Symmetry Inference in Planning Assembly

G. Dakin
Y. Liu
R. J. Popplestone
R. Weiss
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 89-28

March 31, 1989

Symmetry Inference in Planning Assembly

By G.Dakin Y.Liu S.Nair R.J.Popplestone R.Weiss

Abstract

Symmetry of a body is important in assembly both for planning and for sensing during execution. In the past we have presented a theoretical framework for simplifying the complexity of computing the interaction of multiple symmetric features. Here we describe the implementation of this framework and extend it to include finite rotational symmetries.

1 Introduction

Any statement about a feature of a body can only tell us about the location of the body modulo the symmetry group of the feature. Thus when planning or describing an assembly, statements about how features should (or might) relate spatially can be expressed in terms of the symmetry groups of the features. The same is true of the interpretation of sense data.

In this paper we show how to associate symmetry groups with *features* of body models created by the PADL2 system, and how to perform reasoning about spatial relationships between features of bodies occurring in an assembly in this group-theoretic context.

We take the view that a feature is a subset of the entities forming the boundary model of an object[15]. These will in general be infinite entities, such as planes, or cylindrical surfaces. We denote the *Euclidean Group* of all combinations of rotations and translations in 3-space by $\mathcal{E}$. A symmetry group of a feature is that subset $\mathcal{G} \subset \mathcal{E}$ each of whose members maps the feature into itself.

It is important to be able to compute the *intersection* of symmetry groups, since when we consider combinations of features of a single body, they have a symmetry which is the intersection of their individual symmetry groups. In our previous work [12] we showed how to compute the intersection of certain important continuous groups using a method of *characteristic invariants*. The characteristic spatial invariants can be *planes, lines, points*. The characteristic rotational invariants are *vectors*. For example the group $SO(2)$ of rotations about the k axis is characterised by a spatial invariant which is the point (0,0,0), and by a rotational invariant which is the **k** vector.

In the present paper we show how to extend the descriptive apparatus to allow us to reason about subgroups of $\mathcal{E}$ isomorphic to $O(2)$, and about the finite cyclic $\mathcal{C}_n$ and dihedral $\mathcal{D}_{2n}$ subgroups of $\mathcal{E}$.

1.1 Shapes

Constructive Solid Geometry is used to represent shapes as Prolog terms using the following primitives:-

block(X, Y, Z) denotes a cuboid, centroid at the origin, with the stated dimensions along the coordinate axes.

cyl(L, R) denotes a finite cylinder of length L and radius R, centroid at the origin, axis along the Z-axis, and

cone(H, R) denotes a cone of height H and radius R, base centre at the origin, axis along the Z-axis.

sph(R) denotes a sphere of radius R, centre at the origin.

tor($R1, R2$) denotes a torus of minor radius $R1$ and major radius $R2$, centroid at the origin, axis of radial symmetry along the z-axis.

screw(H, R, P) denotes a cylinder, of length L and radius R, with a screw thread, of pitch P, 'cut' in its curved surface.

gear(H, R, N) denotes a cylinder, of length L and radius R, with N gear teeth 'cut' in its curved surface.

These primitives are combined with the boolean operations \/ and /\ for union and intersection, and with \ denoting set subtraction. The term *Shape*@*Loc* denotes a *Shape* relocated by the location *Loc*.

It is a convenience to allow *Shape*@*Vec* to denote the *Shape* translated by the vector *Vec*.

2 Labelling a solid model with the feature symmetry groups

In order to apply the theory to actual robotic reasoning, we have made use of the PADL2 [16] modeller to provide us with a ***boundary representation*** of solids. Solid shapes, specified as Prolog terms in the formalism defined above are input to PADL2, and processed by that system into boundary models. For example:

```
cyl(4,1)  \/ (cyl(4,1)@trans(4,0,0)) \/
            (block(1,2,6)@trans(-1,-1,1)@rot(jj,1.570796)) )
```

denotes an object, consisting of two cylinders "stuck" on to a block with the union operation \/. A ***draw*** predicate is provided which "prints" the Prolog term as a string in PADL2 syntax, and then calls PADL2 to form and display a boundary model of the object. The PADL2 internal representation of the boundary model is extracted using FORTRAN subroutines linked into POPLOG [5,14] as external procedures; POP-11 objects expressing the face-edge-vertex structure of a body are built from the information thus extracted. These objects correspond quite closely with the structures that PADL2 uses internally.

