Abstract—Contact formations have proven useful for programming robots by demonstration for operations involving contact. These techniques require real time recognition of contact formations. Single ended contact formation (SECF) classifiers using only forces/torques measured at the wrist of the robot have been shown to be quite effective for this purpose. To function properly, however, previous SECF classifiers have required a sizable training set and a constant pose between the force/torque sensor and the manipulated object. Thus, if an object is regrasped and the pose changes, one expects to have to repeat the creation of the training set. We discuss the impact sensor-object pose changes have on two successful classifiers. Experimental data shows that they perform poorly when sensor-object pose changes. We derive, experimentally verify and discuss a method to regain the performance of both classifiers while minimizing the retraining necessary.

Index Terms—Contact formation, force sensing, kinematics.

I. INTRODUCTION

When humans perform assembly tasks, they typically use vision until the grasped object contacts the environment. After contact is made, the haptic sense often becomes more important. For example, it is the haptic sense that allows one to wiggle a key into a tight keyhole. A human implicitly uses the concept of contact formations [1] to move the tip of a key from the flat lock plate into the entrance of the keyhole and into the lock cylinder. In an analogous manner, if the robot has a means of identifying contact formations [2] and an ability to reason about these contacts, it has the potential of being programmed to achieve contact formations instead of a detailed enumeration of positions and orientations that frequently cannot be achieved with sufficient accuracy. Considerable work has been performed to determine techniques based upon contact information for directing robot motion, e.g., [3]–[6]. A critical part of this strategy is the correct identification of the contact formations.

There are a number of methods for identifying contact formations. A detailed review of the literature can be found in [6] and [7]. The work presented in this paper was specifically designed to complement the recent work by Skubic and Volz [6], [7]; hence the overview of classifiers will include only their work. Skubic and Volz produced two effective techniques for recognizing the single-ended contact formation (SECF) information needed for manipulation of objects in contact, one based upon use of fuzzy set theory and the other based upon use of neural networks. Both require sizable training sets of experimental data. While they work well, they are limited to situations in which the object being manipulated does not change its pose with respect to the force/torque sensor that is used for obtaining the training data. However, in real situations, there can be variations in this pose as different objects are grasped.

The work presented in this paper extends the two proven SECF classifiers to accommodate changes in the pose between the force/torque sensor mounted in the robot’s wrist and the manipulated object in the gripper. Although the method was tested with these specific classifiers, it should be applicable to other classifiers. Its utility with other classifiers increases with the amount of data required for initially training those classifiers.

First, the paper briefly describes techniques for recognizing contact formations, and shows that pose changes cause problems with the two SECF classifiers. Then, a method that uses a minimal set of measurements to recover the performance of both classifiers without having to repeat generation of the training sets is described. Finally, experiments validating the method are described.

II. RELATED RESEARCH

Several methods of contact formation recognition have been tested. Most of these methods can be broadly classified into one of two main groups: position or geometry based methods and force based methods.

A. Position/Geometry Based Methods

We refer to position/geometry based methods of identifying contact formations as those that use detailed geometric models of the workpieces along with force/moment signals. Desai and Volz [2] identified the contact formation by first formulating a hypothesis and then verifying that it was consistent with measured force signals. They used static equilibrium equations and solved for the contact forces and moments. In cases where the contact forces could not be determined, or ambiguity existed, active force sensing was used. Hirai and Asada [5] derived contact classifiers from a geometric model using polyhedral convex cones. The ranges of possible forces and displacements, which can be measured at each contact
formation, were represented as the union of polyhedral convex cones (PCC). The face vectors of the PCC’s were used to determine the contact formation from the force signals.

McCarragher and Asada [3] used an analysis of rigid body dynamics to identify the contact formation between objects and determine a motion to achieve a desired contact. Constraints were added to the equations of motion with LaGrange multipliers and a velocity constraint matrix. Qualitative templates were then precalculated as the sequence of qualitative states that represent the type of motion expected to achieve the desired contact state.

Position-based approaches require detailed geometric information about the position, size and shape of all the objects involved in a contact formation. This is impractical in many settings and is hard to do in real-time. It is also difficult for these methods to include the effects of friction and sensor noise with sufficient accuracy.

To address problems with sensor uncertainties and friction, Farahat, Graves, and Trinkle [8] solved a linear program to determine the feasibility of a contact formation. Good results were achieved, but the performance rapidly decreases with an increase in geometric uncertainty, and cannot readily be done in real-time.

B. Force Based Approaches

Methods using force signals only have also been proposed, and can be grouped into two categories, static and dynamic approaches.

Dynamic force based approaches try to capture the dynamic effects caused by transition from one contact state to another. For example, Hovland and McCarragher [4] used fast Fourier transforms to capture the dynamic nature of the contact change. To implement this method in real-time involves high computational costs, however. Samples must be collected over a certain time period, which makes real-time identification difficult.

