Chapter 2

Passive Assembly – Assembly Sequence Optimization

Assembly sequence optimization is a very important step in assembly automation. It is possible to reduce assembly time and cost by optimizing assembly sequences. Existing heuristics for optimizing assembly sequences (discussed in section 1.2) merely yield a set of feasible sequences for any particular set of parts to be assembled. Hence the user has to make a 'choice' and pick the final assembly sequence. A heuristic that gives a single optimum assembly sequence (if one exists) is, therefore, helpful in determining the assembly sequence that reduces assembly time and cost. In an attempt to further reduce assembly time, it is necessary to avoid unnecessary indexing of the base fixture or re-orienting parts. Hence, another objective here is to categorize parts/subassemblies based on their preferred directions of assembly.

A heuristic is presented in this chapter to find a single optimum assembly sequence for a set of components to be assembled. As mentioned in chapter 1, the heuristic presented here is an extension of the liaison technique presented by Abell et. al. (Abell et. al., 1991). The heuristic is described and followed by an example to demonstrate its working in the following sections.

2.1 Heuristic

The heuristic below gives an optimal assembly sequence if one exists. We follow the liaison technique to enumerate all the feasible assembly sequences and prune the solution tree by applying constraints based on verifiable assembly IDs. We first outline the heuristic and explain it in detail in the following section with an example. We present a few definitions before describing the heuristic.

2.1.1 Definitions

A *liaison* is defined as a “close bond or connection” between two parts and generally includes physical contact between parts. A liaison exists between two parts if the two parts may be assembled together alone.

A *liaison diagram* is a network of all the liaisons that describe the assembly (Figure 2.2).

Precedence constraints are constraints applied to the liaison process that ensure that infeasible assembly sequences are avoided i.e., avoiding certain sequences that render successful assembly impossible.

User-defined constraints are constraints that are applied by the user to avoid certain awkward assembly moves (which necessitate the use of an extra fixture or other tooling). The application of these constraints helps in pruning the assembly tree.

Assembly ID (assembly index of difficulty) includes the overall time required for an assembly operation. This includes the acquisition time (acquisition ID), re-orientation time (orientation ID) and part insertion time. In our passive assembly environment, all of the above are very important and contribute significantly to the overall assembly time. In the example presented below, however, the components are assumed to be pre-oriented and hence obviate the need to compute orientation ID. As mentioned above, Hunt and Sturges have presented comprehensive work in assembly acquisition and part layout at a workstation. Their results give us an optimum parts layout. Thus, we are now only concerned with part insertion. The assembly ID is calculated using the Design for Assembly calculator (DFA calculator) (Sturges, 1989). The DFA calculator is based on dexterity theory and is designed for a manual assembler but with appropriate scaling can also be used for an automatic assembler. According to the Hitachi AEM method of T-downs, vertically downward assembly is the best for most of the components. Hence, it is quite intuitive to the user that the sequence, which allows the assembly of most components vertically downwards, is the best sequence. This may or may not be the case.

2.1.2 Algorithm to find an optimum assembly sequence

The algorithm to compute the optimum assembly sequence for a given assembly under given constraints is as follows:

1. Represent the assembly as a network of nodes and lines. A node bearing the name of the part represents each part. Liaisons, as relationships between parts, are represented as numbered lines connecting related parts (Figure 2.2).
2. Apply the precedence constraints for the assembly.
3. Generate all the feasible assembly sequences for the applied set of precedence constraints and represent it graphically (Figure 2.3).
4. Apply the user-specified constraints to prune the assembly tree (Figure 2.4).

5. For each of the remaining feasible assembly sequences (Table 2.2), calculate the ID for each part and each orientation of the fixture into which it can be assembled.
6. Calculate the cumulative assembly IDs for each sequence and each orientation of each of the components.

The sequence with the least ID is the best sequence possible.

If there is a situation in which IDs of two or more sequences is equal, the sequence with the lower number of indexing positions of the fixture is the best sequence.

2.1.3 Design of the End Effector

The function of an end effector in assembly automation is crucial. The end effector chosen for a particular operation must be able to provide a stable grasp on the component being handled. For a stable grasp, the geometry of the fingers providing the grasp and force applied on the component are very important. Hence, the best method to provide a stable grasp would be to use a unique grasping configuration for each component i.e., use a different gripper for each component. However, this involves many indexing sequences that add to the assembly time and cost. The solution would be to find a 'single' configuration of fingers so that most of the components (if not all) of the assembly can be assembled.