In order to represent bodies such as gears and screws concisely, we have extended the CSG ***formalism*** to encompass them, as described above. They are presented to PADL2 as circumscribing cylinders, however. Having passed through PADL2 we then identify those surfaces in the boundary model which had originally been specified as gears and screws, and relabel them as such. In effect, we are treating these forms, of great functional importance in Mechanical Engineering, as 'texture' on a surface. This is of course an approximation, but one without very serious consequences, since shape is required:

- For predicting clearance between parts: For this purpose the cylindrical approximation is very adequate, since in almost all designs it is sufficient to plan to keep the cylindrical approximations clear of each other.
- For determining mating relations: Gears and screws (the actual formed surfaces that is) hardly ever interact with other features in a way that demands detailed knowledge of form save when they interact with other gears and screws respectively, and in this case the legality of the interaction can be determined from the form parameters, such as pitch and pressure angle.
- For visualisation: in most cases we are concerned with, the approximation is adequate.

Next, each face F of the model is labelled with its symmetry group, each group being considered as the image $f^{-1}G_{canon}f$ of a canonical subgroup of $\mathcal{E}$ under an inner automorphism. A token denoting the canonical subgroup G_{canon} is obtained by table lookup from the surface type of the face, using a property procedure *gr_canon*. E.g. if F is a conical face *gr_canon*(F) is a token denoting the group $SO(2)$ of all rotations about the Z-axis. The f for the inner-automorphism is the location (transformation matrix) of the face in body coordinates as given by PADL2.

It is possible to use this location because the way in which coordinate systems are embedded in features by PADL2 permits a coherent and consistent choice of canonical groups — largely this is because the Z-axis is chosen by PADL2 to be the axis of symmetry. To put this another way, PADL2 boundary model entities are relocated instances of entities whose symmetry group is one of our canonical set.

Using these group-labelled models, we have implemented the technique described in [12] The transformed canonical groups are themselves held as pairs (G_{canon}, f), where G_{canon} is the token representing the canonical symmetry group of the face, and f is the location of the face. The characteristic invariants called for in [12] are obtained from G_{canon} by table lookup, and are relocated by f to give the characteristic invariants of the group $f^{-1}Gf$.

3 Relating Bodies in an Assembly

When bodies are mated together in an assembly, certain relationships are established between their features. These relationships can be classified as either *fitting* or *against* relationships, where *fitting* implies an areal contact, which means that the features, over the area of contact, have the same symmetry group. Now, if we are using a modeller like PADL2, which has the property that any *area* of a model face *determines the symmetry group of the face surface* then we can conclude the following:

If the two bodies are B_1 and B_2 and the features are F_1 and F_2, with locations f_1 and f_2 in body coordinates, then p_2, the location of B_2 in world-coordinates is given in terms of p_1, the location of B_1 by

$$p_2 \in p_1 f_1' G f_2^{-1} \tag{1}$$

where G is the common symmetry group[1]. In cases where a convex feature derived from a positive occurrence of a primitive in the CSG fits a concave feature derived from a negative occurrence, we have $f_1' = f_1$. However, in the case of mating *plane* faces, which, having zero gaussian curvature, can play the role of being both concave and convex, $f_1' = f_1 \mathbf{rot}(\mathbf{i}, \pi)$, since the outward-pointing normals of the two surfaces will point in opposite directions.

If all of the relationships between a pair of bodies are fitting relationships, then we can regard this as a single fitting relationship between a *compound feature* on each body. A compound feature is just a *set* of geometric entities occurring in the boundary model. The symmetry group of the compound feature is obtained by *intersecting* the symmetry groups of its component features (compound features have *less* symmetry than simple features)[12]. This analysis supposes that the features making up a compound feature are to be *distinguished*. The case where some of the features making up a compound feature can be *identified* as for example when we have equi-spaced parallel holes, equidistant from a common axis, is not covered in this paper.