Static force based approaches use prior training to recognize the contact state from force and/or moment signals. These methods do not require multiple force/moment signals at run time to recognize a contact state. Hara and Yokogawa [9] used fuzzy sets to model contact formations. However, only a small number of contact states for one case were considered. No attempt was made to provide a generalized methodology for applying the approach to other workpieces or contact tasks.

Skubic [6] addressed the above problems and formulated a more general contact recognition scheme. Both fuzzy logic classifiers and neural networks were used to recognize the contact states. This method provides a way to deal with the uncertainties involved with friction and nonlinear sensors. However, as our experiments show, this method cannot deal with a change in object pose from that used during training.

III. CONTACT FORMATIONS AND THEIR RECOGNITION

In this paper, our principal goal is to derive a method for recovering the performance of SECF classifiers when the pose of the manipulated object changes relative to the sensor after training is complete. Sections III-A and III-B review the concept of contact formations and the fuzzy logic and neural network classifiers used in the experiments. These sections do not contain any original work. The work described in Section III-C to the end represents an original contribution. Beginning in Section III-C, we develop the technique for recognizing SECF’s for different grasp positions of the manipulated object.

A. Contact Formations

A contact formation is the set of configurations with identical, elemental contacts between pairs of topological elements [1], [2]. It provides a qualitative description of how two or more objects make contact with each other (e.g., edge 1 of the grasped object touches side B of the environment). Fig. 1 shows three example contact formations. In contrast, a single-ended contact formation (SECF) [6] provides a one-sided description of how a grasped object touches its environment (e.g., edge 1 of the grasped object touches any side in the environment). A SECF does not attempt to distinguish between contact points in the environment. Fig. 2 shows an example of two contact formations in which an edge of the moveable object is touching different sides of a fixed object in the environment. Because both of the contacts involve the same edge of the movable object, they are examples of the same SECF. Skubic [6], [10] also shows how this information can be used to accomplish assembly tasks.

B. Review of Fuzzy and Neural Net SECF Classifiers

The force-based methods used by Skubic [6], [7] to identify the SECF’s utilize a robot with a force/torque sensor mounted on its wrist. The grasped object is held between the sensor and the environment. During training, the object is grasped once and placed a substantial number of times into known SECF’s. During each SECF, force/torque readings are recorded, and
TABLE I

<table>
<thead>
<tr>
<th>Component</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0.221</td>
<td>0.300</td>
<td>-0.023</td>
</tr>
<tr>
<td>Y</td>
<td>0.084</td>
<td>0.077</td>
<td>0.060</td>
</tr>
<tr>
<td>Z</td>
<td>0.022</td>
<td>0.021</td>
<td>0.027</td>
</tr>
</tbody>
</table>

This data are used for training a classifier. During operation, the robot moves the workpiece until a change in the SECF occurs, which is identified from the measured force signals using one of the classifiers. Table I gives typical force and moment values from an edge SECF. Because the object can be placed into the SECF with different orientations, the force and moment readings can vary. The table gives an indication of the variability. Fig. 3 shows the individual X component of force versus the Y component. The figure gives an overview of the variability.

The fuzzy logic classifier is trained using supervised learning; membership functions are generated automatically using the means and standard deviations of the training sets. Force and moment vectors are first normalized. Then, for each sensor component \( j = 1 \cdots 6 \) (a force or moment component measured in the sensor frame) of each SECF \( i \), the mean, \( \mu_{ij} \), and standard deviation, \( \sigma_{ij} \), are calculated. These are used as parameters to generate the membership functions, \( \pi_{ij}(r_j) \), as shown below; here \( r_j \) refers to a measured force or moment component.

\[
\pi_{ij}(r_j) = \begin{cases} 
1 - e^{(-\sigma_{ij}/\mu_{ij} - r_j)^2}, & \text{if } r_j \neq \mu_{ij} \\
1, & \text{if } r_j = \mu_{ij}.
\end{cases}
\]

These individual measures are combined into a single factor to determine if the measurements belong to a specific class. The details of this conjunction are unimportant for the present work. A complete discussion can be found in [6], [7], and [11]. Essentially the conjunction generates a confidence level for each class. The class with the highest confidence wins.

The neural net classifier has six inputs, which are the normalized components of the force and moment vectors. The number of output nodes is the number of SECF classes to be identified. There is one hidden layer. The number of hidden nodes is dependent on the coarse geometric shape of the object. There is one hidden node for each vertex on the object that is part of an included SECF. The output of the hidden layer indicates at which vertex points contact has been established.

The output of the final layer then maps this information to the corresponding SECF class. In this way, the network structure has physical meaning. The values at the output nodes may be interpreted as confidence factors for each class. The highest confidence factor represents the identified class.

C. Overview of the Technique

Since the amount of data needed to train either the neural network or the fuzzy classifier is large, one does not wish to do it frequently. If the pose of the held object changes relative to the force sensor, the forces and moments measured for most SECF’s change. If the pose change is sufficiently large, the classifiers will be unable to properly identify the SECF, as will be demonstrated experimentally later. Our objective is to collect as little information as necessary to allow the classifiers to function after a change in object pose.