The optimum assembly sequence derived by the above heuristic also gives the orientation in which each component should be assembled. Given these orientations, the features of the components that can be utilized for grasping and successful assembly are cataloged. All the grasping configurations required for grasping the cataloged features are then noted. These grasping configurations are then scanned to find a unique geometry that is able to grasp all the components involved. The dimensions of the fingers are found by taking into account the various constraints (spatial and functional) for successful assembly.

This search process may not be able to determine a ‘single’ gripper for all the operations because of the variety of parts involved. In the example shown below, however, it was possible to find a union of the grasping configurations that satisfy the grasping requirements for all the components (figure 2.5). A simple force analysis is then performed to choose the best gripper.

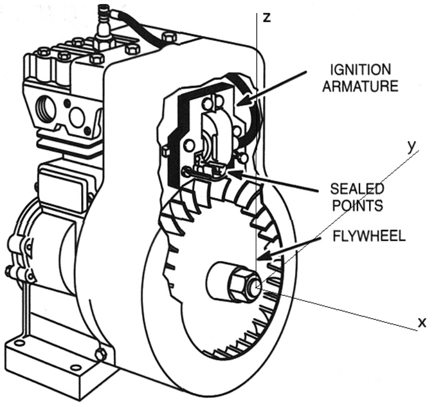
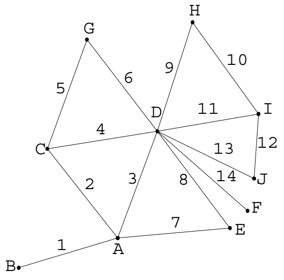


Figure (2.1) – the engine in the 90^0 position; CW and CCW rotations about the y-axis give the 0^0 and 180^0 positions respectively.

2.2 Development of an example

An example representing the assembly of a 2HP Briggs and Stratton engine (Figure 2.1) is developed to demonstrate the heuristic described above. Ten parts of the engine were chosen to represent a good variety of parts in terms of difficulty of assembly (Figure 2.2 - It must be noted here that the piston is meant to represent the piston-connecting rod subassembly). Since the principal assembly directions of all these parts lie in a two-dimensional cartesian space, a fixture that can be indexed to three positions - 0^0, 90^0, 180^0 - is assumed. Figure (2.1) shows the block in the 90^0 position; the 0^0 and 180^0 positions of the block are obtained by rotating the engine by 90^0 clockwise and counter-clockwise respectively about the y-axis. To avoid numerous repositionings and to assure maximum part stability, the engine block is held by the fixture and all the other components are then assembled into it.

A – block; B – valves; C – piston; D – crankshaft;
E – camshaft; F –lifters; G – head; H – key;
I – flywheel; J – clutch housing.



Note: the 'piston' represents the piston + connecting rod subassembly.

Figure (2.2) – the liaison diagram for the ten parts listed above.

2.2.1 Liaison Technique

The constraints listed in Table (2.1) are necessary for proper assembly of the engine. For example, consider the constraint 3→2; which forces the assembly of the crankshaft into the block before the piston (i.e., piston-connecting rod subassembly) is assembled. This is necessary because the installed piston interferes with the crankshaft assembly.

With these precedence constraints determined, an initial set of sequences is derived (Figure 2.3). This set is exhaustive and covers all the possible assembly sequences for the given constraints. To reduce the set of feasible assembly sequences, we apply the following constraints. These constraints also ensure that certain unstable assembly sequences are avoided.

1. Assemble the valves before the lifters for ease of assembly (in the Briggs and Stratton engine used in the example, the valve springs need not be compressed to assemble the camshaft, if the valves are assembled before the lifters). This gives more space in the valve chamber for easier manipulation of the valve spring and the retainer. The effort reduction due to increased maneuvering space can also be calculated as an ID related path. Here we simply make the intelligent choice.
2. Assemble the camshaft immediately after the lifters. This constraint is applied because the lifters are not fastened and are supported only by the camshaft. Hence, any assembly move that is made after the assembly of the lifters and before that of the camshaft renders the lifters unstable. This instability can also be represented as an ID, but only imprecisely according to present dexterity theory.
3. Assemble the head immediately after the piston is assembled. The piston-connecting rod subassembly has one degree of freedom and the connecting rod is free to move about that axis. In order to ensure stability, it is necessary that the connecting rod be immobilized before assembling any other component.