[1] Note that in contrast with earlier work [3][9][10] we are now using a convention of pre-multiplying column vectors by location matrices, and also using a the Z-axis as the axis of symmetry.

This treatment of relationships in terms of the symmetry groups of compound features is substantially more concise than that used in [9], where $O(n^4)$ cases have to be considered, where n is the number of distinct surface types. For the cases covered by [12], we have had to implement the *prox* functions described in that paper. Since there are 3 spatial invariants, there are 9 cases to be considered for the spatial invariants, usually with some sub-cases derived from parallelism. There is one case to be considered for vector invariants. Our present approach also generalises to features with finite symmetries. In addition, as discussed in section 5, it provides a good basis or planning assemblies where the features to be mated are not identified by the user.

4 Cycles and chains of spatial relationships

We are using this approach to produce a concise implementation of much of the spatial relation inference mechanism underlying the RAPT language[3]. 2-cycles of *fits* relationships (where the *plane against plane* relationships are now re-classified as fitting, since they involve areal contact) can be dealt with as relationships between compound features. However, RAPT also requires us to deal with larger cycles. Cases of these, important for assembly, can be treated by using a kind of transitivity that holds among spatial relationships when certain alignments exist. For example, if a block B_3 is placed on a block B_2 which itself is placed on a block B_1, then B_3 can be regarded as being placed on an imaginary surface of B_1 placed at a height equal to the thickness of B_2 above the actual top surface of B_1.

Within the group-theoretic framework this appears generally as follows:

Suppose we have two bodies B_1 and B_2 each of which has features F_1 and F_{21} and these features *fit* each other. Suppose also B_3 is related to B_2 because a feature F_{22} of B_2 fits a feature F_3 of B_3. Then, using condition 1

$$p_2 \in p_1 f_1 G_1 f_{21}^{-1} \tag{2}$$

$$p_3 \in p_2 f_{22} G_2 f_3^{-1} \tag{3}$$

Where as before, the f's are the locations of the corresponding F features, possibly combined with an interface element. Hence combining these conditions:

$$p_3 \in p_1 f_1 G_1 f_{21}^{-1} f_{22} G_2 f_3^{-1} \tag{4}$$

In general this is not of the form corresponding to a simple *fits* relation between B_3 and B_1. However as in the brick example above, in some cases of practical importance it will be so. Let $f = f_{21}^{-1} f_{22}$. Let $S \subset \mathcal{E}$ be the ***subset:***

$$S = G_1 f G_2$$

If S is a coset of some subgroup of $\mathcal{E}$ then we will have a simple fits relationship between B_1 and B_3.

An important case in which we can obtain a simple relationship between B_1 and B_3 is when there is a *commutation condition* between f and one of the groups G_1 or G_2. Suppose we have $G_1 f = f' G_1$. Then we obtain:

$$p_3 \in p_1 f_1 f' G_1 G_2 f_3^{-1} \tag{5}$$

In our brick example, the members of G_1 are of the form $\mathbf{trans}(0, y, z)\mathbf{rot}(\mathbf{k}, \theta)$, and f is of the form $\mathbf{rot}(\mathbf{i}, \pi)\mathbf{trans}(0, 0, c)$, where c is the thickness of the brick B_2. We have

$$\mathbf{trans}(0, y, z)\mathbf{rot}(\mathbf{k}, \theta)\mathbf{rot}(\mathbf{i}, \pi)\mathbf{trans}(0, 0, c)$$

$$= \mathbf{rot}(\mathbf{i}, \pi)\mathbf{trans}(0, 0, c)\mathbf{trans}(0, -y, -z)\mathbf{rot}(\mathbf{k}, -\theta) \in fG_1$$

so that the commutation is exact, i.e. f' = f.

The product G_1G_2 may itself be a subgroup, as for example when $G_1 = G_2$. Otherwise let G be the group-theoretic join [2] $G_1 \cup G_2$ of the two groups G_1 and G_2. Then we can weaken 5 to the following:

$$p_3 \in p_1 f_1 G f_3^{-1} \tag{6}$$

This condition is in general weaker than 4 because the *product* of subgroups and elements is not necessarily equal to the group theoretic join. Nevertheless it does provide useful constraints between B_1 and B_3.

