```julia
using SymPy
```

Rover Leg Kinematics

Eric Stoneking, Feb 2025

First, we find the position of a foot and express its components in the **B** frame. Then we find the Jacobian J, the matrix of partials of the foot position with respect to the joint angles.

We define an **H** frame, fixed at the inner side of the leg's hip joint, oriented so that $\hat{h}_1$ points radially outward from the central axis of **B**, and $\hat{h}_3 = \hat{b}_3$. The location of the hip joint is then $r\hat{h}_1$, where r is a radius length. The angle α is the angle from $\hat{b}_1$ to $\hat{h}_1$.

The upper leg is defined by the **T** frame. The rotation from **H** to **T** is a Body-3 3-2-1 rotation through $(\theta_1, \theta_2, 0)$. The length of **T** (from the hip to the knee) is ℓ_1.

The lower leg is defined by the **F** frame. The rotation from **T** to **F** is a simple rotation about $\hat{f}_2$ through θ_3. The length of **F** from the knee to the foot is ℓ_2.

The position of the foot with respect to the origin of the body is written

$$\vec{p} = r\hat{h}_1 + \ell_1\hat{t}_1 - \ell_2\hat{f}_3 \tag{1}$$

The symbolic manipulations below obtain the expressions for $\vec{p}$'s components in the body frame.

The Jacobian matrix is the 3×3 matrix

$$J_{ij} = \frac{\partial p_i}{\partial \theta_j} \tag{2}$$

We use J to find how to adjust θ to drive $\vec{p}$ toward a commanded location $\vec{p}_c$. Since the Jacobian is a linearization of a nonlinear relation, we don't want to try to cancel errors completely in a single step. We apply a correction gain ρ, $0 < \rho < 1$ to correct a portion of the error.

Suppose

$$\Delta\vec{p}_k = \vec{p}_k - \vec{p}_c \tag{3}$$

is the position error at step k. To reduce the error for the next step, we let

$$\vec{p}_{k+1} = \vec{p}_k - \rho\,\Delta\vec{p}_k \tag{4}$$

The joint errors are related (to first order) to the position errors by

$$\Delta\vec{p}_k = J_k \Delta\vec{\theta}_k \tag{5}$$

If J is invertible, then we adjust the joint angles

$$\vec{\theta}_{k+1} = \vec{\theta}_k - \rho\, J_k^{-1}\,\Delta\vec{p}_k \tag{6}$$

```
r,l1,l2 = symbols("r,\\ell_1,\\ell_2")
sa,ca = symbols("s_\\alpha,c_\\alpha")
s1,c1 = symbols("s_1,c_1")
s2,c2 = symbols("s_2,c_2")
s3,c3 = symbols("s_3,c_3")
b1,b2,b3 = symbols("\\hatb_1,\\hatb_2,\\hatb_3",commutative=false);
h1,h2,h3 = symbols("\\hath_1,\\hath_2,\\hath_3",commutative=false);
t1,t2,t3 = symbols("\\hatt_1,\\hatt_2,\\hatt_3",commutative=false);
f1,f2,f3 = symbols("\\hatf_1,\\hatf_2,\\hatf_3",commutative=false);
```

```
CHB = [ca sa 0;-sa ca 0;0 0 1]
CTH = [c1*c2 s1*c2 -s2;-s1 c1 0; c1*s2 s1*s2 c2]
CFT = [c3 0 -s3;0 1 0;s3 0 c3]
```

$$\begin{bmatrix} c_3 & 0 & -s_3 \\ 0 & 1 & 0 \\ s_3 & 0 & c_3 \end{bmatrix} \tag{7}$$

```
pf = r*h1+l1*t1-l2*f3
```

$$\ell_1\hat{t}_1 - \ell_2\hat{f}_3 + r\hat{h}_1 \tag{8}$$

```
pf = subs(pf,f3 => s3*t1+c3*t3)
```

$$\ell_1\hat{t}_1 - \ell_2\left(c_3\hat{t}_3 + s_3\hat{t}_1\right) + r\hat{h}_1 \tag{9}$$

```
pf = subs(pf,t1 => c1*c2*h1+s1*c2*h2-s2*h3)
pf = subs(pf,t3 => c1*s2*h1+s1*s2*h2+c2*h3)
```