We assume the classifiers have been trained and that the object pose relative to the sensor has changed such that the classifiers no longer work properly. We begin by driving any face of the object into face–face contact with any portion of the environment. We must know which face on the object is making contact but we do not need to know where it contacts the environment. During this contact we measure the force and moment. If the sensor has significant noise, we can take several sensor readings and average them. We need not break contact between readings; therefore, this averaging process is expected to be quick and simple. We only need the direction of the force and moment signals; therefore, we need to press hard enough to get a reliable reading, yet need not worry about regulating the contact in any way.

After obtaining reliable force and moment measurements, we break contact and form another SECF with a different face of the object. The normal of this second face must not be collinear with the normal to the first. The second contact can be made on any surface of the environment; it need not be the same environmental surface as the first. We again measure force and moment (or we average a set of measurements if necessary) and normalize these into unit vectors.

By comparing the two force and two moment readings to similar values obtained during training, we compute the change in object pose. Having the change in object pose, we can use the previously trained classifiers to identify all SECF’s whether they are face, edge, or vertex. Transforming the measured force and moment back into an “as trained” configuration and using these values in the classifiers does this. We use face SECF’s for comparison because they have been shown in our experiments to have the least variability. Experimental results demonstrate how well the method works.

D. Derivation of Transformations

Throughout this derivation, superscripts denote the grasp. The superscript \( i \) will denote the grasp during original training, and the superscript \( j \) will denote the grasp after a change in gripping position. Let \( s \) be a coordinate system located at the force sensor such that forces and moments measured by the sensor are relative to the \( s \) frame. Similarly, let \( o \) be an object coordinate system which is fixed in the held object. All faces
and edges are invariant when measured in the $o$ frame. Let the transformation from $s$ to $o$ during original training be given by the $4 \times 4$ matrix $sT_o^s$. The $3 \times 3$ matrix formed from the upper left portion of $sT_o^s$ is a pure rotation and is denoted $sR_o^s$. The transformation $sR_o^s$ is unknown since we have no information about how the object is grasped during training.

During the training of the classifiers, some face, which we will call face A, was brought into SECF with a surface of the environment. When the SECF is achieved there will be a force generated on the face of the object at location $f$. The position of $f$ from the object frame origin, measured in the object frame is given by $sF_i^o$. The 3 \times 1 moment is $\omega M_i = \sigma_j sR_{s-o}^s \times sF_i^s$. The sensor will measure values in the sensor frame. These values can be expressed in terms of values in the object frame as [12]

\begin{equation}
\begin{bmatrix}
sF_i^s \\
sM_i^s \\
sG_i^s \\
sN_i^s \\
sH_i^s
\end{bmatrix} =
\begin{bmatrix}
sR_0^s \\
\sigma_j sR_{s-o}^s \times sR_0^s \\
sR_0^s \\
\sigma_j sR_{s-o}^s \times sR_0^s \\
sR_0^s
\end{bmatrix}
\begin{bmatrix}
sF_i^o \\
sM_i^o \\
sG_i^o \\
sN_i^o \\
sH_i^o
\end{bmatrix},
\end{equation}

The term $\sigma_j sR_{s-o}^s$ is the position of the object frame’s origin from the sensor origin expressed in the sensor frame. In other words it is the first three rows of the fourth column of $sT_o^s$. The cross product is a matrix operator given by the following [12]:

\begin{equation}
r \times =
\begin{bmatrix}
0 & -r_z & r_y \\
r_z & 0 & -r_x \\
-r_y & r_x & 0
\end{bmatrix}.
\end{equation}

Notice that since $sF_i^s$ and $sF_i^o$ differ by only a rotation, they have the same magnitude. Hence we can determine the magnitude of $sF_i^o$ by computing the magnitude of the measured force $sF_i^s$. Let $\alpha_i$ be the magnitude of the force which is computed from the measurement $sF_i^s$.

We assume that the training process also resulted in a different SECF with another face B of the object. This generates another force and moment. The force will be called $sG_i^o$ and is located at a point $g$. The moment relative to the object is $sN_i^o = \sigma_j sR_{s-o}^s \times sG_i^s$. We will require the forces to be nonparallel, which can be achieved provided the normals to faces A and B are nonparallel. The values that the sensor measures can be related to these values as follows:

\begin{equation}
\begin{bmatrix}
sG_i^o \\
sN_i^o \\
sH_i^o
\end{bmatrix} =
\begin{bmatrix}
sR_0^s \\
\sigma_j sR_{s-o}^s \times sR_0^s \\
sR_0^s
\end{bmatrix}
\begin{bmatrix}
sG_i^s \\
sH_i^s
\end{bmatrix}.
\end{equation}

The magnitude of this measured force can be computed and is denoted $\beta_i$. Finally, we can compute a third force

\begin{equation}
sH_i^o = \frac{sF_i^o}{\alpha_i} \times \frac{sG_i^o}{\beta_i}
\end{equation}

which has known magnitude and direction. We can therefore define the following force in the object frame:

\begin{equation}
sH_i^o = sR_{s-o}^s \cdot sH_i^o.
\end{equation}

We cannot compute $sH_i^o$ since $sR_{s-o}^s$ is unknown, but it is defined.