4. We also discard all sequences in which unstable assemblies are achieved outside the engine block. For example, consider the sequence that permits the assembly of the flywheel and the clutch housing outside the block. A fixture is needed to hold this subassembly before assembling it into the block. The cost of such a fixture could be justified if the aggregate ID reduction and number of units processed were large enough.

Figure (2.4) shows the remaining feasible assembly sequences after applying all the constraints. This is a much-reduced set. We can still apply certain constraints on this set to reduce it further. The constraints are as follows:

1. Assemble the camshaft after the crankshaft since timing marks on the camshaft and the crankshaft should match.
2. Assemble the piston before the camshaft, since the alternative will necessitate re-orientation of the crankshaft for which one needs feedback systems. Since we are dealing with passive assembly, this no-cost decision helps to prune the tree further.
3. We also assume that the flywheel is assembled after all the other components (except the clutch housing).

Thus, we are now left with only the three assembly sequences shown in Table (2.2). For each of the three assembly sequences, there exist several combinations of orientations of the fixture into which the components can be assembled. To arrive at the optimum and most favorable fixture orientation, the assembly ID of each of the components is computed for each assembly position of the fixture. Please note that every re-orientation of the fixture costs time in terms of ID is added to the cumulative ID of that particular sequence. Tables (2.3), (2.4) & (2.5) show the sequences and the cumulative IDs.

Table (2.1) - The Precedence Constraints used in the engine assembly example

Constraints	Remarks
- → 1	The valves can be assembled into the block before any other assembly.
3 → 2	The crankshaft has to be assembled in the block before the piston.
- → 3	Nothing needs to be done before crankshaft assembly.
3 → 4	This forces the assembly of the crankshaft into the block before piston assembly.
2, 3 → 5	The piston and the crankshaft have to be assembled into the block before head assembly.
2, 3 → 6	
14 → 7	The lifters have to be assembled before the camshaft.
- → 8	The crankshaft and camshaft can be assembled independent of any other assembly
- → 9	No assembly is required before the assembly of the key with the crankshaft.
3, 9 → 10	The flywheel can be assembled on the crankshaft only after the assembly of the crankshaft into the block and the key assembly.
3, 9 → 11	
- → 12	Nothing needs to be done before assembling the flywheel and the clutch housing.
9, 10, 11 → 13	The clutch housing can be assembled on the crankshaft only after the crankshaft, key and flywheel are assembled.
- → 14	Nothing needs to be done before the lifter assembly.

