Contact Constraint Analysis and Determination of Geometrically Valid Contact Formations from Possible Contact Primitives *

Jing Xiao[†]and Lixin Zhang[‡]

Abstract

A complete, precise, and systematic analysis on the geometrical nature of contacts between two arbitrary polygons and derivation of the geometric contact constraints between two such polygons are provided. Based on the results, a general algorithm is presented to identify the geometrically valid contact formations (CFs) from a given set S_{pc} of possible principal contacts (PCs) between two polygonal objects with location uncertainties. For any (non-empty) subset of S_{pc}, the algorithm tests if the PCs in the set form a possible CF by verifying the geometrical contact constraints. The obtained set of geometrically valid CFs can serve as input to an additional verifier based on force sensing which can extract the actual CF from the set. The notion of *equivalent CFs* is introduced to describe those contact situations constraining the relative location between two objects to the same region. This concept proves to be extremely useful to the completeness of the analysis and efficiency of the algorithm. Of the many discoveries, a particularly significant one is that *despite uncertainties in object locations*, in many cases, if two or more PCs are formed between two objects, their relative location is fixed or can take up to only four solutions.

1 Introduction

In this section, we introduce the background and motivation of this research, explain the characteristics of our work, and provide the outline of the paper.

1.1 Problem and Research Objective

Automatic contact recognition can play a key role in successfully accomplishing automatic assembly tasks in the presence of uncertainties. However, due to uncertainties,

*This research was supported by an NSF grant under grant number IRI-9210423.

[†]Department of Computer Science, University of North Carolina, Charlotte, NC 28223, USA; e-mail: *xiao@uncc.edu.*

[‡]Currently with Datastream System Inc., 50 Datastream Plaza, Greenville, SC 29605, USA; e-mail: *zhangl@dstm.com.*

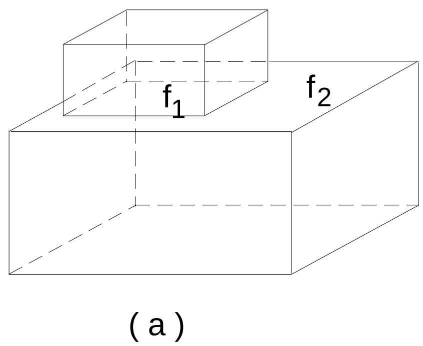

Figure 1: An example

the contact recognition task itself is also non-trivial, especially for nonconvex objects. Two major and related issues are involved in the task: (1) how to represent a contact state, and (2) how to recognize a contact state in the presence of uncertainties.

To describe a contact state between two objects in terms of the contacting topological surface elements (i.e., faces, edges, and vertices) has many advantages and is the most common kind of contact representations. One such representation was introduced by Lozano-Pérez [12] in which point contacts, for example, vertex-edge contacts in 2D objects and vertex-face contacts in 3D objects, were used as contact primitives, and a contact state was described as a set of such primitives. Desai *et al.* [3] defined contact primitives differently; they used the single contacts between a pair of topological surface elements of two polyhedra as primitives, called *elemental contacts* (ECs), and described a contact state (between the two polyhedra) in terms of the set of ECs formed, called a *contact formation* (CF).

Noticing that some ECs are redundant in characterizing the essence of a contact state relating to contact motions, Xiao later [18] proposed *principal contacts* (PCs) as those ECs necessary for characterizing motion freedom and defined a contact state in terms of the set of PCs formed, still called a contact formation. PCs are higher-level primitives and are less sensitive to location uncertainties of the contacting objects than ECs or the primitives proposed in [12]. For example, in Fig. 1a, the contact state can be simply described by the single PC $f_1 - f_2$, whereas, more primitives will be required if the state is to be described in terms of either one of the other two kinds of primitives. The state in Fig. 1b is described by the same PC as that in Fig. 1a. Indeed, the two states are so similar that they may not need to be distinguishable for planning a contact motion and may not possible to be distinguished due to uncertainties (such as location and force/moment sensing uncertainties). However, by using the other types of primitives, these two states will be described differently. Thus, the concept of PC has advantages.

Only a few researchers addressed the problem of contact recognition *in the presence of uncertainties*, and their methods were based almost universally on the approach of hypotheses-and-tests. Desai and Volz[4, 5] used force/moment sensing data; Xiao and Su[19] used vision sensing to verify the contact hypotheses — a set of *possible* contact states taking into account uncertainties. More recently, Farahat, Graves, and Trinkle[8] used a linear programming approach to the force/moment testing of different contact hypotheses. Spreng[14] used test motions for verifying contact hypotheses in terms of motion freedoms.

Clearly, a key problem is how to obtain effectively the initial contact hypotheses (possible contact states) in the first place. From the view point of the configuration space obstacle (C-obstacle) of one contacting object with respect to the other, such initial contact hypotheses could be the contact states involved in the intersection of the C-obstacle with the location uncertainty region of the relative location. However, it is difficult to generate the boundary expression of a C-obstacle analytically for an object which can both translate and rotate in a world of 3D objects: no previous work has done that [15], and since such boundary is usually of high-dimensional curved surfaces, it is also difficult to perform the intersection. To avoid the above problems, Xiao [20] introduced a method of growing objects by their location uncertainties in the 3D physical space, based on which, Xiao and Zhang [21] developed and implemented an algorithm to recognize the set (S_{pc}) of all the PCs (principal contacts), which were possibly established due to location uncertainties of objects, where the objects were arbitrary 3D polyhedra.

The set S_{pc} can be used as contact hypotheses for contact verifiers based on other sensing means, such as force sensing[4, 5, 8] and vision sensing[19], to extract the actual contact formation (CF), i.e., the set of PCs actually formed between the two objects, in spite of uncertainties[1]. To facilitate this process, we find it desirable to have a mechanism for automatically testing whether a subset of S_{pc} forms a geometrically valid (or possible) CF or not, observing that (1) a valid CF must consist of PCs whose co-existence is geometrically possible and (2) not all PCs can possibly co-exist. In this paper, we introduce such a mechanism based on analyzing geometric contact constraints. With this mechanism, all geometrically possible CFs can then be obtained from S_{pc} (one of which is the actual CF); such information will be extremely useful for contact verifiers based on force sensing.

In general, the contacting topological elements of objects give three types of constraints on the relative locations of objects The *equality constraints* are to specify, for example, that two vertices in contact have the same location. The *overlap constraints* are inequality constraints specifying that the contacting elements have overlapping regions; for example, two parallel edges in contact must have an overlapping segment. The *non-penetration constraints* are also inequality constraints specifying that the contacting objects do not penetrate into each other. As our work is about how to derive and analyze such contact constraints, it is necessary to clarify the relation and difference between our approach and previous work also related to contact constraints.

[1] Note that if two contacting objects are both convex, then the CF between them consists of only a single PC. In a general case, however, a CF may consist of many PCs.

1.2 Related Work

Some of the earliest work was done by Ambler, Popplestone, and Bellos [1, 13], Taylor [16], and Lozano-Pérez [11] for specifying assembly operations. These researchers (1) specified the spatial relationships among objects by symbolic descriptions of the relationships among topological elements (features) of the objects, which were mostly contact relationships, and then (2) derived the relative locations of objects from such symbolic descriptions. The work in [1, 13] consider only the equality constraints given by the spatial relationships among object features. These constraints are described in symbolic equations and solved by a symbolic equation simplifier and an equation solver. However, not all symbolic equations can be solved by this method, especially when there are degenerate contacts, such as vertex-edge, vertex-vertex, edge-edge contacts.

Taylor [16] extended the work by Ambler *et. al* [1] by considering many types of inequality constraints, such as overlap constraints. However, Taylor restricted his study to cases involving only a known rotation axis to avoid the complexity of non-linear constraint equations (introduced by rotations), and linearized the constraint equations by dividing up the range of that single rotation. The resulting linear equations were solved by linear programming methods. As in any linearization approach, there is a tradeoff between accuracy and the speed of his algorithm.

Lozano-Pérez [11] focused his study to the spatial relationships of objects with simple geometries, such as cuboids and cylinders. He only considered the constraints given by the spatial relationships among faces of the objects, which include contact constraints. These contact constraints consisted of equality constraints and, in certain cases, overlap constraints. A contact state usually involved several such constraints, and Lozano-Pérez computed the set of legal locations of the objects determined by the contact state by finding the intesection of the sets of legal locations determined by individual constraints. The method, though efficient for the very restricted cases and constraints considered, is not extendable as a general strategy.

In order to represent C-obstacles of a world of polyhedral objects, Lozano-Pérez [12] and Donald [6] proposed a different form of constraints, called C-constraints. Each C-constraint corresponds to a primitive point contact (as introduced in Section 1.1) between two objects. A C-constraint requires that (1) two contacting features do not result in interpenetration of their respective objects in the neighborhood of the contacting points, called the applicability condition, and (2) two contacting features are on the same supporting line (for 2D objects) or plane (for 3D objects), called the C-surface condition.

The C-constraints were later used by Brost [2] and Farahat *et. al* [7] to analyze the characteristics of contact configurations between 2D objects in a configuration space. Brost [2] used numerical methods to solve the trigonometry equations derived from the C-surface conditions of C-constraints. As pointed out in [7], Brost's analysis missed some non-generic contact types. Farahat *et. al* [7] discovered many conditions to discriminate generic from non-generic contact situations and showed that the contact formation cells were smooth surfaces. They also showed the number of possible configurations in a generic contact state. However, the analyses in both [2] and [7] did not consider all contact constraints. First, the C-constraints were introduced to represent C-obstacles, and they do not represent all the contact constraints. To completely characterize a contact, more constraints should be considered, such as overlap constraints and *global* non-penetration constraints, which are much stronger than the applicability conditions in the C-constraints, for the applicability conditions only require non-penetration *locally*. Secondly, both [2] and [7] only considered the C-surface conditions in the C-constraints.