Suppose the robot sets the object down then picks it up again such that the pose of the object relative to the sensor has changed. This time let the grasp be denoted by the superscript $j$. We drive face A of the object into face SECF with any surface of the environment and measure the force and moment. After this we break the contact and drive face B of the object into SECF with any surface of the environment and measure the force and moment. We will deal with moment later. We can define and compute terms for grip $j$ that are similar to grip $i$. For example, we can write

\begin{equation}
\begin{bmatrix}
sF_j^o \\
sM_j^o \\
sG_j^o \\
sN_j^o \\
sH_j^o
\end{bmatrix} =
\begin{bmatrix}
sR_0^o \\
\sigma_j sR_{s-o}^o \times sR_0^o \\
sR_0^o \\
\sigma_j sR_{s-o}^o \times sR_0^o \\
sR_0^o
\end{bmatrix}
\begin{bmatrix}
sF_j^o \\
sM_j^o \\
sG_j^o \\
sN_j^o \\
sH_j^o
\end{bmatrix},
\end{equation}

\begin{equation}
\begin{bmatrix}
sG_j^o \\
sN_j^o \\
sH_j^o
\end{bmatrix} =
\begin{bmatrix}
sR_0^o \\
\sigma_j sR_{s-o}^o \times sR_0^o \\
sR_0^o
\end{bmatrix}
\begin{bmatrix}
sG_j^o \\
sN_j^o \\
sH_j^o
\end{bmatrix}.
\end{equation}

We can also compute force magnitudes $\alpha_j$ and $\beta_j$.

Now arrange the first three rows from (1)–(6) to find the following identities:

\begin{equation}
\begin{bmatrix}
sF_i^o \\
sG_i^o \\
sH_i^o
\end{bmatrix} = (sR_0^o)^{-1}
\begin{bmatrix}
sF_i^o \\
sG_i^o \\
sH_i^o
\end{bmatrix},
\end{equation}

\begin{equation}
\begin{bmatrix}
sF_j^o \\
sG_j^o \\
sH_j^o
\end{bmatrix} = (sR_0^o)^{-1}
\begin{bmatrix}
sF_j^o \\
sG_j^o \\
sH_j^o
\end{bmatrix}.
\end{equation}

Because the normalized forces are relative to the object coordinate frame they are equal from one grasp to the next, except for minor differences caused by friction force, i.e., by assuming they are equal we can write the $[sF_i^o/\alpha_i, sG_i^o/\beta_i, sH_i^o]$ as:

\begin{equation}
\begin{bmatrix}
sF_i^o \\
sG_i^o \\
sH_i^o
\end{bmatrix} = (sR_0^o)^{-1}
\begin{bmatrix}
sF_i^o \\
sG_i^o \\
sH_i^o
\end{bmatrix}.
\end{equation}

The $3 \times 3$ matrices comprised of forces are nonsingular because the first two columns are nonparallel unit vectors and the third column is the cross product of the first two. We can solve this equation for the quantity $(sR_0^o)^{-1}$ by post multiplying by the inverse of $[sF_i^o/\alpha_i, sG_i^o/\beta_i, sH_i^o]$. Once the rotational transform $(sR_0^o)^{-1}$ is known, we can convert any force measured during the $j$ grip into an “equivalent” force in the $i$ grip as

\begin{equation}
\frac{sF_i^o}{\alpha_i} = (sR_0^o)^{-1}sF_j^o/\alpha_j.
\end{equation}

Rather than computing $sH_i^j$ via the cross product, it might be possible to reduce algorithm uncertainty by measuring a third force and moment and computing a “best fit” rotation matrix. We chose not to do this since one of the objectives was to reduce the amount of data collection required for implementing the method.
Now focus on the moment terms. First consider the lower three rows in (1) which are rewritten in the following:

\[ sM_i = s^i\delta_{s=0} \times R_0^i \omega F_i + sR_0^i M^i \]

\[ = s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i} + sR_0^i (\delta^i_{s=0} \times \omega F_i) \]

\[ \frac{sM_i}{\alpha_i} - s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i} = sR_0^i \left( \frac{\partial \omega F_i}{\partial \alpha_i} \times \omega F_i \right) \]

\[ (sR_0^i)^{-1} \left( \frac{\partial \omega F_i}{\partial \alpha_i} \times s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i} \right) = \omega^i_{s=0} \times \frac{\partial F_i}{\partial \alpha_i}. \]