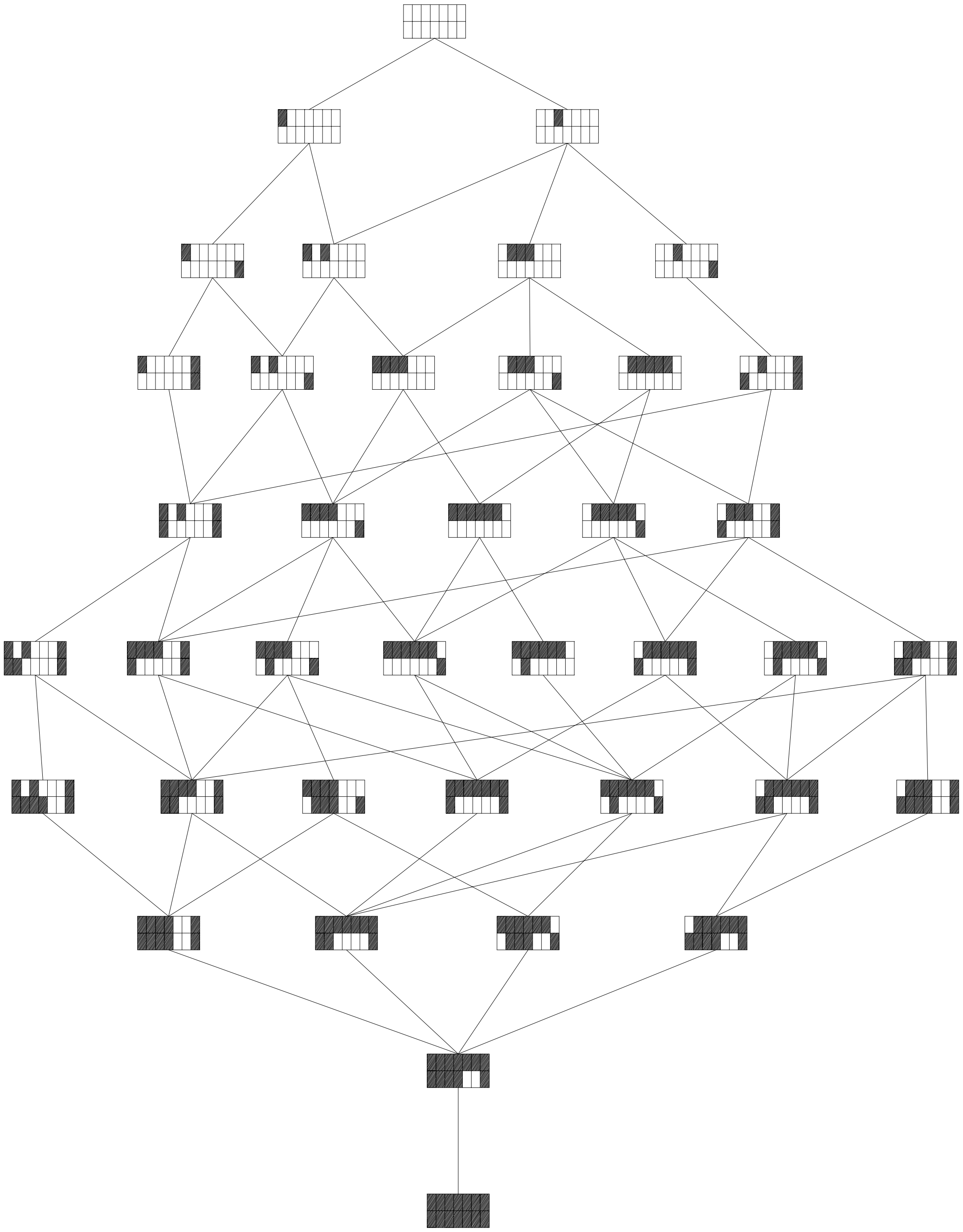


Figure (2.3) - Graphical representation of the feasible assembly sequences after applying the precedence constraints.

