation matrix *in any of the trials;* the SVD approach generated a transformation matrix in each one of the trials. In Table III the result of one SVD application is presented. In that table, the axis and angle of rotation, and the distance between the camera and the reference object are stated at the top. The theoretical transformation matrix is given in the upper left corner, the calculated one is given in the upper right corner, and the difference between the two (the error matrix) is the lower right corner. In the theoretical matrix, the X and Y values of the translation vector were not measured due to the difficulty in measuring the distance between skewed lines. In the lower right corner, two sets of parameters are shown: the calculated Z coordinates (in millimeters) of the reference points based on which the transformation matrix was calculated, and the error angles calculated from the error matrix.

A. Conclusions

Several conclusions can be reached based on the experimental results.

1) The calculation of the camera parameters by the algorithm outlined is stable and accurate and thus can be used as a module in other algorithms.

2) The accuracy of the depth calculation is dependent of the constraints of the setup due to the need to solve three or four nonlinear equations. The accuracy went from 0.07 percent when the frames were parallel, aligned, and centered to 0.26 percent when they were only parallel.

3) The matrix inversion approach for calculating the transformation matrix based on measured position data is inapplicable. The SVD approach resulted in calculating a transformation matrix even for very perturbed data.

4) The accuracy of the calculated transformation matrix is dependent upon the depth calculation accuracy. The largest errors appear in the third row of the transformation matrix, which is dependent on the Z values. The angular error, which *is* not related to the Z values in the case of rotation around Zc, is very small.

V. SUMMARY

The problem of compensating for the poor docking accuracy of an AGV-mounted robot was discussed. A software solution to that problem was presented in which the basic idea is to modify the robot task according to the docking error. Two data processing algorithms, for two data collection devices, were developed: one that employs touch trigger probe mounted as a robot end effector, and one that uses vision system for that purpose. The need for a new method that should be able to calculate the transformation matrix from actually measured and perturbed data was demonstrated and such a method was presented. Finally, results from some experiments that confirm the validity of the proposed method were presented and discussed.

ACKNOWLEDGMENT

We would like to express our deepest gratitude to Prof. B. Ravani, currently with the Mechanical Engineering Department, University of California at Davis, for making us aware of the SVD and his thoughtful comments throughout the project; and to thank Prof. Hershkovitz, currently with the Mathematics Department, Technion-Israel Institute of Technology, Haifa, Israel, for his suggestions.

REFERENCES

[1] D. F. Rogers and J. A. 'Adams, *Mathematical Elements for Computer Graphics.* New York, NY: McGraw-Hill, 1978.

[2] T. R. Kane, P. W. Likins, and D. A. Levinson, *Space Craft Dynamics.* New York, NY: McGraw-Hill, 1983.

[3] N. A. Duffie and G. W. Halldin,. "The use of tactile sensing and computer graphics in the automation of mold production for plastics processing," Progress Rep. 1 to NSF, Dec. 1983.

[4] J. W. Lunden, "Optomation II-Microprocessor Based Vision Processing Instrumentation for Automatic Inspection," *in IEEE Ind. Electron. Conir. Instrum. Soc. Meet. Rec.*, pp. 445-449, 1981.

[5] F. Ozguner and S. J. Tsai, "Design and implementation of a binocular-vision system of a multi-legged walking robot," *IEEE Trans. Ind. Electron., vol.* IE-32, no. 1, pp. 26-31, Feb. 1985.

[6] K. Mandel, "On-line correction of mobile robot docking errors," M.S. thesis, University of Wisconsin-Madison, Aug. 1985.

[7] O. Bottema and B. Roth, *Theoretical Kinematics.* Amsterdam, The Netherlands: North-Holland, 1979.

[8] C. Sub and C. Radcliffe, *Kinematics and Mechanisms Design.* New York, NY:'Wi1ey, 1978.

[9] A. J. Laub and G. R. Shiflett, "A linear algebra approach to the analysis of rigid body displacement from initial and final position data," *ASME J. Appl. Mech., vol.* 104, Mar. 1982.

[10] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

[11] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math., vol.* 14, pp. 403-420, 1970.

[12] G. E. Forsyth and C. B. Moler, *Computer Methods for Mathematical Computations.* Englewood Cliffs, NJ: Prentice-Hall, 1977.

Kostia Mandel (M'86) received the B.S. and M.S. degrees in mechanical engineering from the University of Wisconsin-Madison, in 1983 and 1985, respectively.

He is currently a Member of Technical Stuff in the Advanced Design and Production Methodology group at AT&T Bell Laboratories in Columbus, OH, working on developing methods and tools for description and control of dynamically reconfigurable processes. His work includes analysis of generic process characteristics, parameterization of those characteristics, representation of processes, application of AI techniques for capturing process knowledge, and utilization of this process knowledge for process design and control.

Mr. Mandel is a member of HE, ASME, CASA/SME, and Tau Beta Pi.

Neil A. Duffie received the B.S. degree in computer science, the M.S. degree in engineering, and the Ph.D. degree in mechanical engineering from the University of Wisconsin-Madison in 1972, 1974, and 1980, respectively.

He currently is an Associate Professor of Mechanical Engineering at the University of Wisconsin-Madison where he is engaged in teaching and research in the areas of manufacturing systems, precision engineering, computer control systems, artificial intelligence, and robotics. He is co-author of a textbook entitled *Computer Control of Machines and Processes.* He is also Associate Director of the Manufacturing Systems Engineering Program at the University of Wisconsin-Madison, Co-Director of the Wisconsin Manufacturing Automation and Robotics Research Consortium, and Director of Automation and Robotics Research for the Wisconsin Center for Space Automation and Robotics. He is an industrial consultant in the areas of manufacturing systems, machine control, automated inspection, and robotics.