If we manipulate the other expressions for moment, we can write three more expressions similar to the above. The results follow:

\[ (sR_0^i)^{-1} \left( \frac{\partial N_j}{\partial \beta_j} - s^j\delta_{s=0} \times sR_0^j \frac{\partial G_j}{\partial \beta_j} \right) = \omega^j_{s=0} \times \frac{\partial G_j}{\partial \beta_j} \]

\[ (sR_0^j)^{-1} \left( \frac{\partial M_j}{\partial \alpha_j} - s^j\delta_{s=0} \times sR_0^j \frac{\partial G_j}{\partial \beta_j} \right) = \omega^j_{s=0} \times \frac{\partial G_j}{\partial \beta_j} \]

\[ (sR_0^j)^{-1} \left( \frac{\partial N_j}{\partial \beta_j} - s^j\delta_{s=0} \times sR_0^j \frac{\partial G_j}{\partial \beta_j} \right) = \omega^j_{s=0} \times \frac{\partial G_j}{\partial \beta_j}. \]

If the objects making contact are rigid and the friction forces are minimal, the pressure distribution between the objects relative to the object coordinate system will be approximately constant from grip to grip. Therefore

\[ \frac{\partial^i_{s=0} \times \omega F_i}{\alpha_i} \approx \frac{\partial^j_{s=0} \times \omega F_j}{\alpha_j} \]

and

\[ \frac{\partial^i_{s=0} \times \omega G_i}{\beta_i} \approx \frac{\partial^j_{s=0} \times \omega G_j}{\beta_j} \]

hence it follows:

\[ (sR_0^i)^{-1} \left( \frac{sM_i}{\alpha_i} - s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i} \right) \]

\[ \approx (sR_0^i)^{-1} \left( \frac{sM_j}{\alpha_j} - s^j\delta_{s=0} \times sR_0^j \frac{\partial G_j}{\partial \beta_j} \right) \]

and

\[ (sR_0^i)^{-1} \left( \frac{sN_i}{\beta_i} - s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i} \right) \]

\[ \approx (sR_0^j)^{-1} \left( \frac{sN_j}{\beta_j} - s^j\delta_{s=0} \times sR_0^j \frac{\partial G_j}{\partial \beta_j} \right). \]

Focus on the first of these

\[ \frac{sM_i}{\alpha_i} - s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i} \]

\[ \approx [(sR_0^i)(sR_0^i)^{-1}] \frac{sM_j}{\alpha_j} + \left[ (sR_0^i)(sR_0^i)^{-1} \right] \left( \frac{s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i}}{\alpha_i} \right). \]

Then

\[ \frac{sM_i}{\alpha_i} \approx [(sR_0^i)(sR_0^i)^{-1}] \frac{sM_j}{\alpha_j} + \left[ (sR_0^i)(sR_0^i)^{-1} \right] \left( \frac{s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i}}{\alpha_i} \right). \]

Remember that $\frac{\partial F_i}{\partial \alpha_i} \approx \frac{\partial F_j}{\partial \alpha_j}$. Hence

\[ \frac{sM_i}{\alpha_i} \approx [(sR_0^i)(sR_0^i)^{-1}] \frac{sM_j}{\alpha_j} + \left[ (sR_0^i)(sR_0^i)^{-1} \right] \left( \frac{s^i\delta_{s=0} \times sR_0^i \frac{\partial F_i}{\partial \alpha_i}}{\alpha_i} \right) \]

\[ - \left[ (sR_0^j)(sR_0^j)^{-1} \right] \left( \frac{s^j\delta_{s=0} \times sR_0^j \frac{\partial F_j}{\partial \alpha_j}}{\alpha_j} \right) \]

\[ \frac{sM_i}{\alpha_i} \approx [(sR_0^i)(sR_0^i)^{-1}] \frac{sM_j}{\alpha_j} + \left[ (sR_0^i)(sR_0^i)^{-1} \right] \left( \frac{s^i\delta_{s=0} \times sR_0^i \frac{\partial F_i}{\partial \alpha_i}}{\alpha_i} \right) \]

\[ - \left[ (sR_0^j)(sR_0^j)^{-1} \right] \left( \frac{s^j\delta_{s=0} \times sR_0^j \frac{\partial F_j}{\partial \alpha_j}}{\alpha_j} \right). \]

The following identity is useful and proven in the Appendix:

\[ (Rw) \times (Rw) = R(\omega \times w). \]

Using this, the following ensues from (10):

\[ \frac{sM_i}{\alpha_i} \approx [(sR_0^i)(sR_0^i)^{-1}] \frac{sM_j}{\alpha_j} + \left[ (sR_0^i)(sR_0^i)^{-1} \right] \left( \frac{s^i\delta_{s=0} \times sR_0^i \frac{\partial F_i}{\partial \alpha_i}}{\alpha_i} \right) \]

\[ - \left[ (sR_0^j)(sR_0^j)^{-1} \right] \left( \frac{s^j\delta_{s=0} \times sR_0^j \frac{\partial F_j}{\partial \alpha_j}}{\alpha_j} \right) \]

\[ (sR_0^i)^{-1} \left( \frac{sN_i}{\beta_i} - s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i} \right) \]

\[ \approx (sR_0^i)^{-1} \left( \frac{sN_j}{\beta_j} - s^j\delta_{s=0} \times sR_0^j \frac{\partial G_j}{\partial \beta_j} \right). \]