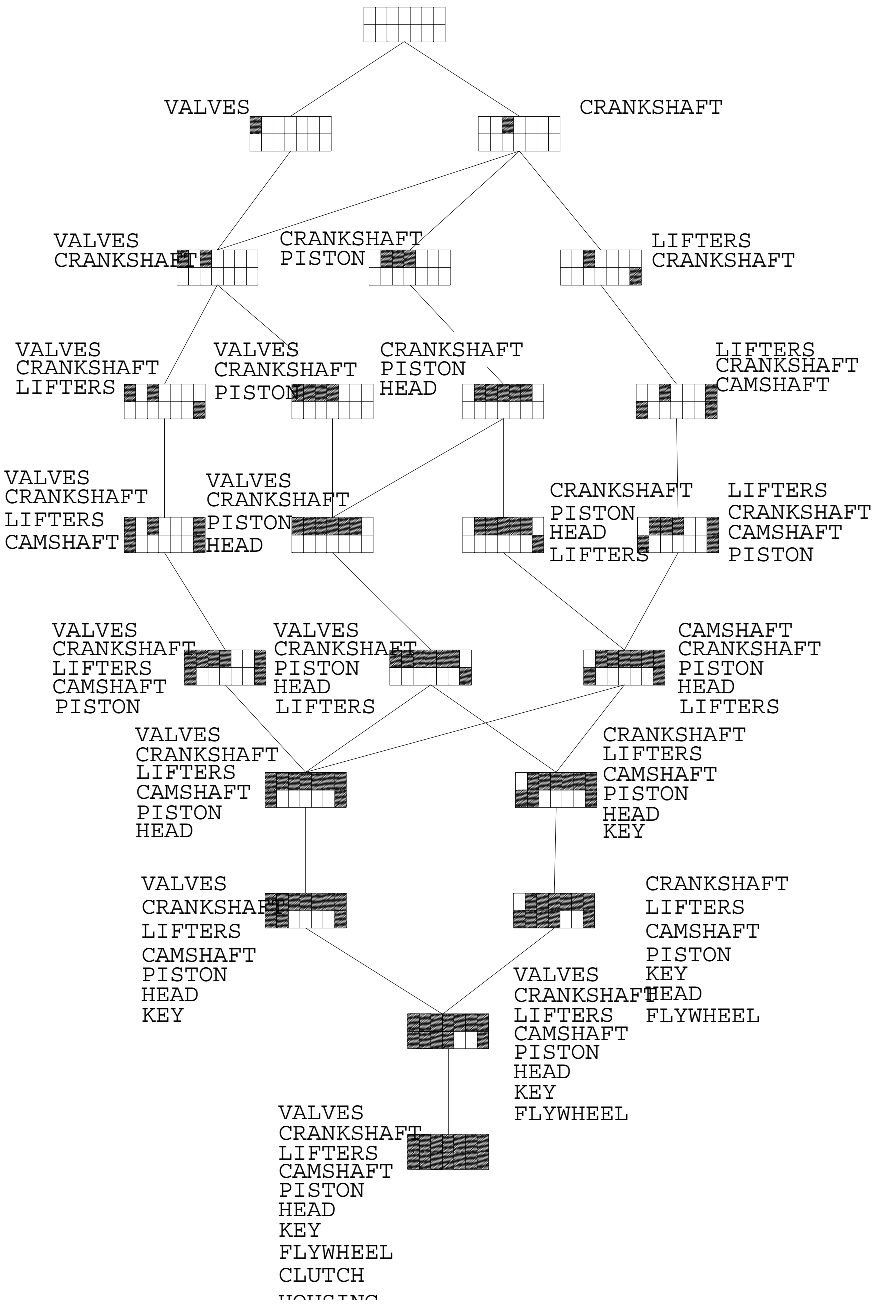


Figure (2.4) - Graphical representation of the feasible assembly sequences after applying the precedence and user-defined constraints.

Table (2.2) - the feasible assembly sequences after applying the Constraints

Sequence 1	Sequence 2	Sequence 3
Valves	Crankshaft	Crankshaft
Crankshaft	Piston	Valves
Piston	Head	Piston
Head	Lifters	Head
Lifters	Camshaft	Lifters
Camshaft	Valves	Camshaft
Key	Key	Key
Flywheel	Flywheel	Flywheel
Clutch Housing	Clutch Housing	Clutch Housing

Table (2.3) - table shows the IDs computed for sequence 1 shown in table (2.2).
The numbers in columns indicate the orientation of the fixture and the assembly ID for that position of the fixture.

A	1	2	3	4
Crankshaft	0^0 0.6	90^0 1.3	90^0 1.3	90^0 1.3
Valves	0^0 11.9	0^0 11.9	90^0 12.3	90^0 12.3
Piston	90^0 2.2	90^0 2.2	90^0 2.2	90^0 2.2
Head	0^0 3.5	0^0 3.5	0^0 3.5	0^0 3.5
Lifters	0^0 0.9	0^0 0.9	0^0 0.9	0^0 0.9
Camshaft	0^0 3.0	0^0 3.0	0^0 3.0	0^0 3.0
Key	90^0 1.6	180^0 3.0	180^0 3.0	90^0 1.6
Flywheel	90^0 3.0	180^0 2.1	180^0 2.1	90^0 3.0
Clutch	90^0 5.9	180^0 5.0	180^0 5.0	90^0 5.9
Cumulative ID	37.4	37.0	36.5	36.9

Tables (2.4) & (2.5) - These tables show the IDs computed for sequences 2 and 3 respectively shown in Table (2.2). The numbers in columns indicate the orientation of the fixture and the assembly ID for that position of the fixture.