1.3 This Work

Several major characteristics distinguish our work from the previous work.

1. To tackle our problem, we take into account *all* contact constraints (i.e., equality constraints, overlap constraints, and non-penetration constraints) as well as the location uncertainties of contacting objects.

As reviewed above, the analyses in [1, 13, 11, 2, 7] did not cover all contact constraints and location uncertainties. Consequently, what we report in this paper is far more complete than the related previous results. For instance, we have obtained similar results as in [7] about the number of possible configurations in a generic contact state, but these are only a small part of the results presented in this paper.

2. We take into account all possible contact states among arbitrary 2D polygons and our analysis is complete and precise.

Some previous work [2, 7] focused on a set of *common* contact situations among 2D polygons, but there seems no previous work that offers a complete, general and precise analysis on *all* contact states among arbitrary 2D polygons, which our work in this paper provides.

There is also previous work handling 3D polyhedra, but these analyses either only considered certain types of contact constraints [1, 13, 11] and certain simple types of objects [11] or used approximations [16]. We expect to extend our work to general contacts between 3D polyhedra.

3. Our analysis is based on using principal contacts (PCs) as primitives, unlike the work in [2, 7], which used lower-level primitives (point contacts).

As introduced in Section 1.1, PCs have special advantages, and the fact that PCs are higher-level primitives simplifies our analysis.

4. We introduce the concept of *equivalent CFs* as the CFs which constrain the relative location between the two contacting objects in the same region, i.e, have equivalent contact constraints.

The concept of equivalent CFs greatly facilitates the symbolic reasoning in our algorithm for analyzing *all* contact situations and enables the completeness of the analysis and efficiency of the algorithm.

The paper is organized as the following. In Section 2, we introduce conventions used in this paper and describe object parameters. In Section 3, we provide an

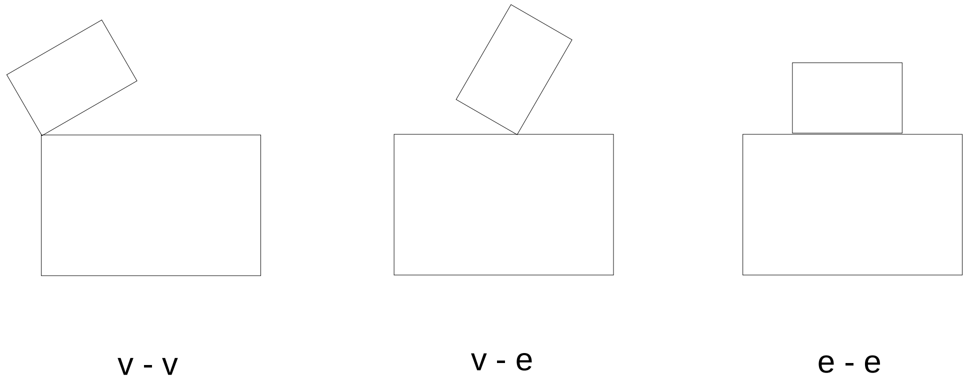

Figure 2: Principal contacts in 2D space

overview of our method of using contact constraints to identify a valid CF. In Section 4, we analyze contact constraints and present the results. In Section 5, we present our algorithm in detail based on the results described in Section 4. We conclude the paper in Section 6.

2 Notations and Nomenclatures

In this section, we review the contact models, define coordinate systems for objects and their topological elements, and describe the corresponding geometric parameters.

2.1 Principal Contacts and Contact Formations

For polygons, there are two topological elements: edges and vertices. Accordingly, there are three types of PCs as shown in Figure 2, which are denoted as vertex-vertex (*v*-*v*) PC, vertex-edge (*v*-*e*) PC and edge-edge (*e*-*e*) PC.

A CF is a set of PCs formed between two objects. There are seven types of CFs which only have two PCs. These are *(v-v, v-v)*, *(e-e, e-e)*, *(v-v, e-e)*, *(v-v, v-e)*, *(v-e, e-e)*, *(v-e, v-e)*, *(e-v, v-e)* CFs.

2.2 Coordinate Systems (Frames)

We define the frames for polygons and their edges and vertices (Fig. 3) using the right-hand rule. In addition,

- the frame for an vertex v is defined such that its origin is at v and its orientation is the same as that of the object which the vertex v belongs;

- the frame for an edge e is defined in such a way that its positive OX axis is along the outward normal of e and its origin is at one vertex of e so that e is on the positive OY axis.

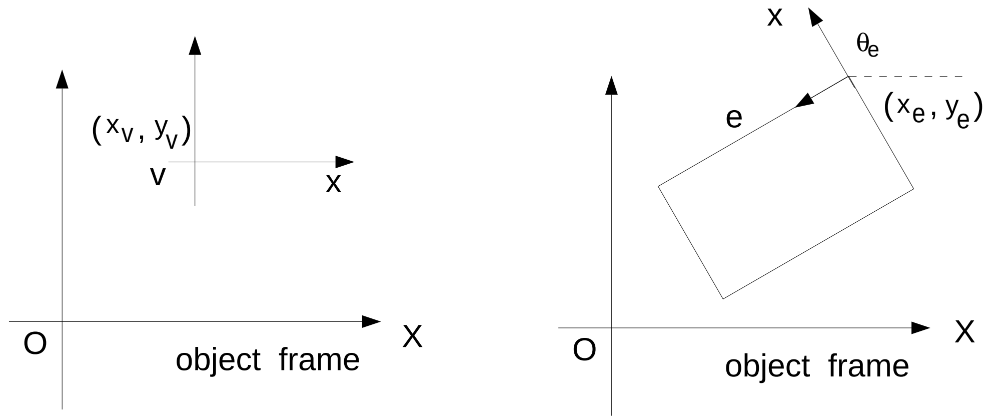

Figure 3: Frames for polygons and their topological elements

2.3 Known Geometric Parameters

We consider two polygons H and F and also use H and F to denote their object frames respectively. Similarly, we use v and e to denote both a vertex and an edge and their corresponding frames respectively. Let P denote either H or F and u denote either v or e. Then, u_i^P denotes the i-th element of polygon P, and we are ready to introduce the following parameters:

${}^P T_{u_i^P}$	3×3 Homogeneous transformation matrix from the element frame to the object frame
$d_{u_i^P u_j^P}$	Distance between two vertices, the line of an edge and a vertex, or the lines of two edges
$l_{e_i^P}$	Length of the edge
$\theta_{e_i^P e_j^P}$	Angle between the two edges

It is important to note that since we do not consider modeling uncertainty in the geometric models of the objects, we assume the above parameters have no uncertainty and are *known precisely*.

We can further express ${}^H T_{v_i^H}$, ${}^F T_{v_i^F}$, ${}^H T_{e_i^H}$, and ${}^F T_{e_i^F}$ respectively as the following:

$$ {}^H T_{v_i^H} = \begin{bmatrix} 1 & 0 & a_{ix}^H \\ 0 & 1 & a_{iy}^H \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{a}_i^H \\ 0 & 1 \end{bmatrix} = T_{trans}(a_{ix}^H, a_{iy}^H) $$

$$ {}^F T_{v_i^F} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{a}_i^F \\ 0 & 1 \end{bmatrix} = T_{trans}(a_{ix}^F, a_{iy}^F) $$

$$ {}^H T_{e_i^H} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{b}_i^H \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}(\alpha_i) & \mathbf{0} \\ 0 & 1 \end{bmatrix} = T_{trans}(b_{ix}^H, b_{iy}^H) \cdot T_{rot}(\alpha_i) $$

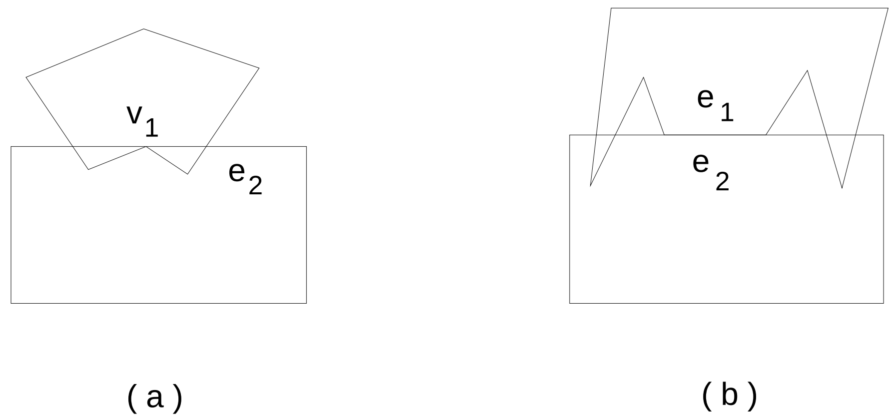

Figure 4: Penetration between two objects

$$^{F}T_{e_i^F} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{b}_i^F \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}(\beta_i) & \mathbf{0} \\ 0 & 1 \end{bmatrix} = T_{trans}(b_{ix}^F, b_{iy}^F) \cdot T_{rot}(\beta_i)$$

where $\mathbf{R}(*)$ is a 2×2 rotation matrix (about OZ axis).

