Gough - Stuart platform leg length calculation

[Ref. 'The Mathematics of the Stewart Platform' unknown author, &
http://memememememememe.me/post/stewart-platform-math/]

Mid z position of platform $z_{mid} := 22.19mm$

Platform displacement

Surge (x)	$x := 0mm$	Roll (rx)	$\varphi := 0deg$
Sway (y)	$y := 0mm$	Pitch (ry)	$\theta := 0deg$
Heave (z)	$z := 0mm$	Yaw (rz)	$\psi := 0deg$

roll rotation matrix $R_x := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Pitch rotation matrix $R_y := \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Yaw rotation matrix $R_z := \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

< Only need to calc 4 values in each table & range of valid angles is going to be limited, so can precalculate a subset of cos & sin values. Also need to develop fixed point sizing to maximise resolution for 16bit operations

Combined rotation matrix $R_B := R_x \cdot R_y \cdot R_z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Base anchor point co-ordinates (local)

leg: 0 1 2 3 4 5

$$b := \begin{pmatrix} 4 & 26 & 22 & -22 & -26 & -4 \\ 27.7 & -10.39 & -17.32 & -17.32 & -10.39 & 27.7 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot mm$$

axis
x
y
z

Platform anchor point co-ordinates (local)

$$p := \begin{pmatrix} 5.5 & 7.5 & 2 & -2 & -7.5 & -5.5 \\ 5.48 & 2.02 & -7.5 & -7.5 & 2.02 & 5.48 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot mm$$

point of interest offset from platform coordinate

$$offset := \begin{pmatrix} 10 \\ 0 \\ 10 \end{pmatrix} \cdot mm$$

< relative to the actual position of interest on the platform or the end of the item being actuated. This then compensates for the translation effect of rotation.

Translation array $T := \begin{pmatrix} x + offset_0 \\ y + offset_1 \\ z + z_{mid} + offset_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 0 \\ 32.19 \end{pmatrix} \cdot mm$

Leg length vectors

$$\text{vec}_{leg} := \left| \begin{array}{l} \text{for } i \in 0..5 \\ \quad l^{\langle i \rangle} \leftarrow T + R_B \cdot \left(p^{\langle i \rangle} - \text{offset}\right) - b^{\langle i \rangle} \\ \text{return } l \end{array} \right.$$

$$\text{vec}_{leg} = \begin{pmatrix} 1.5 & -18.5 & -20 & 20 & 18.5 & -1.5 \\ -22.22 & 12.41 & 9.82 & 9.82 & 12.41 & -22.22 \\ 22.19 & 22.19 & 22.19 & 22.19 & 22.19 & 22.19 \end{pmatrix} \cdot \text{mm}$$

leg mid length

$$\text{leg}_{mid} := 31.44\text{mm}$$

get length from vectors

$$\text{leg}_{len} := \left| \begin{array}{l} \text{for } i \in 0..5 \\ \quad l_{len}^{\langle i \rangle} \leftarrow \sqrt{\left(\text{vec}_{leg_{0,i}}\right)^2 + \left(\text{vec}_{leg_{1,i}}\right)^2 + \left(\text{vec}_{leg_{2,i}}\right)^2} \\ \text{return } l_{len} - \text{leg}_{mid} \end{array} \right.$$

$$\text{leg}_{len} = \left(-1.576 \times 10^{-3} \quad 2.872 \times 10^{-3} \quad 5.644 \times 10^{-3} \quad 5.644 \times 10^{-3} \quad 2.872 \times 10^{-3} \quad -1.576 \times 10^{-3}\right) \cdot \text{mm}$$