B	**1**	**2**
Valves	90^0 12.3	90^0 12.3
Crankshaft	90^0 1.3	90^0 1.3
Piston	90^0 2.2	90^0 2.2
Head	0^0 3.5	0^0 3.5
Lifters	0^0 0.9	0^0 0.9
Camshaft	0^0 3.0	0^0 3.0
Key	180^0 3.0	90^0 1.6
Flywheel	180^0 2.1	90^0 3.0
Clutch	180^0 5.0	90^0 5.9
Cumulative ID	36.5	36.9

C	**1**	**2**
Crankshaft	90^0 1.3	90^0 1.3
Valves	90^0 12.3	90^0 12.3
Piston	90^0 2.2	90^0 2.2
Head	0^0 3.5	0^0 3.5
Lifters	0^0 0.9	0^0 0.9
Camshaft	0^0 3.0	0^0 3.0
Key	180^0 3.0	90^0 1.6
Flywheel	180^0 2.1	90^0 3.0
Clutch	180^0 5.0	90^0 5.9
Cumulative ID	36.0	36.9

2.2.2 Calculation of IDs

In Tables (2.3), (2.4) & (2.5), the cumulative IDs for all the feasible sequences and different positions of the fixture are listed. The number left justified in each column indicates the position of the fixture/block. The three positions of the block are as shown in Figure (2.4). The cumulative IDs are computed after the addition of addition of a cost of 1.6 for each re-orientation of the fixture/block. It is clear from the above that the sequence with the lowest ID is the last; case C1. This gives the optimum assembly sequence and the necessary indexing positions of the fixture. If two or more sequences show equal cumulative IDs, the sequence with the least number of indexing positions of the fixture gives the optimum assembly sequence. Since the range of cumulative ID values is not wide, near optimal plans remain as viable alternatives.

We noted earlier that according to the Hitachi AEM method of T-downs, vertically downward assembly is the best for most of the components. In most product assemblies, the base fixture has to be re-oriented so that vertical assembly of all the components is possible. It can also be seen from our example that the optimum assembly sequence demands re-orientation of the base fixture. This re-orientation can be achieved either by using the robot to perform the re-orientation or by using an indexing fixture. Using the robot to perform the re-orientation adds time to the assembly process while an indexing fixture costs money to manufacture. Hence, there is a cost associated with the re-orientation task. A decision regarding the necessity of a fixture can be made based on the number of units to be assembled, the ID reduction per unit and the cost of an ID. A heuristic for developing constraints based on additional fixtures is:

Fixture cost β ID reduction/unit x number of units x cost of ID   (in \$/bit)............(2.1)

Here the fixture cost includes the engineering cost and the workspace size increase. ID has the units of bits/sec and can easily be set to a dollar figure per second.

2.2.3 Design of a gripper

We will now attempt to show that the heuristic presented can be useful in designing a gripper. The heuristic gives an optimal assembly sequence and the orientations in which the different components are presented to the assembler. This gives us a list of component features that can be used for grasping. It is common to find several features in each component that can be made use of for grasping and assembling. For example, consider the flywheel. The flywheel can be grasped either around the circumference with jaws like v-blocks or the vanes on the flywheel can be used to grasp it as shown in figure (2.5). Similarly, all the components in the assembly are investigated for different grasping features and cataloged. The geometry of the fingers (shown in figure (2.5) and figure (2.7) for the engine example) is chosen as a result of the union of the set of all the grasping features cataloged. Simple principles of geometry and trigonometry are used for this calculation.

Once the geometry of the fingers is identified, a force analysis can be carried out in order to identify the force requirements of the gripper. The heaviest component is the most difficult to handle in terms of the grasping force necessary. In our case, the flywheel is the heaviest and most difficult to handle. It can be seen from figure (2.5) (a free body diagram of the forces and moments) that the grasping position is such that the weight (W) of the flywheel acting downwards exerts a moment on the fingers grasping it, thus trying to open them out. This moment translates to a couple (F * x) that is necessary to counter its effect. By knowing the couple arm x, the gripper force F can be easily calculated. Thus we see that the heuristic can be used in designing a gripper.

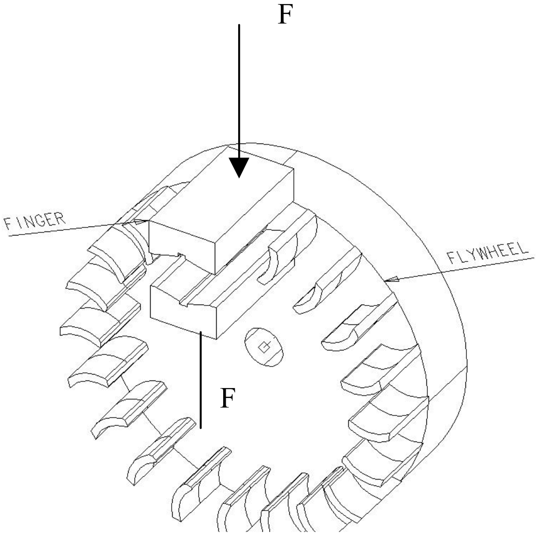


Figure (2.5) – Three-dimensional view of the grasping of the flywheel.