2.4 Spatial Parameters

We use the homogeneous transformation matrices $^{W}T_H$ and $^{W}T_F$ to represent the locations of H and F respectively in a world (reference) coordinate system W. $^{W}T_H$ and $^{W}T_F$ are subject to location uncertainties, and thus only their sensed (or estimated) values $^{W}\hat{T}_H$ and $^{W}\hat{T}_F$ are known.

We use the homogeneous transformation matrix $^{F}T_H$ to denote the relative location of H with respect to F. Based on $^{W}\hat{T}_H$ and $^{W}\hat{T}_F$, the estimated value $^{F}\hat{T}_H$ of $^{F}T_H$ can be decided:

$$^{F}\hat{T}_H = (^{W}\hat{T}_F)^{-1} \cdot {}^{W}\hat{T}_H$$

As we will describe in more detail later, in many cases where objects H and F are in a CF of two or more PCs, $^{F}T_H$ is fully constrained and can be precisely determined regardless the uncertainties in $^{W}T_H$ and $^{W}T_F$.

3 Overview

As introduced in Section 1.1, there are three types of contact constraints on the relative location $^{F}T_H$ of two contacting objects H and F: *equality constraints*, *overlap constraints*, and *non-penetration constraints*. For each type of PC formed between H and F, we can express the equality constraint and the overlap constraint (which is an inequality) easily in terms of the parameters of the contacting elements and bounds on certain free variables (see Section 4.1).

It is, however, not always possible to express the non-penetration constraints for a PC in terms of only the contacting elements in the PC. This is because the existence of a PC may cause topological elements other than the ones in the PC to penetrate into the other object. For example, in Fig. 4a, the PC v_1-e_2 cannot hold because it leads to the penetration of e_2 into the other object; whereas, in Fig. 4b, the PC e_1-e_2 cannot hold because it leads to the penetration of elements other than e_1 and e_2 into the other object. Thus, to explicitly formulate the non-penetration constraints is rather difficult.

Fortunately, our analyses (Section 4) show that although objects H and F have location uncertainties, if two or more PCs are formed between object H and object F, under the equality and overlap constraints alone, ${}^FT_H$ is constrained, in many cases, to a unique location or at most 4 possible locations. We can take advantage of this fact in dealing with non-penetration constraints.

In particular, our solution is the following **two-step approach**:

1. Given a set of PCs between H and F, check the explicit equality and overlap constraints among all the PCs.
   - If there are conflicts in satisfying these constraints, report "the PCs do not form a valid CF" and exit.
   - Otherwise, solve for ${}^FT_H$ based on the constraints.
2. Check if H and F interpenetrate at all the solutions of ${}^FT_H$ derived from Step 1. If yes, report "the PCs do not form a valid CF" and exit. Otherwise, report "the PCs can form a valid CF" and exit.

In this approach, there are several issues we need to address:

(1) How to devise a general algorithm to implement Step 1 for an arbitrary set of given PCs?

This is the focus of this paper. In Sections 4 and 5, we address this issue in detail. Our analyses show that in those cases where ${}^FT_H$ has a finite number of solutions, these solutions can be found from the equality and overlap constraints of just two or three PCs. After ${}^FT_H$ is solved, the validity of constraints from the rest of the PCs can be checked at those locations of ${}^FT_H$.

The cases where ${}^FT_H$ has innumerable solutions include all single-PC cases and some cases with two or more PCs (Fig. 6a and b show examples of such two-PC cases). For those cases with two or more PCs, we can always find a smaller set of *equivalent* PCs which constrain the objects in the same way as the original set of PCs do. For example, the set of two e-e PCs in Fig. 6a or b constrain the objects in the same way as the upper e-e PC. By recursively replacing the original set of PCs by a smaller set of equivalent PCs, we can eventually find an equivalent set with only one or two PCs. Then, we can check if the equality and overlap constraints of the (equivalent) PC or two PCs are satisfied.

(2) How to detect interpenetration between H and F given a fixed location ${}^FT_H$ in Step 2?

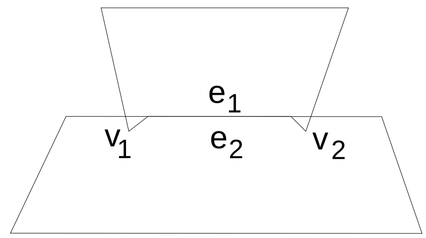

Figure 5: An invalid CF detectable in Step 1 of the *two-step* approach

There exist fast algorithms for checking intersection between two polyhedral objects with a known relative location; see for instance [9, 10, 17].

(3) For cases where ${}^{F}T_{H}$ has innumerable solutions, how to deal with Step 2?

If a set of PCs constrain the relative location between H and F to a region of infinite locations, it means that we cannot find the exact relative location from the contact constraints. Since the exact relative location can only be known to be within some uncertainty neighborhood of infinite locations of an estimated location and we obviously cannot enumerate each location to check the violation of the non-penetration constraint at it, we are unable to report "the PCs definitely do not form a valid CF" *by nature* [2].

Nevertheless, we can report "the PCs can form a valid CF" sometimes. For instance, we can select a few solutions of ${}^{F}T_{H}$ in the uncertainty neighborhood of the estimated location ${}^{F}\hat{T}_{H}$ (Section 2.4), and check if H and F has no interpenetration at, at least, one of those relative locations. If so, we can report "the PCs can form a valid CF".

It is important to emphasize that for many such sets of PCs which constrain ${}^{F}T_{H}$ to innumerable solutions, Step 1 alone is able to report "the PCs definitely do not form a valid CF". For example, for a given set of PCs: $\{v_1\text{-}e_2,\ e_1\text{-}e_2,\ v_2\text{-}e_2\}$ corresponding to the objects shown in Fig. 5, by checking the equality and overlap constraints (as detailed in Section 4), Step 1 will find contradictions in the constraints and report that the PCs do not form a valid CF.

(4) In what cases the weak conclusion "the PCs can form a valid CF" can become the strong one "the PCs definitely form a valid CF"?

From the description of the two-step approach, it is clear that given a set of PCs between H and F, if (a) there is no conflict in satisfying all equality and overlap constraints and ${}^{F}T_{H}$ can be solved (from the constraints) with a finite number of solutions, and (b) H and F *do not* interpenetrate at *all* of these solutions of ${}^{F}T_{H}$, then we can report "the PCs definitely form a valid CF".

[2] Note also that no previous work considered checking non-penetration constraints in such situations.

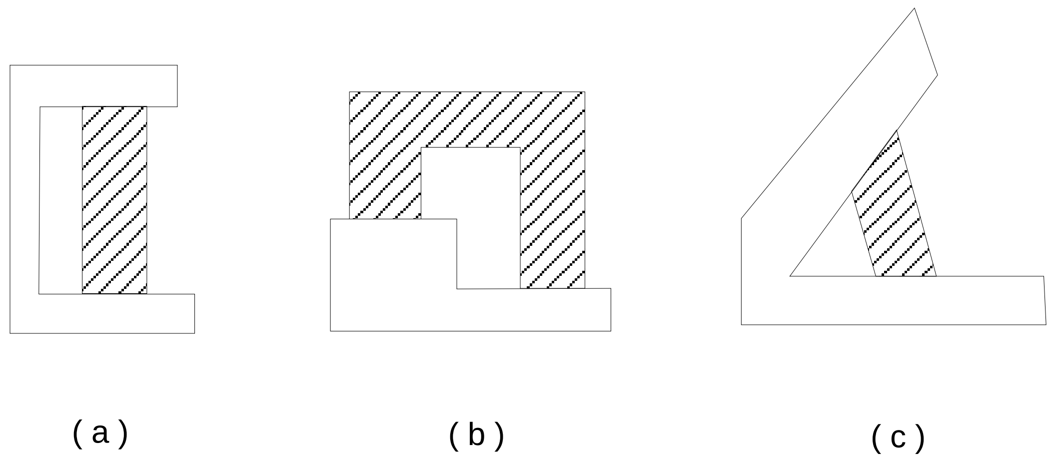

Figure 6: Some CFs of (e-e, e-e) type

Note that even if we only know the estimated location ${}^F\hat{T}_H$ due to location sensing uncertainty, in the above case, since the contact constrains ${}^FT_H$ to a few possible locations, we are able to exhaustively check the validity of the non-penetration constraint to make sure that no matter where ${}^FT_H$ actually is, the PCs always form a valid CF. This process can even be simplified sometimes when we can accurately identify where ${}^FT_H$ actually is by comparing ${}^F\hat{T}_H$ against all the solutions of ${}^FT_H$ under the contact. In such situations, we only need to check if H and F interpenetrate at the actual ${}^FT_H$ to conclude whether the PCs definitely form a valid CF.

