Assembly 2 Docs

Hamish

March 23, 2015

Abstract

Documents for personal use, for keeping tracks of the maths behind the Assembly 2 work bench for FreeCAD v0.15.

Contents

1 Solver approach

Adjust the placement variables (position and rotation variables, 6 for each object) as satisfy all constraints; by adding one constraint at a time and adjusting the degrees-of-freedom. Ideally, this degrees-of-freedom can be identified, so that they can be adjusted with having to check previous constraints. For non-simple systems this is not practical however as there are to many combinations to hard-code.

Therefore a hierarchical constraint system is used. When attempting the solve the current constraint, the placement variables are also adjusted/refreshed according to the previous constraints to allow for non-perfect degrees-of-freedom. This is done in a hierarchical way, with parent constraints minimally adjusting the placement variables as to satisfy the assembly constraints.

2 Plane mating constraint

2 parts

1. rotating objects as to align selected faces (not done if face and vertex are selected), **axis alignment union**
2. moving the objects as to specified offset is satisfied, **plane offset union**

3 plane offset union - analytical solution

inputs

$\mathbf{a}$ - normal vector of reference face.

$\mathbf{p}$ - point on object 1.

$\mathbf{q}$ - point on object 2.

α - specified offset

$\mathbf{d}_1, \mathbf{d}_2...$ - linear motion degree-of-freedom for object 1/2 (max of 3, min of 1)

required displacement in the direction of $\mathbf{a}$:

$$r = \mathbf{a} \cdot (\mathbf{p} - \mathbf{q}) - \alpha \tag{1}$$

require components for each $\mathbf{d}$, $v_1, v_2, ...$ therefore equal to

$$\mathbf{a} \cdot (v_1\mathbf{d}_1 + v_2\mathbf{d}_2 + v_3\mathbf{d}_3) = r \tag{2}$$

which has infinite solutions if more than one degree-of-freedom with $\mathbf{d} \cdot \mathbf{a} \neq 0$

Therefore looking for least norm solution of:

$$r = a_x(v_1 d_{1,x} + v_2 d_{2,x} + \dots) + a_y(v_1 d_{1,y} + v_2 d_{2,y} + \dots) + a_z(v_1 d_{1,z} + v_2 d_{2,z} + \dots) \tag{3}$$

which gives

$$\begin{bmatrix}(a_x d_{1,x} + a_y d_{1,y} + a_z d_{1,z}) & (a_x d_{2,x} + a_y d_{2,y} + a_z d_{2,z}) & (a_x d_{3,x} + a_y d_{3,y} + a_z d_{3,z})\end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = [r] \tag{4}$$

$$\begin{bmatrix}\mathbf{a} \cdot \mathbf{d}_1 & \mathbf{a} \cdot \mathbf{d}_2 & \mathbf{a} \cdot \mathbf{d}_3\end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = r \tag{5}$$

$$A\mathbf{v} = [r] \tag{6}$$

then solve for least norm using numpy.linalg.lstsq