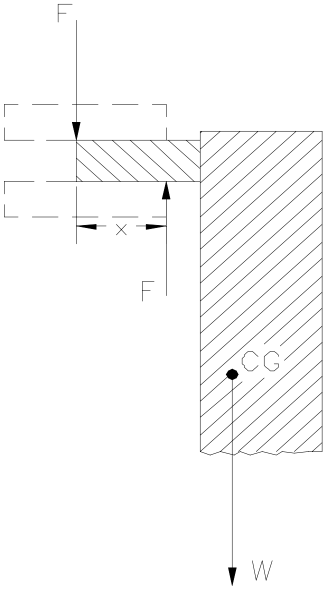


Figure (2.6) - Freebody diagram showing the forces and moments in grasping the flywheel.

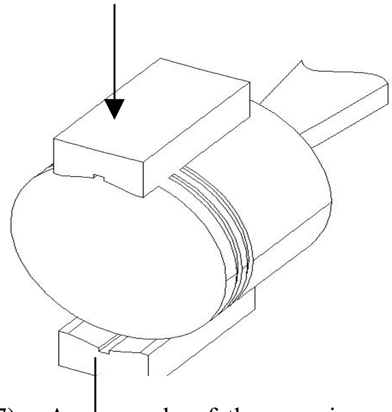


Figure (2.7) - An example of the grasping configuration holding a cylindrical component (piston). The piston rings are assumed to be pre-compressed at the workstation.

In the above example, we merely attempt to show that the results of the heuristic can be used to find a single grasping configuration for all the parts in the assembly. The advantage of using a single grasping configuration as against using more than one gripper is to avoid indexing the gripper during the assembly. This results in reducing assembly time. However, some decisions regarding providing necessary tooling at the workstation may need to be taken based on the task on hand. For example, consider the assembly of the piston; we suggest a configuration shown in figure (2.7). It is necessary to compress the piston rings before assembling the piston into the cylinder. This can be achieved by pre-compressing the piston rings at the workstation by using a piston ring compressor and grasp the ring compressor instead of the grasping the piston itself. Gripper design itself is a very detailed topic and needs to be studied in greater detail. As mentioned above, this section is only meant to demonstrate the use of the heuristic developed to synthesize a single gripper.

2.3 Limitations of the Heuristic

The ID values and their costs in practice will vary widely if one considers alternatives to manual assembly. If one were to apply a conventional industrial robot to a given assembly task, ID values would be found to depend strongly on robot repeatability and part tolerances. This "first cost" in characterizing the task-effector IDs may be burdensome to the automation engineer, and may obviate robotics in favor of manual assembly, unless the ID values are found to be very low. These low values can be realized through engineered compliances that reduce or eliminate difficulty due to part insertion (Sturges and Laowattana, 1995).

Unfortunately, ID evaluation remains a manual process, although it is fairly accurate and easy to learn (Hinckley, 1993). Algorithmic ID determination is presently a research issue, although some headway has been made (Yang and Sturges, 1992). The heuristic presented is similarly a manual process that may discourage the optimization of assembly plans, but some algorithmic approaches have been reported (Abell et. al., 1991).

In this chapter, we presented a heuristic to determine an optimal sequence for any given assembly. The liaison technique was used to enumerate all the possible assembly sequences based on precedence constraints. The number of feasible assembly sequences was reduced by applying constraints to avoid some unstable assembly sequences. For the remaining sequences aggregate IDs were calculated based on the individual IDs of each component and different positions of the base fixture. The sequence with the least aggregate ID is the optimum sequence for the assembly. The optimum assembly also gives the preferred orientations of the components. Using this information, the feasible grasping features of each component is cataloged. A union of all the feasible grasping configurations is found and is used in the design of a single gripper.