Practically, we may not even have to check whether H and F interpenetrate if they are rigid bodies and the "concavity depth" of the objects satisfy certain relationship with the location uncertainty bound. From the context of our problem, which is to recognize the contact formation between two objects *when there is a collision detected between the two*[3], we see that as H and F are both rigid bodies, it is impossible for them to actually interpenetrate upon a collision. Only *virtual* interpenetration may occur at the *estimated* relative location ${}^F\hat{T}_H$ *due to location uncertainties* of H and F. Moreover, whether a PC which causes interpenetration is perceived as a *possible* PC under location uncertainty depends on whether the depths of the contours in H and/or F are small enough to be less than or equal to the displacements caused by uncertainties. For example, in Fig. 4a, the v_1-e_2 PC can be perceived as possible only if the depth of the contour at v_1 is smaller than or equal to the vertical displacement of v_1 caused by location uncertainty. Similar arguments apply to Fig. 4b. Hence, if the contours in H and/or F are deep enough, in the set S_{pc} of possible PCs taking into account location uncertainty (recall Section 1 for introduction), there will be no PC that can cause interpenetration. In such cases, it does not matter whether ${}^FT_H$ has innumerable solutions under the equality and overlap constraints of the PCs: as long as the equality and overlap constraints are satisfied, we can report that the PCs definitely form a valid CF.

[3]Simply detecting if there is a collision or not can be easily accomplished by a force sensor in a guarded motion.

4 Equality and Overlap Constraints

We now describe in detail the equality and overlap constraints imposed by different types of CFs.

4.1 Constraints for Single PCs

For each type of PC (formed between H and F), we can express the equality constraints and overlap constraints (which are inequalities) on the relative location ${}^{F}T_{H}$ as the following:

- for a v^H-v^F type of PC,

$$ {}^{F}T_{H} = {}^{F}T_{v^F} \cdot T_{rot}(\theta) \cdot {}^{H}T_{v^H}^{-1} $$

- for a v^H-e^F type of PC,

$$ \begin{aligned} {}^{F}T_{H} = {}^{F}T_{e^F} \cdot T_{trans}(0, y) \cdot T_{rot}(\theta) \cdot {}^{H}T_{v^H}^{-1} \\ 0 < y < l_{e^F} \end{aligned} $$

- for an e^H-e^F type of PC,

$$ \begin{aligned} {}^{F}T_{H} = {}^{F}T_{e^F} \cdot T_{trans}(0, y) \cdot T_{rot}(\pi) \cdot {}^{H}T_{e^H}^{-1} \\ 0 < y < l_{e^H} + l_{e^F} \end{aligned} $$

- for an e^H-v^F type of PC

$$ \begin{aligned} {}^{F}T_{H} = {}^{F}T_{v^F} \cdot T_{rot}(\theta) \cdot T_{trans}(0, y) \cdot {}^{H}T_{e^H}^{-1} \\ -l_{e^H} < y < 0 \end{aligned} $$

where $T_{trans}(*, *)$ and $T_{rot}(*)$ denote 3×3 homogeneous transformation matrices for translation and rotation respectively, and θ and y are variables. These constraints show that although there are uncertainties in the locations of H and F, if a PC is formed between H and F, the relative location ${}^{F}T_{H}$ has only one or two degrees of freedom. Note that the overlap constraints are the constraints on the ranges of y's. Whereas, the constraints on the ranges of θ's are the non-penetration constraints, which are difficult to formulate in general terms because they depend on specific shapes of H and F (see Fig. 7 for examples). The non-penetration constraints are handled in the second step of the two-step approach presented in Section 3.

4.2 Constraints for CFs with Two PCs

Before we present the constraints and derivation in detail, the following subsection describes how we approach the derivation and defines the kinds of constraints to be expected.

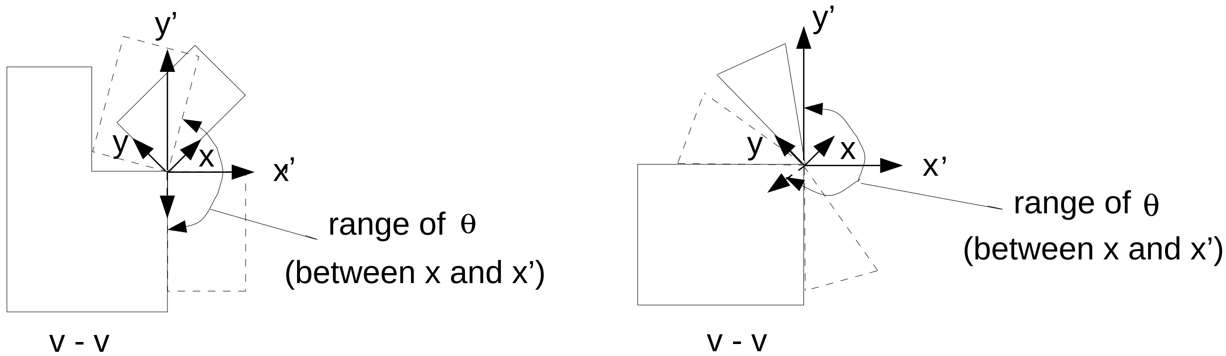

Figure 7: The range of θ depends on object shapes

4.2.1 General analysis

Clearly, the co-existence of two PCs require the right-hand sides (RHS) of their two corresponding equality constraints to be equal, which provide a new 3×3 matrix equation $\mathcal{E}$ in terms of the free variables in the equality constraints of the individual PCs (i.e., the θ's and the y's). For all types of the two-PC CFs, the matrix equation $\mathcal{E}$ has the following common characteristics:

- The last rows on both sides of $\mathcal{E}$ are the same constant vector [0 0 1].
- The six scalar equations of $\mathcal{E}$ are transcendental and not independent.
- $\mathcal{E}$ has at most four variables.

As shown in our detailed derivation later, there are the following types of results from solving $\mathcal{E}$:

1. For some types of two-PC CFs, all the free variables in $\mathcal{E}$ can be solved. Thus, ${}^{F}T_{H}$ has a unique or a finite number of solutions (depending on the number of solutions of the free variables).
2. For some other types of two-PC CFs, some variables in $\mathcal{E}$ can be solved. Thus, an equivalent single PC can be found which constrains ${}^{F}T_{H}$ in the same way as the two-PC CF.
3. For the remaining types of two-PC CFs, some variables can be represented as functions of some other variables. However, none of the variables can be fully solved.

Replacing the variables in the overlap constraints of the individual PCs in a two-PC CF by the expressions of these variables from solving $\mathcal{E}$, we can obtain a set of inequalities among *known parameters only* of the two objects H and F. Moreover, from solving the transcendental equations in $\mathcal{E}$, there will arise additional equality or inequality relationships among known parameters of H and F only. We call such derived relationships among known parameters of the objects under a two-PC CF the *inter-PC constraints*. Because the inter-PC constraints are in terms of known parameters only, they are *the* constraints that we can use to rule out a given two-PC set which is not a valid CF in the first step of the two-step approach (outlined in Section 3).

4.2.2 Theorems for two-PC CFs

There are seven types of two-PC CFs between two polygons as enumerated in Section 2.1. We now present the results of our derivation for these seven types of two-PC CFs in the forms of seven theorems. For the sake of brevity, we present the proofs of Theorem 2 and Theorem 4 in the appendices. The proofs for the rest of the theorems, with similar style, can be found in [22]. For convenience, we use the following additional notations (of known parameters) in our presentation.

$$\begin{aligned} \mathbf{a}^H &= \mathbf{a}_2^H - \mathbf{a}_1^H = [a_x^H \ a_y^H]^t & \qquad \mathbf{a}^F &= \mathbf{a}_2^F - \mathbf{a}_1^F = [a_x^F \ a_y^F]^t \\ \mathbf{b}^H &= \mathbf{b}_2^H - \mathbf{b}_1^H = [b_x^H \ b_y^H]^t & \qquad \mathbf{b}^F &= \mathbf{b}_2^F - \mathbf{b}_1^F = [b_x^F \ b_y^F]^t \\ \mathbf{c}^H &= \mathbf{b}_2^H - \mathbf{a}_1^H = [c_x^H \ c_y^H]^t & \qquad \mathbf{c}^F &= \mathbf{b}_2^F - \mathbf{a}_1^F = [c_x^F \ c_y^F]^t \\ \mathbf{d}^H &= \mathbf{a}_2^H - \mathbf{b}_1^H = [d_x^H \ d_y^H]^t & \qquad \mathbf{d}^F &= \mathbf{a}_2^F - \mathbf{b}_1^F = [d_x^F \ d_y^F]^t \end{aligned}$$

where ^{t} denotes transpose and $\mathbf{a}_i^H, \mathbf{a}_i^F, \mathbf{b}_i^H$, and $\mathbf{b}_i^F$ are known parameters defined in Section 2.3.