We can manipulate (9) in a similar manner using the fact that $\frac{\partial G_i}{\partial \beta_i} \approx \frac{\partial G_j}{\partial \beta_j}$ to find

\[ \frac{sN_i}{\beta_i} - \left[ (sR_0^i)(sR_0^i)^{-1} \right] \frac{sN_j}{\beta_j} \]

\[ \approx \left[ (sR_0^i)(sR_0^i)^{-1} \right] \left( \frac{\partial F_i}{\partial \alpha_i} \right) \frac{s^i\delta_{s=0} \times sR_0^i \frac{\partial G_i}{\partial \beta_i}}{\alpha_i}. \]

Note that the left-hand side vectors in the last two equations are known since (7) determines $(sR_0^i)(sR_0^i)^{-1}$, and all the $\alpha$ and $\beta$ are known.

The objective now is to determine the vector $[(sR_0^i)(sR_0^i)^{-1}](sR_0^j)(sR_0^j)^{-1}]$. Focus first on (12) and define as a matter of convenience the following 3 × 1 vectors:

\[ \frac{sM_i}{\alpha_i} - \left[ (sR_0^i)(sR_0^i)^{-1} \right] \frac{sM_j}{\alpha_j} = M \]

\[ \frac{s^i\delta_{s=0} \times sR_0^i \frac{\partial F_i}{\partial \alpha_i}}{\alpha_i} = r \]

Hence $M = r \times F$. Note that $M$ and $F$ are known, $F$ is a unit vector and $M$ is perpendicular to $r$ and $F$. Now $r$
can be expressed as a component perpendicular to \( F \) and a component parallel to \( F \) which we write as \( \eta = r_\perp + r_\parallel F \). Note that \( r_\perp \) is a vector but \( \eta \) is a scalar. If \( M = 0 \), then \( \eta || F \) and \( r_\perp = 0 \). Assuming \( M \neq 0 \), then \( M \times F = (r_\perp \times F) \times F = (r_\perp \times F) = r_\perp \). In either case the only thing left to be determined is \( \eta \).

Now consider (13) and define terms such that it can be written in simplified form \( N = (r_\perp + \eta F) \times G \). Note that \( N \) and \( G \) are known, \( G \) is a unit vector and \( N \) is perpendicular to \( (r_\perp + \eta F) \) and \( G \). We can write this as \( N - G = \eta F \times G = \eta H \) where the only thing unknown is \( \eta \); therefore

\[ \eta = \frac{||N - \eta \times G||}{||H||}. \]

After determining \( s R_1^1 \left( \sigma_{x-z-\alpha}^i - \sigma_{x-z-\alpha}^j \right) \times s R_1^1 \left( \sigma_{x-z-\alpha}^i \right)^{-1} \), the SECF’s during grip \( j \) can be determined by transforming the measured forces and moments into equivalent grip \( i \) values using (8) and (11). The method is as follows.

1) Grip the object with grip \( i \) and train the classifiers.
   a) Choose two convenient faces of the object that are as close to perpendicular to each other as possible.
   b) One at a time, bring each chosen face into contact with the environment and record the forces and moments produced. Call these measurements \( s F_i^1, s G_i^1, s M_i^1 \), and \( s N_i^1 \).
   c) Compute the magnitudes of the force signals defining \( a_i \) and \( b_i \).

2) After gripping the object with grip \( j \) bring the same two faces into contact with the environment thus finding the quantities: \( s F_j^1, s G_j^1, s M_j^1 \), and \( s N_j^1 \) then compute \( a_j \) and \( b_j \).

3) Use the data from steps 1 and 2 to find \( \left( s R_1^1 \left( \sigma_{x-z-\alpha}^i \right) \right) \times \left( s R_1^1 \left( \sigma_{x-z-\alpha}^j \right)^{-1} \right) \) and \( \left( s R_1^1 \left( \sigma_{x-z-\alpha}^i \right) \right) \times \left( s R_1^1 \left( \sigma_{x-z-\alpha}^j \right)^{-1} \right) \), using the techniques presented above.

4) When a SECF for grip \( j \) must be identified, use one of the following techniques:
   a) Convert the classifier to operate directly with grip \( j \) data as follows:
      i) Convert all forces and moments of grip \( i \) (the training grip) into forces and moments of grip \( j \) using (8) and (11) with the data from step 3.
      ii) Use data from the sensor at grip \( i \) into the transformed classifier.
   b) Convert the sensor data into grip \( i \) and use the original classifier as follows:
      i) Whenever a sensor reading is taken convert force and moment from grip \( j \) into grip \( i \) using (8) and (11) with the data from step 3.
      ii) Use the transformed data in the original classifier.

The next section demonstrates results using both methods.

