

Assembly 2 Docs

Hamish

June 19, 2015

Abstract

Documents for personal use to help track of the maths behind the Assembly 2 workbench add-on for FreeCAD v0.15+.

Contents

1 Solver approach	1
2 Plane mating constraint	1
3 plane offset union - analytical solution	1
4 Numerically Reducing the System's Degrees-of-Freedom	2
4.1 assuming h is a linear system	2
4.2 assuming h is a quadratic system	2
4.3 trail and error approach	3
5 generating repairing/orthogonal basis functions	3

1 Solver approach

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

Therefore a hierarchical constraint system is used. When attempting to 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

a - normal vector of reference face.

p - point on object 1.

q - point on object 2.

α - specified offset

d₁, d₂... - linear motion degree-of-freedom for object 1/2 (max of 3, min of 1)

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

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

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

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

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:

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

which gives

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

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

$$\mathbf{A} \mathbf{v} = [r] \quad (6)$$

then solve for least norm using `numpy.linalg.lstsq`

4 Numerically Reducing the System's Degrees-of-Freedom

New degrees-of-freedom need to be determined which allow for adjustment without the constraint's equality function, h , being violated. Therefore h as a function of the current DOF \mathbf{x} needs be satisfied so that

$$h(\mathbf{x}) = 0 \quad (7)$$

The DOF of the new system will therefore result in a change of \mathbf{x} ($\Delta \mathbf{x}$) so that

$$h(\mathbf{x} + \Delta \mathbf{x}) = 0 \quad (8)$$

4.1 assuming h is a linear system

$$\begin{bmatrix} dh/dx1 & dh/dx2 & \dots \end{bmatrix} \mathbf{x} + c = 0 \quad (9)$$

$$\mathbf{b} \cdot \mathbf{x} + c = 0 \quad (10)$$

DOF therefore need to satisfy

$$\mathbf{b} \cdot (\mathbf{x} + \Delta \mathbf{x}) + c = 0 \quad (11)$$

leading to (eq 11- eq 10)

$$\mathbf{b} \cdot \Delta \mathbf{x} = 0 \quad (12)$$

Therefore any vector orgonal to \mathbf{b} is a degree of freedom.

4.2 assuming h is a quatdratic system

$$\frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b} \cdot \mathbf{x} + c = 0 \quad (13)$$

The changes in \mathbf{x} are allowed which satisfy

$$\frac{1}{2} (\mathbf{x} + \Delta \mathbf{x})^T \mathbf{A} (\mathbf{x} + \Delta \mathbf{x}) + \mathbf{b} \cdot (\mathbf{x} + \Delta \mathbf{x}) + c = 0 \quad (14)$$

$$\frac{1}{2} (\mathbf{x} + \Delta \mathbf{x})^T (\mathbf{A} \mathbf{x} + \mathbf{A} \Delta \mathbf{x}) + \mathbf{b} \cdot (\mathbf{x} + \Delta \mathbf{x}) + c = 0 \quad (15)$$

$$\frac{1}{2} (\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{A} \Delta \mathbf{x} + \Delta \mathbf{x}^T \mathbf{A} \mathbf{x} + \Delta \mathbf{x}^T \mathbf{A} \Delta \mathbf{x}) + \mathbf{b} \cdot (\mathbf{x} + \Delta \mathbf{x}) + c = 0 \quad (16)$$

since $\frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b} \cdot \mathbf{x} + c = 0$,

$$\frac{1}{2}(\mathbf{x}^T \mathbf{A} \Delta \mathbf{x} + \Delta \mathbf{x}^T \mathbf{A} \mathbf{x} + \Delta \mathbf{x}^T \mathbf{A} \Delta \mathbf{x}) + \mathbf{b} \cdot \Delta \mathbf{x} = 0 \quad (17)$$

furthermore assuming that $\mathbf{x}^T \mathbf{A} \Delta \mathbf{x} = \Delta \mathbf{x}^T \mathbf{A} \mathbf{x}$ ($\mathbf{A} = \mathbf{A}^T$) gives

$$\frac{1}{2}\Delta \mathbf{x}^T \mathbf{A} \Delta \mathbf{x} + (\mathbf{b} + \mathbf{x}^T \mathbf{A}) \cdot \Delta \mathbf{x} = 0 \quad (18)$$

Find solutions for $\Delta \mathbf{x}$ allows for the degrees-of-freedom of the resulting system to be determined. By definition these DOFs can be altered without violating $h(\mathbf{x})$ (eq. 13), hence not requiring the previous constraints to be resatisfied/resolved.

Scenarios do occur however, where $h(\mathbf{x})$ is a complicated functional which is not quadratic.

4.3 trail and error approach

Here the approach is to pass dimensions of \mathbf{x} directly through, to create non-perfect or false degrees of freedom. The passed through dimensions of \mathbf{y} when altered require that the previous constraints in the system be resolved. The approach inside assembly 2 is determined the \mathbf{y} with largest number of dimensions, which can passed through and changed, while still be able to satisfy previous constraints.

Order probably matters, but anyway, the going to be assumed that order does not matter.

5 generating repairing/orthogonal basis functions

Use the Gram–Schmidt process https://en.wikipedia.org/wiki/Gram%E2%80%93Schmidt_process : the Gram–Schmidt process is a method for orthonormalising a set of vectors in an inner product space, most commonly the Euclidean space \mathbb{R}^n .