Theorem 1 *For the (v_1^H-v_1^F, v_2^H-v_2^F) type of CF, the following inter-PC constraint holds:*

$$d_{v_1^H v_2^H} = d_{v_1^F v_2^F} \neq 0 \tag{1}$$

and there exists a unique ${}^F T_H$*:*

$${}^F T_H = {}^F T_{v_1^F} \cdot T_{rot}(\theta) \cdot {}^H T_{v_1^H}^{-1} \tag{2}$$

where θ satisfies

$$\cos\theta = \frac{a_x^H a_x^F + a_y^H a_y^F}{(a_x^H)^2 + (a_y^H)^2}; \quad \sin\theta = \frac{a_x^H a_y^F - a_y^H a_x^F}{(a_x^H)^2 + (a_y^H)^2} \tag{3}$$

Theorem 2 *For the (v_1^H-v_1^F, e_2^H-e_2^F) type of CF, the following inter-PC constraints hold:*

$$d_{v_1^H e_2^H} = d_{v_1^F e_2^F} \tag{4}$$

$$0 < k < l_{e_2^H} + l_{e_2^F} \tag{5}$$

where

$$k = c_x^H \sin\alpha_2 - c_y^H \cos\alpha_2 + c_x^F \sin\beta_2 - c_y^F \cos\beta_2 \tag{6}$$

and there exists a unique ${}^F T_H$*:*

$${}^F T_H = {}^F T_{e_2^F} \cdot T_{trans}(0, k) \cdot T_{rot}(\pi) \cdot {}^H T_{e_2^H}^{-1}$$

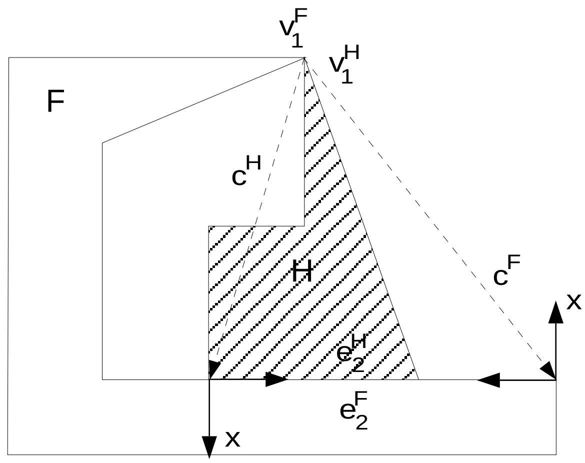

Figure 8: A geometrical illustration of the inter-PC constraint $d_{v_1^H e_2^H} = d_{v_1^F e_2^F}$ for a $(v\text{-}v,\ e\text{-}e)$ CF

Fig. 8 shows an illustration of the inter-PC constraint (4).

Theorem 3 *For the (v_1^H-v_1^F, v_2^H-e_2^F) type of CF, the following inter-PC constraints hold:*

$$d_{v_1^H v_2^H} \geq d_{v_1^F e_2^F} \tag{7}$$
$$0 < k < l_{e_2^F} \tag{8}$$

where

$$k = a_x^H \sin\theta + a_y^H \cos\theta + c_x^F \sin\beta_2 - c_y^F \cos\beta_2 \tag{9}$$

and θ satisfies the following equation

$$\sin(\theta + \phi) = \frac{c_x^F \cos\beta_2 + c_y^F \sin\beta_2}{\sqrt{(a_x^H)^2 + (a_y^H)^2}}, \text{ where } \phi = atan2(a_x^H, -a_y^H), \tag{10}$$

Moreover, ${}^F T_H$, expressed as

$${}^F T_H = {}^F T_{e_2^F} \cdot T_{trans}(0, k) \cdot T_{rot}(\theta) \cdot {}^H T_{v_2^H}^{-1}$$

has at most two solutions, subject to the solutions of θ in (10) and the corresponding values of k subject to the inter-PC constraint (8).

Theorem 4 *For the (e_1^H-e_1^F, e_2^H-e_2^F) type of CF, the following inter-PC constraint holds:*

$$\theta_{e_1^H e_2^H} = \theta_{e_1^F e_2^F} \tag{11}$$

Moreover, there are two cases:

- *If* $e_1^F \not\parallel e_2^F$ *(i.e.,* $\sin(\beta_2 - \beta_1) \neq 0$*), then the following additional inter-PC constraints hold:*

$$0 < k_1 < l_{e_1^H} + l_{e_1^F} \quad (12)$$
$$0 < k_2 < l_{e_2^H} + l_{e_2^F} \quad (13)$$

where

$$k_1 = \frac{[\cos\beta_2 \quad \sin\beta_2]}{\sin(\beta_2 - \beta_1)}(\mathbf{R}(\beta_1 - \alpha_1)\mathbf{b}^H + \mathbf{b}^F) \quad (14)$$
$$k_2 = \frac{[\cos\beta_1 \quad \sin\beta_1]}{\sin(\beta_2 - \beta_1)}(\mathbf{R}(\beta_1 - \alpha_1)\mathbf{b}^H + \mathbf{b}^F) \quad (15)$$

and there exists a unique ${}^F T_H$*:*

$${}^F T_H = {}^F T_{e_1^F} \cdot T_{trans}(0, k_1) \cdot T_{rot}(\pi) \cdot {}^H T_{e_1^H}^{-1} \quad (16)$$

- *If* $e_1^F \parallel e_2^F$ *(see, for example, Fig. 6a and b), then the following additional inter-PC constraints hold:*

$$d_{e_1^H e_2^H} = d_{e_1^F e_2^F} \quad (17)$$

$$\begin{cases} -(l_{e_1^H} + l_{e_1^F}) < k < l_{e_2^H} + l_{e_2^F} & \text{if } \beta_2 = \beta_1 \\ -(l_{e_1^H} + l_{e_1^F} + l_{e_2^H} + l_{e_2^F}) < k < 0 & \text{if } \beta_2 = \beta_1 \pm \pi \end{cases} \quad (18)$$

where

$$k = [\sin\beta_1 \quad -\cos\beta_1](\mathbf{R}(\beta_1 - \alpha_1)\mathbf{b}^H + \mathbf{b}^F) \quad (19)$$

and $\beta_2 \in [0, 2\pi]$*. Moreover, the two-PC CF is equivalent to a single e-e PC, which has the following equality and overlap constraints:*

$${}^F T_H = {}^F T_{e_1^F} \cdot T_{trans}(0, y) \cdot T_{rot}(\pi) \cdot {}^H T_{e_1^H}^{-1} \quad (20)$$

$$\begin{cases} \max(-k, 0) < y < \min(l_{e_2^H} + l_{e_2^F} - k, l_{e_1^H} + l_{e_1^F}) & \text{if } \beta_2 = \beta_1 \\ \max(-k - l_{e_2^H} - l_{e_2^F}, 0) < y < \min(l_{e_1^H} + l_{e_1^F}, -k) & \text{if } \beta_2 = \beta_1 \pm \pi \end{cases} \quad (21)$$

Theorem 5 *For the* $(v_1^H\text{-}e_1^F,\ e_2^H\text{-}e_2^F)$ *type of CF, there are two cases:*

- *If* $e_1^F \not\parallel e_2^F$ *(i.e.,* $\sin(\beta_2 - \beta_1) \neq 0$*), then the following inter-PC constraints hold:*

$$0 < k_1 < l_{e_1^F} \quad (22)$$
$$0 < k_2 < l_{e_2^H} + l_{e_2^F} \quad (23)$$

where

$$k_1 = \frac{[\cos\beta_2 \quad \sin\beta_2]}{\sin(\beta_2 - \beta_1)}(\mathbf{R}(\beta_2 - \alpha_2)\mathbf{c}^H + \mathbf{b}^F) \quad (24)$$
$$k_2 = \frac{[\cos\beta_1 \quad \sin\beta_1]}{\sin(\beta_2 - \beta_1)}(\mathbf{R}(\beta_2 - \alpha_2)\mathbf{c}^H + \mathbf{b}^F) \quad (25)$$

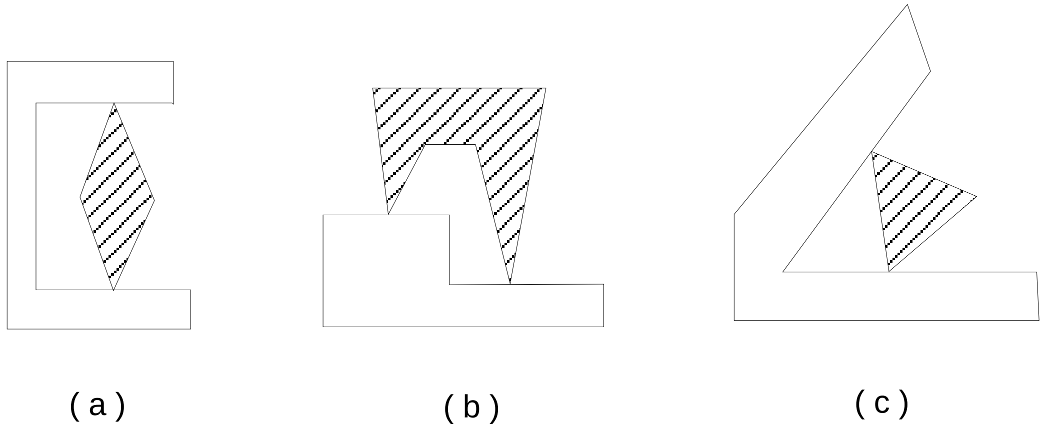

Figure 9: Some CFs of (v-e, v-e) type

and there exists a unique ${}^F T_H$*:*

$$ {}^F T_H = {}^F T_{e_2^F} \cdot T_{trans}(0, k_2) \cdot T_{rot}(\pi) \cdot {}^H T_{e_2^H}^{-1} \tag{26} $$

- *If* $e_1^F \| e_2^F$*, then the following inter-PC constraints hold:*

$$ d_{v_1^H e_2^H} = d_{e_1^F e_2^F} \tag{27} $$

$$ \begin{cases} -l_{e_1^F} < k < l_{e_2^H} + l_{e_2^F} & \textit{if } \beta_1 = \beta_2 \\ 0 < k < l_{e_1^F} + l_{e_2^H} + l_{e_2^F} & \textit{if } \beta_1 = \beta_2 \pm \pi \end{cases} \tag{28} $$

where

$$ k = [\sin\beta_2 \quad -\cos\beta_2](\mathbf{R}(\beta_2 - \alpha_2)\mathbf{c}^H + \mathbf{b}^F) \tag{29} $$

and $\beta_1 \in [0, 2\pi]$*. Moreover, the two-PC CF is equivalent to a single e-e PC, which has the following equality and overlap constraints:*

$$ {}^F T_H = {}^F T_{e_2^F} \cdot T_{trans}(0, y) \cdot T_{rot}(\pi) \cdot {}^H T_{e_2^H}^{-1} \tag{30} $$

$$ \begin{cases} \max(k, 0) < y < \min(l_{e_1^F} + k, l_{e_2^H} + l_{e_2^F}) & \textit{if } \beta_1 = \beta_2 \\ \max(k - l_{e_1^F}, 0) < y < \min(k, l_{e_2^H} + l_{e_2^F}) & \textit{if } \beta_1 = \beta_2 \pm \pi \end{cases} \tag{31} $$