leg length limits

$$\text{leg}_{max} := 1.24\text{mm}$$

$$\text{leg}_{min} := -1.24\text{mm}$$

Check that min and max lengths are not exceeded

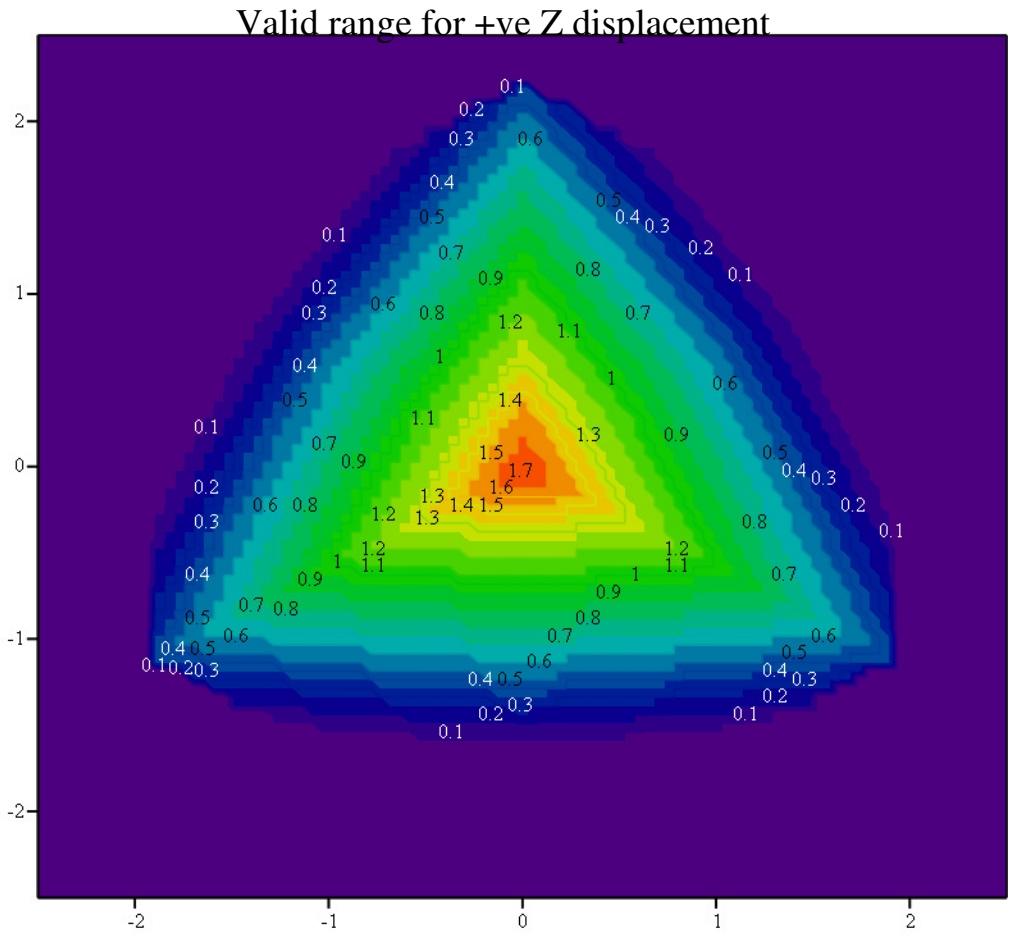
$$\left| \begin{array}{ll} \text{"OK"} & \text{if } \max\left(\text{leg}_{len}\right) < \text{leg}_{max} \wedge \min\left(\text{leg}_{len}\right) > \text{leg}_{min} \quad = \text{"OK"} \\ \text{"FAIL"} & \text{otherwise} \end{array} \right.$$

Derived function for populating contour chart; positive displacement

```
Check(x, y) := | temp_k ← 0
               | for k ∈ 0..30
               |   vec_leg ← | for n ∈ 0..5
               |             |   l^⟨n⟩ ← [ x·mm + offset_0 ; y·mm + offset_1 ; (k/10)·mm + z_mid + offset_2 ] + R_B·(p^⟨n⟩ − offset) − b^⟨n⟩
               |             | vec_leg ← 1
               |   leg_len ← | for n ∈ 0..5
               |             |   l_len^⟨n⟩ ← sqrt( (vec_leg_0,n)^2 + (vec_leg_1,n)^2 + (vec_leg_2,n)^2 )
               |             | leg_len ← l_len − leg_mid
               |   temp_k ← k/10 if max(leg_len) < leg_max ∧ min(leg_len) > leg_min
               | return temp_k

range := 2.5 mm        num_points := 100
```

build a grid of values: F := CreateMesh(Check, −range, range, −range, range, num_points, num_points)



Derived function for populating contour chart; negative displacement

$$\text{Check\_neg}(x, y) := \left| \begin{array}{l} \text{temp\_k} \leftarrow 0 \\ \text{for } k \in 0..30 \\ \quad \left| \begin{array}{l} \text{vec}_{\text{leg}} \leftarrow \left| \begin{array}{l} \text{for } n \in 0..5 \\ \quad l^{\langle n \rangle} \leftarrow \begin{pmatrix} x \cdot mm + \text{offset}_0 \\ y \cdot mm + \text{offset}_1 \\ \frac{-k}{10} \cdot mm + z_{mid} + \text{offset}_2 \end{pmatrix} + R_B \cdot \left(p^{\langle n \rangle} - \text{offset} \right) - b^{\langle n \rangle} \\ \text{vec}_{\text{leg}} \leftarrow l \end{array} \right. \\ \text{leg}_{\text{len}} \leftarrow \left| \begin{array}{l} \text{for } n \in 0..5 \\ \quad l_{\text{len}}{}^{\langle n \rangle} \leftarrow \sqrt{\left(\text{vec}_{\text{leg}_{0,n}} \right)^2 + \left(\text{vec}_{\text{leg}_{1,n}} \right)^2 + \left(\text{vec}_{\text{leg}_{2,n}} \right)^2} \\ \text{leg}_{\text{len}} \leftarrow l_{\text{len}} - \text{leg}_{\text{mid}} \end{array} \right. \\ \text{temp\_k} \leftarrow \frac{-k}{10} \text{ if } \max\left(\text{leg}_{\text{len}} \right) < \text{leg}_{\text{max}} \wedge \min\left(\text{leg}_{\text{len}} \right) > \text{leg}_{\text{min}} \end{array} \right. \\ \text{return temp\_k} \end{array} \right.$$

range = 2.5 mm num_points = 100

build a grid of values: F_neg := CreateMesh(Check_neg, −range, range, −range, range, num_points, num_points)