IV. EXPERIMENTAL RESULTS

A. Effect of Pose Changes on Original Fuzzy Logic Method

Tables II and III list the performance of the fuzzy classifier [6], [7] in terms of percentage recognition of the sample data for four edges and four faces\(^1\) before and after an orientation and position change of a square block workpiece. On analyzing the performance results presented in Tables II and III we can see that the performance degradation for the face SECF’s is more than the edge SECF’s. This is expected since for small orientation changes, some of the new force signals from the edges still lie in the original cone because the spread of the training data cone is large. The force signal spread of the faces are much less than edges so the probability that the new rotated force signals fall in the original cone is much less compared to that of the edges; hence, there is less probability that they will be correctly identified. Since pose change is a nonlinear phenomenon, the performance of the original classifiers after a pose change is dependent on the magnitude of pose change. Because the algorithm of this paper parameterizes rigid body pose change, it works well even for large pose changes. As a result, the level of improvement provided by the new method will vary. If the pose change is small, the original classifiers will generally perform well and the level of improvement will not be as significant. If the pose change is large enough the original classifier may fail completely; hence, the improvement provided by the new method is very significant. Data is shown where the original classifier works modestly and where they fail completely. In both cases, the new technique works exceptionally well.

\(^1\) The algorithm was applied to many more SECF’s than what are shown in the results. All results were typical of what is shown here.

\(^2\) In this case, more cases than the 10 reported in the results were tested. Due to space restrictions, only typical results are shown.
TABLE V
PERFORMANCE OF A FUZZY LOGIC STATE CLASSIFIER BEFORE AND
AFTER A POSE CHANGE FOR FIVE-FACE SECF’S OF A PENTAGON

<table>
<thead>
<tr>
<th>Performance/Face</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before pose change</td>
<td>90.3</td>
<td>91.7</td>
<td>89.4</td>
<td>88.9</td>
<td>90.5</td>
</tr>
<tr>
<td>After pose change</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE VI
PERFORMANCE OF A FUZZY LOGIC STATE CLASSIFIER BEFORE AND
AFTER ADAPTATION (USING THE METHOD OF THIS PAPER) FOR FOUR-EDGE SECF’S OF A SQUARE BLOCK

<table>
<thead>
<tr>
<th>Edge</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perf. before adaptation</td>
<td>42.0</td>
<td>36.5</td>
<td>32.7</td>
<td>38.3</td>
</tr>
<tr>
<td>Perf. after adaptation</td>
<td>94.0</td>
<td>92.4</td>
<td>89.2</td>
<td>90.8</td>
</tr>
<tr>
<td>Improvement</td>
<td>+52</td>
<td>+56</td>
<td>+56</td>
<td>+52</td>
</tr>
</tbody>
</table>

TABLE VII
PERFORMANCE OF A FUZZY LOGIC STATE CLASSIFIER BEFORE AND
AFTER ADAPTATION (USING THE METHOD OF THIS PAPER) FOR FOUR-FACE SECF’S OF A SQUARE BLOCK

<table>
<thead>
<tr>
<th>Face</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perf. before adaptation</td>
<td>0</td>
<td>5.1</td>
<td>0</td>
<td>4.3</td>
</tr>
<tr>
<td>Perf. after adaptation</td>
<td>92.1</td>
<td>90.9</td>
<td>93.4</td>
<td>94.2</td>
</tr>
<tr>
<td>Improvement</td>
<td>+92</td>
<td>+86</td>
<td>+93</td>
<td>+90</td>
</tr>
</tbody>
</table>

TABLE VIII
AVERAGE AMOUNT OF DATA FOR A SQUARE BLOCK

<table>
<thead>
<tr>
<th>Face Data</th>
<th>Edge Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data To Train</td>
<td>@1331 per face</td>
</tr>
<tr>
<td>Data To Adapt</td>
<td>2 faces @</td>
</tr>
<tr>
<td>Savings</td>
<td>Up To 19,643</td>
</tr>
</tbody>
</table>

even a small change in the orientation or position can cause the new signals to lie outside the original sets.

B. Performance after Using the Adaptation Methodology for the Fuzzy Logic Classifier

Tables VI and VII list the percentage recognition and improvement percentage of the fuzzy state classifier for the square block workpiece after the adaptation. Table VIII gives the average amount of data used in the original training of the classifiers and the amount of data taken to perform the adaptation for the square block. In the original training, data is collected on each SECF that is to be identified. This means that training data may need to be collected for up to five faces (the top is not used because that is where the gripper attaches) and 12 edges. The Tables IX and X list the percentage recognition and improvement percentage of the fuzzy state classifier for the pentagon block workpiece after the adaptation. The pentagon block workpiece. The results demonstrate that the methodology causes the SECF classifier to be as successful as when it was originally trained.

C. Performance Using the Adaptation Methodology for the Neural Network Classifier

The training of a neural network involves determining the ideal set of weights in each of its neurons or nodes. A trained neural network performance depends on the weights determined by training. There is no known direct relationship between weights obtained by training and physical entities like force/moment means and standard deviations. Therefore for a neural network to be able to adapt to the new situation, retraining is required. Our method can still work however by transforming the data before presenting it to the neural network. In this way we can efficiently adapt a neural network based recognition tool. Table XI lists the results of adapting the neural network based classification.