Theorem 6 *For the* (v_1^H-e_1^F, v_2^H-e_2^F) *type of CF, there are two cases:*

- *If* $e_1^F \| e_2^F$ *(e.g., see Fig. 9a and b), then the following inter-PC constraints hold:*

$$ d_{v_1^H v_2^H} \geq d_{e_1^F e_2^F} \tag{32} $$

$$\begin{cases} -l_{e_1^F} < k < l_{e_2^F} & \textit{if } \beta_2 = \beta_1 \\ -(l_{e_1^F}+l_{e_2^F}) < k < 0 & \textit{if } \beta_2 = \beta_1 \pm \pi \end{cases} \tag{33}$$

where $\beta_2 \in [0, 2\pi]$, *and*

$$k = [-\sin\beta_1 \quad \cos\beta_1](\mathbf{R}(\beta_1+\theta)\mathbf{a}^H - \mathbf{b}^F) \tag{34}$$

has two possible values subject to the two solutions of θ *satisfying the following equation:*

$$\sin(\theta+\phi) = \frac{b_x^H \cos\beta_1 + b_y^H \sin\beta_1}{\sqrt{(a_x^H)^2 + (a_y^H)^2}}, \textit{ where } \phi = \textit{atan2}(a_x^H, -a_y^H) \tag{35}$$

Consequently, the two-PC CF can be equivalent to two possible e-e PCs, whose equality and overlap constraints depend on the two possible values of θ *and the corresponding values of* k*, as expressed below:*

$$^FT_H = {}^FT_{e_1^F} \cdot T_{trans}(0, y) \cdot T_{rot}(\pi) \cdot ({}^HT_{v_1^H} \cdot T_{rot}(\pi - \theta))^{-1} \tag{36}$$

$$\begin{cases} \max(-k, 0) < y < \min(l_{e_1^F}, l_{e_2^F} - k) & \textit{if } \beta_2 = \beta_1 \\ \max(-k - l_{e_2^F}, 0) < y < \min(-k, l_{e_1^F}) & \textit{if } \beta_2 = \beta_1 \pm \pi \end{cases} \tag{37}$$

- *If* $e_1^F \not\parallel e_2^F$, *i.e.,* $\sin(\beta_2 - \beta_1) \neq 0$ *(e.g., see Fig. 9c), then the following inter-PC constraints hold:*

$$-|k_0| < k_1 < |k_0| + l_{e_1^F} \tag{38}$$
$$-|k_0| < k_2 < |k_0| + l_{e_2^F} \tag{39}$$

where

$$k_0 = \|\mathbf{a}^H\| / \sin(\beta_1 - \beta_2) \tag{40}$$
$$k_1 = (b_x^F \cos\beta_2 + b_y^F \sin\beta_2)/\sin(\beta_2 - \beta_1) \tag{41}$$
$$k_2 = (b_x^F \cos\beta_1 + b_y^F \sin\beta_1)/\sin(\beta_2 - \beta_1) \tag{42}$$

Moreover, ^FT_H *can be expressed as*

$$^FT_H = {}^FT_{e_1^F} \cdot T_{trans}(0, k(\theta)) \cdot T_{rot}(\theta) \cdot {}^HT_{v_1^H}^{-1} \tag{43}$$

where

$$k(\theta) = k_0 \sin(\theta + \phi + \beta_1 - \beta_2) + k_1$$

and θ *is a free variable in the range* $[0, 2\pi]$. *The two-PC CF in this case cannot be equivalent to a single PC.*

Cases			Inter-PC Constraints	
$k_1 = k_2 = 0$			$0 < m_2 - m_1 < l_{e_2^F} + l_{e_1^H}$	*
			$-l_{e_1^H} < m_1 + m_2 < l_{e_2^F}$	*
$k_1 = k_2 \neq 0$			$0 < m_2 - m_1 < l_{e_2^F} + l_{e_1^H}$	*
			$m_1 < 0$ and $m_2 > 0$	
			$m_1 > -l_{e_1^H}$ and $m_2 < l_{e_2^F}$	
$k_1 = -k_2 \neq 0$			$-l_{e_1^H} < m_1 + m_2 < l_{e_2^F}$	*
			$m_1 > -l_{e_1^H}$ and $m_2 > 0$	
			$m_1 < 0$ and $m_2 < l_{e_2^F}$	
$\lvert k_1 \rvert \neq \lvert k_2 \rvert$	$\lvert k_1 \rvert < \lvert k_2 \rvert$	$k_2 > 0$	$f_1(\gamma_1) > -l_{e_1^H}$	
			$f_1(\gamma_2) < 0$	
		$k_2 < 0$	$f_1(\gamma_1) < 0$	
			$f_1(\gamma_2) > -l_{e_1^H}$	
	$\lvert k_1 \rvert > \lvert k_2 \rvert$	$k_1 > 0$	$f_2(\gamma_3) < l_{e_2^F}$	
			$f_2(\gamma_4) > 0$	
		$k_1 < 0$	$f_2(\gamma_3) > 0$	
			$f_2(\gamma_4) < l_{e_2^F}$	

where

$$k_1 = d_x^H \cos\alpha_1 + d_y^H \sin\alpha_1$$

$$k_2 = c_x^F \cos\beta_2 + c_y^F \sin\beta_2$$

$$m_1 = d_x^H \sin\alpha_1 - d_y^H \cos\alpha_1$$

$$m_2 = c_x^F \sin\beta_2 - c_y^F \cos\beta_2$$

$$f_1(\theta) = \frac{d_x^H \cos\theta - d_y^H \sin\theta - k_2}{\sin(\theta + \alpha_1)}$$

$$\gamma_1 = \arccos(k_1/k_2) - \alpha_1$$

$$\gamma_2 = 2\pi - \arccos(k_1/k_2) - \alpha_1$$

$$f_2(\theta) = \frac{k_1 - c_x^F \cos\theta - c_y^F \sin\theta}{\sin(\theta - \beta_2)}$$

$$\gamma_3 = \arccos(k_2/k_1) + \beta_2$$

$$\gamma_4 = 2\pi - \arccos(k_2/k_1) + \beta_2$$

Table 1: Inter-PC Constraints of the *(e-v, v-e)* CF

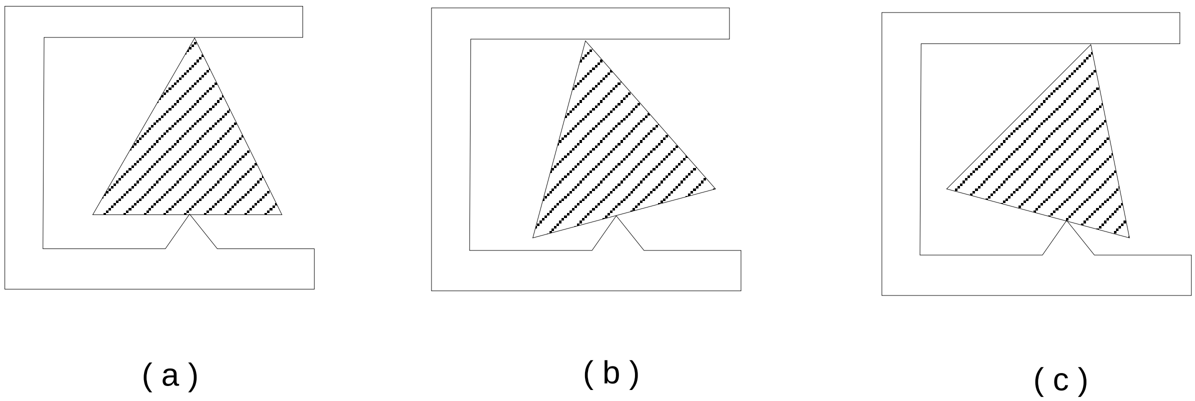

Figure 10: Three configurations of an (e-v, v-e) CF with $k_1 = k_2 \neq 0$

Theorem 7 *For the (e_1^H-v_1^F, v_2^H-e_2^F) type of CF, at least one inter-PC constraint in Table 1 holds, depending on the values of k_1 and k_2 (whose expressions are also in Table 1). The ${}^FT_H$ can be generally expressed as*

$$^FT_H = {}^FT_{v_1^F} \cdot T_{rot}(\theta) \cdot T_{trans}(0, y) \cdot {}^HT_{e_1^H}^{-1} \tag{44}$$

where, depending on the values of k_1 and k_2 and the corresponding inter-PC constraints, ${}^FT_H$ becomes either a function of only one variable y or a function of one variable θ. However, this type of two-PC CF cannot be equivalent to a single PC (Fig. 10 shows an example).

4.2.3 Summary

The above theorems demonstrate the following.