The approach described in this section is being used to implement the operations associated with Table 2 of the RAPT implementation described in [3].

5 Planning what body features to relate

In specifying assemblies it is often the case that the actual mating *features* of bodies are not specified — only the fact that bodies themselves are to be mated. Therefore we have studied how to infer sets of possible mating features of two bodies. Given the combinatorics of this problem, we have explored one approach which is the identification of compound features of a body which are instances of compound features appearing in a library.

Some of the library features are quite specific such as: *countersink, counterbore, keyway* and certain cases of *spline*. More generic assembly-relevant features are *insertors, containers, multi-insertors* and *multi-containers*, which are in effect general protrusions, concavities, and combinations of these. Feature definitions refer to the faces of the features of a single body and relationships between them, such as being adjacent, perpendicular, parallel etc..

Having identified these standard compound features in two bodies, the symmetry groups of the compound features are calculated. Candidate mating features will have the same symmetry group.

Dimensional consistency of candidate mating features is also required. There are two kinds of dimensions involved.

- The parameters of each PADL2 surface that is a component of one compound feature should be consistent with the parameters of the corresponding surface component of the other compound feature.
- Sets of characteristic invariants used in calculating the intersection groups have intrinsic dimensions (e.g. the length of the common perpendicular between line invariants and the angle between them): these dimensions should be consistent between corresponding compound features.

[2] That is the closure under the group operations of the set-theoretic union

6 The characteristic invariants for the Euclidean Group

As discussed earlier, the symmetry group of compound features is the intersection of the symmetry groups of the components. By using characteristic invariants as in [12], this has already been solved for an important class of the subgroups of the Euclidean group, viz. the translational groups, $\mathcal{T}^1, \mathcal{T}^2, \mathcal{T}^3$, the rotation groups $SO(2), SO(3)$, the group of the plane $\mathcal{G}_{plane}$, the group of the cylinder $\mathcal{G}_{cyl}$, and the identity group. This section discusses the extension of the work to certain discrete subgroups of these groups, in particular, the finite rotation groups and to the orthogonal group $O(2)$. Note that an element of $O(2)$ can be written as a three-dimensional rotation. The finite rotation groups considered are the cyclic subgroups $\mathcal{C}_n$ of $SO(2)$ and the dihedral subgroups $\mathcal{D}_{2n}$ of $O(2)$. The general approach that is taken here is that the intersection of these discrete groups can be computed from the intersection of the corresponding continuous groups which contain them. Thus, if $H_1 \subset G_1$, $H_2 \subset G_2$, then $H_1 \cap H_2 \subset G_1 \cap G_2$.

6.1 Cyclic groups

A cyclic group of finite order n can be considered as subgroup of $SO(2)$, and can be represented as

$$\text{rot}\left(\vec{v}, \frac{2k\pi}{n} | k \in \mathbf{Z}\right)$$

Thus, it has a rotational invariant which is a vector v pointing along the axis of rotation, and an integer n. The intersection of two cyclic groups is the set of solutions to the equation:

$$\text{rot}\left(\vec{v_1}, \frac{2k_1\pi}{n_1}\right) = \text{rot}\left(\vec{v_2}, \frac{2k_2\pi}{n_2}\right)$$

This set is just the trivial group unless $v_1 = v_2$. If $v_1 = v_2$, then the intersection is also a cyclic group having the same axis of rotation, and the order of this group is the greatest common divisor of n_1 and n_2. As a result, if $n = max(n_1, n_2)$, then the intersection of two such subgroups can be computed in $O(\log(n))$ time [8]. This analysis also applies to subgroups of the group of the cylinder and the group of the plane, since the quotients by the translations are cyclic rotation groups.