V. CONCLUSION

Our experiments show that immediately after training, both the fuzzy logic and neural network classifier correctly identify single ended contact formations with good reliability. The experiments also show that if the pose between the sensor and the object changes, neither classifier works well and would need to be retrained. Furthermore, experiments show that our approach for identifying and applying coordinate transformations to force/moment measurements allows us to regain the original performance of both classifiers without any retraining. The method only requires collecting force measurements from two face contacts. These results are important since it is no longer necessary to retrain an identifier each time another object of the same type is grasped.
There are some known limitations to the current work. The method does not take into account the effects of gravity. These effects were very small compared to the contact forces, but as the mass of the object increases the force and moment change after a large pose change may become comparable to the contact forces causing failure of our algorithm. This, however, is relatively easy to take into account. Other limitations are not as easily handled. It was assumed that the grasped objects are rigid bodies. Flexibility of the object would affect the contact recognition task. The theory developed, both in this work, and most previous work, is for polyhedral objects. It would be useful to develop a theory for curved surface contact formations. Finally, one might want to make the adaptation procedure more automated by implementing a system which automatically collects data and determines the required transformations. These problems are left for future work.

**APPENDIX**

**Proof of Identity**

Let $R$ be a rotation of $\theta$ about unit vector $n$. Define $v$ and $w$ with components parallel to $n$ and perpendicular to $n$ so that $v = v_1 + v_2$ and $w = w_1 + w_2$. Therefore, $Rv = (v_1 + v_2) + Rv_2$ and $Rw = (w_1 + w_2) + Rw_2$. It follows that $Rv \times Rw = (v_1 \times w_1 + v_2 \times w_2) + R(v_2 \times w_2)$ since $Rv_2$ and $Rw_2$ are perpendicular to $n$, then $Rv_2 \times Rw_2$ is either zero or parallel to $n$. Therefore $Rv_2 \times Rw_2 = R(v_2 \times w_2)$. We therefore have $Rv \times Rw = R(v_1 \times w_1 + v_2 \times w_2)$, therefore $R(v_1 \times w_1 + v_2 \times w_2) = R(v \times w)$. Hence $(Rv) \times (Rw) = R(v \times w)$.

**REFERENCES**


**Louis J. Everett** (M’89) received the B.S. degree from the University of Texas, El Paso, the M.S. degree from Stanford University, Stanford, CA, and the Ph.D. degree from Texas A&M University, College Station, in 1983, all in mechanical engineering. Industrial experience includes Texas Instruments, Bell Laboratories, IBM, and NASA. He is an Associate Professor of mechanical engineering at Texas A&M University. His research interests include modeling, vibration, and mechatronics. He has authored or contributed to four textbooks currently in print.

Dr. Everett is a Member of ASME and ASEE.

**Rakesh Ravuri** received the B.E. degree from Manglore University, Karnataka Regional Engineering College, Surathkal, India, and the M.S. degree from Texas A&M University, College Station, in 1998, all in mechanical engineering. Industrial experience includes Indian Space Research Organization, Electro Scientific Industries. He is a Software Engineer with Electro Scientific Industries. His research interests include mechatronics, computer graphics, vision systems, and intelligent manufacturing systems.

**Richard A. Volz** (M’60–SM’86–F’97) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Northwestern University, Evanston, IL, in 1960, 1961, and 1964, respectively.

He is the Royce E. Wisenbaker Professor of Engineering in the Computer Science Department, Texas A&M University, College Station. He served as Department Head from 1988 to 1997, and has recently embarked upon a new initiative on “Training Systems Science and Technology.” Prior to joining Texas A&M, he was Director of the Robotics Research Laboratory and Professor of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. He is the author of dozens of research papers and has led over $11,000,000 in funded research projects.

Dr. Volz received the Decoration for Exceptional Civilian Service from the U.S. Air Force and two NASA Special Service Awards. He has served as the General Chairman of the 1990 IEEE International Conference on Robotics and Automation, and as Editor-in-Chief of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION from 1994 to 1999. He has served on five federal advisory boards: the Air Force Scientific Advisory Board, the Ada Board, the Aerospace Safety Advisory Panel, a Congressional oversight committee on NASA, the NASA Space Station Advisory Panel, and the NASA Center of Excellence in Information Technology Advisory Panel.

**Marjorie Skubic** (S’90–M’97) received the Ph.D. degree in computer science from Texas A&M University, College Station, in 1997.

Prior to graduate school, she worked in industry for 14 years as a Software Engineer, specializing in real-time systems. Industrial experience during that time included Texas Instruments, TRW, and Staefa Control System. She is an Assistant Professor with the Computer Engineering and Computer Science Department, University of Missouri, Columbia. Her research interests include sensory perception, robot programming by demonstration, and haptic interfaces.

Dr. Skubic is a Member of ACM and ASEE.