- Except for
  - **type 1** CFs: the (v-e, v-e) type of CFs where the two edges are not parallel, and
  - **type 2** CFs: the (e-v, v-e) type of CFs,

  the rest of the two-PC CFs either constrain ${}^FT_H$ to one or two relative locations or are equivalent to some single-PCs.

- For a two-PC CF which constrains ${}^FT_H$ to one or two locations, the inter-PC constraints are the necessary and sufficient conditions to satisfy the equality and overlap constraints of both PCs in the CF.

- For a two-PC CF which is equivalent to a single-PC, the inter-PC constraints together with the equality and overlap constraints of the equivalent PC constitute the necessary and sufficient conditions to satisfy the equality and overlap constraints of both original PCs in the CF.

- For **type 1** and **type 2** CFs, the inter-PC constraints given are only necessary conditions, since they are given by considering that the range of variable θ spans 2π, which is usually larger than the actual range of θ. As explained in Section 4.1, the actual range of θ, which should satisfy the non-penetration constraint, is difficult to formulate in general terms because it depends on the specific shapes of objects H and F.

These results are important for analyzing a general CF with three or more PCs and ultimately for implementing Step 1 of the two-step approach described in Section 3.

4.3 Analyses of CFs with Three or More PCs

The following two types of three-PC CFs which contain the **type 1** and/or the **type 2** CFs as subsets deserve special attention:

- **type 3** CF — (v-e, v-e, v-e), where the three edges are not parallel
- **type 4** CF — (e-v, v-e, v-e), where the edges in (v-e, v-e) are not parallel

because these CFs sometimes do constrain ${}^F T_H$ to only a few solutions. In fact, by similar derivations as presented in Section 4.2, we can achieve the following results:

- For a **type 3** CF, depending on the parameter values of the contact elements, either it is equivalent to a **type 1** CF, or it yields at most two solutions for ${}^F T_H$.
- A **type 4** CF gives at most four solutions for ${}^F T_H$.

For the sake of brevity, we do not present the detailed derivations here, which can be found in [22].

In general, for a many-PC CF, i.e., one with three or more PCs, based on the types of the two-PC or three-PC subsets in it, there exist the following cases:

- ${}^F T_H$ can be solved from a two-PC subset of the CF with one or two solutions.
- ${}^F T_H$ can be solved from a **type 3** three-PC subset with at most two solutions.
- ${}^F T_H$ can be solved from a **type 4** three-PC subset with at most four solutions.
- The many-PC CF can be recursively reduced to be equivalent to an e-e PC or a **type 1** CF.

The above facts enable us to implement the two-step approach outlined in Section 3 in the next section.

5 Verification

Now we are ready to present a general algorithm for verifying whether a given set of PCs can constitute a geometrically valid CF.

Algorithm 1 *Identifying CFs in 2D space*
Input: $S = \{PC_i,\ i = 1, 2, \cdots, n\}$.
1. If S *has only one PC, call* **Check 1** *and go to 9.*
2. If S *is a two-PC set* (v-e, v-e) *where the* e*'s are not parallel, then call* **Check 2***; go to 9.*
3. If S *is a two-PC set* (e-v, v-e)*, call* **Check 2***; go to 9.*
4. If S *has a two-PC subset which has a* v-v *PC, call* **Check 3** *and go to 9.*
5. If S *has a two-PC subset which has an* (e-e) *PC, then*

- *if the* e*'s from the same object are parallel, then*
  - *if the inter-PC constraints of the CF corresponding to the two-PC subset are not satisfied, go to 10;*
  - *otherwise, replace*[4] *the two-PC subset by its equivalent* e-e *PC and go to 1;*
- *otherwise, call* **Check 3** *with this subset; go to 9.*

6. If S *has a two-PC subset* (v-e, v-e) *where the* e*'s are parallel, then*

- *if the inter-PC constraints of the corresponding* (v-e, v-e) *CF are not satisfied, go to 10;*
- *otherwise, for each of the* (e-e) *PCs which are equivalent to it, replace the subset with the equivalent PC and go to 1;*

7. If S *has a* (v-e, v-e, v-e) *subset, then*

- *if the inter-PC constraints of the* **type 3** *CF are not satisfied, go to 10;*
- *otherwise,*
  - *if the subset is equivalent to a* (v-e, v-e) *two-PC set, then replace it by the equivalent* (v-e, v-e) *set and go to 2;*
  - *otherwise, call* **Check 3** *with the subset* (v-e, v-e, v-e)*; go to 9.*

8. If S *has an* (e-v, v-e, v-e) *subset, then*

- *if the inter-PC constraints of the* **type 4** *CF are not satisfied, go to 10;*
- *otherwise, call* **Check 3** *with the subset* (e-v, v-e, v-e).

[4]The replacement should be done *together* with the equality and overlap constraints of the equivalent PC. This explanation applies to the rest of replacements as well.

9. Report result and exit.
10. Report "Cannot form a valid CF" and exit.

The algorithm **Check 1** is outlined below.

Algorithm 2 *Check 1*
Input: a single PC
1. Select a number of relative locations in a neighborhood of ${}^F\hat{T}_H$ (based on location uncertainty), which satisfy the equality and overlap constraint of the PC.
2. Check if H and F interpenetrate at all locations. If so, report "Uncertain case"; else, report "S can form a valid CF". Return.

The algorithm **Check 2** is outlined below:

Algorithm 3 *Check 2*
Input: a two-PC set of type (v-e, v-e) *or type* (e-v, v-e)
1. If the inter-PC constraints of the corresponding CF are not satisfied, report "S cannot form a valid CF" and return.
2. Select a number of relative locations in a neighborhood of ${}^F\hat{T}_H$ which satisfy the constraints in Step 1 and check if H and F interpenetrate at all locations. If so, report "Uncertain case"; else, report "S can form a valid CF". Return.

The algorithm **Check 3** is outlined below.

Algorithm 4 *Check 3*
Input:
(a) a set S of PCs, and
(b) a two-PC or three-PC subset S_b of S which constrains ${}^FT_H$ to a few solutions
1. Check if S_b satisfies the inter-PC constraints of the corresponding CFs. If not, go to 7.
2. Derive the relative locations, i.e., the solutions of ${}^FT_H$, from S_b.
3. If there exist PCs in $S - S_b$ violating their respective equality and/or overlap constraints at every solution of ${}^FT_H$, go to 7.
4. If the equality and overlap constraints of all PCs in $S - S_b$ are satisfied only at some solutions of ${}^FT_H$, go to 6.
5. If H and F do not interpenetrate at all solutions of ${}^FT_H$, report "S definitely forms a CF" and return.
6. If H and F do not interpenetrate at, at least, one of those solutions of ${}^FT_H$ which satisfy the constraints in Step 4, report "S can form a CF" and return.
7. Report "S cannot form a CF" and return.

Note that the algorithm **Check 3** is able to report (back to the main algorithm) the strong answer "S definitely forms a CF". With the information of the objects' shapes fed in, however, the algorithms **Check 1** and **Check 2** can also reach this strong answer in many cases. The reason is summarized in Section 4.2.3 and in Section 3.

6 Conclusions

We reported a study on constraints imposed by arbitrary CFs between two arbitrary polygons. Based on the general, complete, and precise analysis of the constraints, we developed a general algorithm to verify, in the presence of location uncertainties of the contacting objects, if a given set of PCs (contact primitives) can form a geometrically valid CF. The time complexity of the algorithm is linear to the number of given PCs. This complete analysis of contact constraints is not only useful for automatic recognition of contacts but may also contribute to grasp planning. Future directions may include considering modeling uncertainties of objects and extending this analysis to 3D polyhedral objects.

References

[1] A.P. Ambler and R.J. Popplestone, "Inferring the Positions of Bodies from Specified Spatial Relationships," *Artificial Intelligence*, 6(2), pp. 157-174, 1975.

[2] R.C. Brost, *Analysis and Planning of Planar Manipulation Tasks*, PhD thesis, Carnegie Mellon University, 1991.

[3] R. Desai, J. Xiao and R. Volz, "Contact Formations and Design Constraints: A New Basis for the Automatic Generation of Robot Programs", *NATO ARW: CAD Based Programming for Sensor Based Robots*, B. Ravani, Ed., Springer-Verlag, pp. 361-395, 1988.

[4] R. Desai, *On Fine Motion in Mechanical Assembly in Presence of Uncertainty*, PhD thesis, the Univ. of Michigan, 1989.

[5] R. Desai and R. Volz, "Identification and Verification of Termination Conditions in Fine Motion in Presence of Sensor Errors and Geometric Uncertainties," Proc. IEEE Int. Conf. Robotics & Automation, pp. 800-807, May 1989.

[6] B.R. Donald, "On Motion Planning with Six Degree of Freedom: Solving the Intersection Problems in Configuration Space," Proc. IEEE Inter. Conf. Robotics & Automation, pp. 536-541, 1985.

[7] A.O. Farahat, P.F. Stiller, and J.C. Trinkle, "On the Algebraic Geometry of Contact Formation Cells for Systems of Polygons," *IEEE Trans. Robotics & Automation*, 11(4), pp. 522-536, August 1995.

[8] A.O. Farahat, B.S. Graves, and J.C. Trinkle, "Identifying Contact Formations in the Presence of Uncertainty," submitted to *IEEE Trans. Robotics & Automation.*

[9] E.G. Gilbert, D.W. Johnson, and S.S. Keerth, "A Fast Procedure for Computing the Distance between Complex Objects in Three-dimensional Space," *IEEE J. of Robotics and Automation*, 4(2), pp. 193-203, April 1988.