6.2 The orthogonal group

The group O(2) is the smallest group which includes all rotations about the Z-axis together with a rotation of π about an axis in the X-Y plane. This means that O(2) contains an infinite number of subgroups of order two. We define its rotational characteristic set to be the set of two vectors that lie along the principal axis and are negatives of each other, $\{\vec{k}, -\vec{k}\}$. This makes it possible to intersect O(2) with other O(2) and SO(2) groups and their finite subgroups. For example, two copies of O(2) whose principal axes are orthogonal intersect in a dihedral group of order 4, $\mathcal{D}_4$. A copy of O(2) will intersect an orthogonal SO(2) in a cyclic group $\mathcal{C}_2$.

The dihedral groups are finite subgroups of O(2), each of which is a semi-direct product of a cyclic group of order n with a rotation by π radians about an orthogonal axis. The order of such a dihedral group is $2n$, and can be represented as a subgroup of O(2) by a pair of vectors $\{\vec{k}, -\vec{k}\}$ together with n, half the order of the group. The only ambiguity arises when $n = 2$, and this can be handled as a special case.

7 From Relations to Actions

Supposing now that our robotic system desires to achieve a world state characterised by a 'goal graph' of relations, we see that each relationship expressed in the 'goal graph' has to be achieved by executing an action. Since, in general, relationships are not achieved simultaneously by a single action, some actions will be done before others, and consequently some actions must preserve relations already established by other actions. This, of course, corresponds to the pre-condition/post-condition treatment of actions and conditions of planning work in AI. However, in assembly, the interaction of actions and relations is strongly dependent upon geometry. Moreover, as we shall see, the need to preserve an existing relation can actually modify behavior during an action to achieve another.

We say that an action preserves a relationship if, when the relationship holds before the action it holds after. This can be regarded as a kind of commutativity condition. There are two ways in which an action may preserve a relationship.

- Passively: the relationship is maintained by existing forces (friction, gravity), and the action applies no forces which can disturb the relationship. For example, the action may be the insertion of a part remote from the parts between which the relationship holds.
- Actively: The action may affect bodies between which the relationship holds. For example, we may have achieved a relationship in which the hub of a wheel *fits* a shaft, and be attempting to slide the wheel along the shaft until the side of the hub is *against* a face that provides lateral location.

Of course it is also the case that some actions cannot preserve a relationship, or cannot be executed if a relationship holds. E.g., you cannot put an object in a box if you have already put the lid on.

Passive preservation has no implications for the lower level controller— deciding that an action passively preserves a relation is a matter of verifying the stability of the relation and the clearance of the chosen trajectories.

Active preservation gives rise to most of the variety of control required for an assembly task. Given that a relation holds between two bodies, then a relative movement of the bodies in a certain subspace of twist space will preserve the relation.

In the case of *fits* relations, where body features of the same geometric form have areal contact, the two features have a common symmetry group, and the relation preserving subspace can readily be derived from the canonical invariants of the group, which are described in [12].

However, given uncertainty in body shape, robot location, etc., there will be a difference between the nominal relational preserving subspace of twist space as calculated by the planner, and the actual as present to the robot. It will be necessary to use force sensing to enable the robot to comply with the actual subspace.

Following the approach described in [4], a quasi-static behavior of the robot is defined using a stiffness matrix. The stiffness matrix for a given action which preserves a *fits* relation should be chosen so that the stiffness will be high along the directions in which a twist preserves the relation, and low along complementary directions in which compliance is desired.

In some cases this will be sufficient to maintain the existing relationship, while a twist is applied in the direction which, while preserving the existing relationship, brings the consummation of the desired relationship nearer. This is the case when a cylindrical shaft fits a cylindrical hole — wrenches can be exerted in directions orthogonal to those in which twists can be achieved. However, in general, the situation is more complicated, and depends on the geometry of the contacting features, and not just on their symmetry group(s), because certain wrenches will cause the related surfaces to separate. In these cases it will be necessary to apply a bias wrench, which is in the opposite direction to the separating wrenches, and whose magnitude is determined by the magnitude of the uncertainties in the force measuring system.

The analysis of the conditions under which separation can occur depends on the finite extent of the contacting surfaces. In general it is difficult to compute the conditions under which the separation of the surfaces can occur. However there are some simple cases it should be possible to treat, for example:

- To maintain contact between plane surfaces, the bias wrench should be along the outward normal of the plane surface of the manipulated body, with a line of action contained within the convex hull of the contacting area.
- Where we have fitting surfaces with a single continuous rotational symmetry (the SO(2) or O(2) group), it is possible to determine that separation cannot occur in a given relative location by projecting the contact surface onto a one-dimensional space which is a parameter for the continuous symmetry. If the projection covers more than π, then separation cannot occur.