[10] E.G. Gilbert and C-P. Foo, "Computing the Distance Between General Convex Objects in Three-Dimensional Space," *IEEE Trans. Robotics & Automation*, 6(1), pp. 53-61, February 1990.

[11] T. Lozano-Pérez, "The Design of a Mechanical Assembly System," Tech. Report, AI-TR-397, MIT, 1976.

[12] T. Lozano-Pérez, "Spatial Planning: A Configuration Space Approach," *IEEE Trans. Computers*, C-32(2), pp. 108-120, 1983.

[13] R.J. Popplestone, A.P. Ambler, and I. Bellos, "An Interpreter for a Language for Describing Assemblies," *Artificial Intelligence*, 14, pp. 79-107, 1980.

[14] M. Spreng, "Dealing with Unexpected Situations during the Execution of Robot Motions," Proc. IEEE Int. Conf. Robotics & Automation, Sacramento, CA, pp. 64-69, April 1991.

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

[16] R.H. Taylor, *The Synthesis of Manipulator Control Programs from Task-Level Specifications*, PhD thesis, Stanford University, 1976.

[17] F. Thomas and C. Torras, "Interference Detection between Non-Convex Polyhedra Revisited with a Practical Aim," Proc. IEEE Int. Conf. Robotics & Automation, San Diego, pp. 587-594, May 1994.

[18] J. Xiao, "Automatic Determination of Topological Contacts in the Presence of Sensing Uncertainties," Proc. IEEE Int. Conf. Robotics & Automation, Atlanta, pp. 65-70, May 1993.

[19] J. Xiao and M. Su "Automatic Verification of Topological Contacts by Vision," Proc. IEEE Int. Sym. Intelligent Cont., Chicago, pp. 314-319, August 1993.

[20] J. Xiao, "Towards Obtaining All Possible Contacts - Growing a Polyhedron by Its Location Uncertainty," Proc. IEEE/RSJ/GI Int. Conf. Intelli. Robots & Sys., Munich, Germany, pp. 1788-1795, September 1994. An extended version is in *IEEE Trans. Robotics & Automation*, 12(4), pp. 553-565, August, 1996.

[21] J. Xiao and L. Zhang, "An Efficient Algorithm (FAPRIC) for Finding the Principal Contacts Possibly Established due to Uncertainties," Proc. IEEE Int. Conf. Robotics & Automation, Nagoya, Japan, pp. 427-432, May 1995.

[22] J. Xiao and L. Zhang, "A General Algorithm to Determine Geometrically Valid Contact Formations from Possible Contact Primitives," Tech. Report, Dept. of Computer Science, Univ. of North Carolina - Charlotte, Oct. 1994.

A Proof of Theorem 2

Proof: The two PCs in the (v_1^H-v_1^F, e_2^H-e_2^F) CF give the following constraints:

$$^FT_H = {}^FT_{v_1^F} \cdot T_{rot}(\theta) \cdot {}^HT_{v_1^H}^{-1} \tag{45}$$

$$^FT_H = {}^FT_{e_2^F} \cdot T_{trans}(0, y) \cdot T_{rot}(\pi) \cdot {}^HT_{e_2^H}^{-1} \tag{46}$$

$$0 < y < l_{e_2^H} + l_{e_2^F} \tag{47}$$

Equating the RHS of equations (45) and (46) gives

$$\begin{bmatrix} \mathbf{R}(\theta) & -\mathbf{R}(\theta)\mathbf{a}_1^H + \mathbf{a}_1^F \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -\mathbf{R}(\beta_2)\mathbf{R}(-\alpha_2) & \mathbf{R}(\beta_2)\mathbf{R}(-\alpha_2)\mathbf{b}_2^H + R(\beta_2)\mathbf{p} + \mathbf{b}_2^F \\ 0 & 1 \end{bmatrix} \tag{48}$$

where $\mathbf{p} = [0\ y]^t$. From Equation (48), we can solve for θ, which is,

$$\theta = \beta_2 - \alpha_2 \pm \pi, \ (\theta \in [0, 2\pi])$$

Substituting the above solution for θ in (48), we can obtain the following equation for y:

$$y \begin{bmatrix} \sin\beta_2 \\ -\cos\beta_2 \end{bmatrix} = \mathbf{R}(\beta_2 - \alpha_2)\mathbf{c}^H + \mathbf{c}^F \tag{49}$$

By premultiplying $[\sin\beta_2\ \ -\cos\beta_2]$ to both sides of Equation (49) yields the solution: $y = k$ where k is as expressed in Equation (6), which, together with Equation (46), provides the expression of ^FT_H. Substituting k for y in (47) yields the inter-PC constraint (5).

By premultiplying $[\cos\beta_2\ \ \sin\beta_2]$ to both sides of Equation (49) yields the following:

$$\begin{bmatrix} \cos\alpha_2 & \sin\alpha_2 \end{bmatrix} \begin{bmatrix} c_x^H \\ c_y^H \end{bmatrix} + \begin{bmatrix} \cos\beta_2 & \sin\beta_2 \end{bmatrix} \begin{bmatrix} c_x^F \\ c_y^F \end{bmatrix} = 0 \tag{50}$$

Now we show that the above equation is equivalent to the inter-PC constraint (4). The first and second terms of the above equation are the projections of $\mathbf{c}^H$ and $\mathbf{c}^F$ on the OX axes of e_2^H and e_2^F respectively. The projections have different signs since the OX axes of e_2^H and e_2^F point to opposite directions due to the e_2^H-e_2^F PC, and the absolute values of the projections are exactly $d_{v_1^H e_2^H}$ and $d_{v_1^F e_2^F}$ respectively. Thus, the above equation can be rewritten as $d_{v_1^H e_2^H} - d_{v_1^F e_2^F} = 0$, which is the inter-PC constraint (4). Fig. 8 shows an illustration. ▮

B Proof of Theorem 4

Proof: The two PCs in the (e_1^H-e_1^F, e_2^H-e_2^F) CF give the following constraints:

$$^FT_H = {}^FT_{e_1^F} \cdot T_{trans}(0, y_1) \cdot T_{rot}(\pi) \cdot {}^HT_{e_1^H}^{-1} \tag{51}$$

$$0 < y_1 < l_{e_1^H} + l_{e_1^F} \tag{52}$$

$$^F T_H = ^F T_{e_2^F} \cdot T_{trans}(0, y_2) \cdot T_{rot}(\pi) \cdot ^H T_{e_2^H}^{-1} \tag{53}$$

$$0 < y_2 < l_{e_2^H} + l_{e_2^F} \tag{54}$$

By equating the RHS of the equations (51) and (53) and undergoing similar derivations as in the proof of Theorem 2, we have

$$\alpha_2 - \alpha_1 = \beta_2 - \beta_1$$

which is the inter-PC constraint equation (11), and subsequently, we obtain

$$\begin{bmatrix} -\sin\beta_1 & \sin\beta_2 \\ \cos\beta_1 & -\cos\beta_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{R}(\beta_1 - \alpha_1)\mathbf{b}^H + \mathbf{b}^F \tag{55}$$

In solving Equation (55), the following cases are encountered.

• If $\sin(\beta_2 - \beta_1) \neq 0$, which means $e_1^F \not\parallel e_2^F$, then Equation (55) has a unique set of solutions k_1 and k_2 for y_1 and y_2, expressed in (14) and (15). Thus, $^F T_H$ has a unique solution (16), and the two inter-PC constraints (12) and (13) can be obtained by substituting k_1 and k_2 for y_1 and y_2 in (52) and (54).

• If $\beta_2 = \beta_1$ or $\beta_1 \pm \pi, \beta_2 \in [0, 2\pi]$, which means $e_1^F \| e_2^F$, then replacing β_2 by β_1 and by $\beta_1 \pm \pi$ gives

$$k \begin{bmatrix} \sin\beta_1 \\ -\cos\beta_1 \end{bmatrix} = \mathbf{R}(\beta_1 - \alpha_1)\mathbf{b}^H + \mathbf{b}^F \tag{56}$$

where

$$y_2 = \begin{cases} y_1 + k & \text{if } \beta_2 = \beta_1 \\ -y_1 - k & \text{if } \beta_2 = \beta_1 \pm \pi \end{cases} \tag{57}$$

Premultiplying the two sides of Equation (56) by $[\cos\beta_1 \quad \sin\beta_1]$ gives the following equation

$$\begin{bmatrix} \cos\alpha_1 & \sin\alpha_1 \end{bmatrix} \begin{bmatrix} b_x^H \\ b_y^H \end{bmatrix} + \begin{bmatrix} \cos\beta_1 & \sin\beta_1 \end{bmatrix} \begin{bmatrix} b_x^F \\ b_y^F \end{bmatrix} = 0$$

which can be shown, by similar analysis as in the proof of Theorem 2, to be equivalent to the inter-PC constraint equation (17).

Premultiplying the two sides of Equation (56) by $[\sin\beta_1 \quad -\cos\beta_1]$ gives (19), the expression of k.

From Equation (57) and the inequalities (52) and (54), the inter-PC constraint (18) can be derived.

From (57), (52) and (54), and substituting y for y_1, the constraint (21) can be derived. Substituting y for y_1 in (51) accordingly gives (20). (20) and (21) represent the equality and overlap constraints of a single e-e PC, which means that in this case, the *(e-e, e-e)* CF is equivalent to a single e-e PC. ∎