8 Conclusion and future work

The algorithms described in our previous paper [12] have been implemented in the POP-11 language [5,14]. The PADL-2 modeller is linked in as a package of external procedures, and the group labelling algorithms implemented. An inference engine, modelled on that of RAPT[3] has been implemented, using the group-theoretic method, and providing treatment of multiple bodies and situations. The theory of section 4 is currently being implemented, and we expect to extend our implementation of [12] to include the additional groups described in Section 6 during the coming year. In addition, we plan to investigate cases in which finite symmetries are created by multiple features that are interchangeable, e.g. equally-spaced, parallel, identically-shaped holes.

References

[1] Barrett,R., Ramsay,A. and Sloman A., [1985] POP-11 A Practical Language for Artificial Intelligence, Ellis Horwood, Chichester, England and John Wiley N.Y.,USA.

[2] Clocksin W. and Mellish C. 1981, Programming in Prolog, Springer Verlag N.Y..

[3] Corner D.F., Ambler A.P., and Popplestone R.J., 1983, Reasoning about the Spatial Relationships Derived from a RAPT Program for Describing Assembly by Robot, Proc 8th. IJCAI, Karlsruhe,DBR., pp 842-844.

[4] Grupen, R.A., 1988, General Purpose Grasping and Manipulation with Multifingered Robot Hands, Ph.D Thesis, University of Utah, Computer Science Department, Merrill Engineering Building, Salt Lake City, UT 84112.

[5] Hardy S., 1984, A New Software Environment for List-Processing and Logic Programming, in Artificial Intelligence, Tools, Techniques and Applications, eds. O'Shea,T. and Eisenstadt,M., Harper and Row, N.Y..

[6] J.M.Hervé 1978 Analyse Structurelle des Mécanismes par Groupe des Déplacements. Mechanism and Machine Theory, Vol 13 No 4.

[7] Shokichi Iyanaga and Yukiyosi Kawada 1968, The Encyclopaedic Dictionary of Mathematics, English edition M.I.T. Press, Cambridge. MA.USA. and London, Japanese original published by Iwanami Shoten, Tokyo.

[8] Knuth, D.E., *The Art of Computer Programming*, Vol.2, Addison-Wesley, 1981.

[9] Popplestone,R.J., Ambler A.P., and Bellos I. 1980, An Interpreter for a Language for Describing Assemblies, Artificial Intelligence Vol 14 No 1, pp 79-107

[10] Popplestone,R.J, 1984, Group Theory and Robotics, Robotics Research: The First International Symposium, (eds Brady,M. and Paul,R.), MIT Press, Cambridge MA and London.

[11] Popplestone, R.J. 1988 "The Edinburgh Designer System as a Framework for Robotics: The Design of Behavior", AI EDAM, Vol 1 No 1.

[12] Popplestone R.J. Weiss R.,Liu,Y. [1988] "Using Characteristic Invariants to Infer New Spatial Relationships from Old" *Proc. IEEE Conference on Robotics and Automation*, Philidelphia, April 1988, pp 1107-1112.

[13] Requicha A.A.G. and Tilove R.B., 1978 Mathematical Foundations of Constructive Solid Geometry: General Topology of Closed Regular Sets, TM27A, Production Automation Project, University of Rochester, Rochester N.Y..

[14] Sloman A. and Hardy S. 1983, POPLOG: A Multi-purpose Multi Language Program Development Environment. AISB Quarterly, No 47, 1983.

[15] Woodbury,R.F. and Oppenheim,I.J. [1988] An Approach to Geometric Reasoning in Robotics, IEEE Transactions on Aerospace and Electronic Systems.

[16] Requicha, A.A.G. and Voelcker, *Solid Modeling: A historical Summary and Contemporary Assessment*, IEEE Computer Graphics Applications, vol.2, no. 2, pp. 9-24, March 1982.