$$\ell_1 \left(c_1c_2\hat{h}_1 + c_2s_1\hat{h}_2 - s_2\hat{h}_3\right) - \ell_2 \left(c_3 \left(c_1s_2\hat{h}_1 + c_2\hat{h}_3 + s_1s_2\hat{h}_2\right) + s_3 \left(c_1c_2\hat{h}_1 + c_2s_1\hat{h}_2 - s_2\hat{h}_3\right)\right) + r\hat{h}_1 \tag{10}$$

```
pf = subs(pf,h1 => ca*b1+sa*b2)
pf = subs(pf,h2 => -sa*b1+ca*b2)
pf = subs(pf,h3 => b3)
```

$$\ell_1 \left(c_1c_2 \left(c_\alpha\hat{b}_1 + s_\alpha\hat{b}_2\right) + c_2s_1 \left(c_\alpha\hat{b}_2 - s_\alpha\hat{b}_1\right) - s_2\hat{b}_3\right) - \ell_2 \left(c_3 \left(c_1s_2 \left(c_\alpha\hat{b}_1 + s_\alpha\hat{b}_2\right) + c_2\hat{b}_3 + s_1s_2 \left(c_\alpha\hat{b}_2 - s_\alpha\hat{b}_1\right)\right) \tag{11}$$

```
px = FindComponent(pf*b1)
px = collect(px,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$\ell_1 \left(c_1c_2c_\alpha - c_2s_1s_\alpha\right) + \ell_2 \left(-c_1c_2c_\alpha s_3 - c_1c_3c_\alpha s_2 + c_2s_1s_3s_\alpha + c_3s_1s_2s_\alpha\right) + c_\alpha r \tag{12}$$

```
py = FindComponent(pf*b2)
py = collect(py,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$\ell_1 \left(c_1c_2s_\alpha + c_2c_\alpha s_1\right) + \ell_2 \left(-c_1c_2s_3s_\alpha - c_1c_3s_2s_\alpha - c_2c_\alpha s_1s_3 - c_3c_\alpha s_1s_2\right) + rs_\alpha \tag{13}$$

```
pz = FindComponent(pf*b3)
pz = collect(pz,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$-\ell_1s_2 + \ell_2 \left(-c_2c_3 + s_2s_3\right) \tag{14}$$

Partials

```
dpxdθ1 = diff(px,s1)*c1 + diff(px,c1)*(-s1)
dpxdθ1 = collect(dpxdθ1,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$c_1 \left(-\ell_1 c_2 s_\alpha + \ell_2 \left(c_2 s_3 s_\alpha + c_3 s_2 s_\alpha\right)\right) + s_1 \left(-\ell_1 c_2 c_\alpha - \ell_2 \left(-c_2 c_\alpha s_3 - c_3 c_\alpha s_2\right)\right) \tag{15}$$

```
dpxdθ2 = diff(px,s2)*c2 + diff(px,c2)*(-s2)
dpxdθ2 = collect(dpxdθ2,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$\ell_2 c_2 \left(-c_1 c_3 c_\alpha + c_3 s_1 s_\alpha\right) + s_2 \left(-\ell_1 \left(c_1 c_\alpha - s_1 s_\alpha\right) - \ell_2 \left(-c_1 c_\alpha s_3 + s_1 s_3 s_\alpha\right)\right) \tag{16}$$

```
dpxdθ3 = diff(px,s3)*c3 + diff(px,c3)*(-s3)
dpxdθ3 = collect(dpxdθ3,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$\ell_2 \left(c_3 \left(-c_1 c_2 c_\alpha + c_2 s_1 s_\alpha\right) - s_3 \left(-c_1 c_\alpha s_2 + s_1 s_2 s_\alpha\right)\right) \tag{17}$$

```
dpydθ1 = diff(py,s1)*c1 + diff(py,c1)*(-s1)
dpydθ1 = collect(dpydθ1,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$c_1 \left(\ell_1 c_2 c_\alpha + \ell_2 \left(-c_2 c_\alpha s_3 - c_3 c_\alpha s_2\right)\right) + s_1 \left(-\ell_1 c_2 s_\alpha - \ell_2 \left(-c_2 s_3 s_\alpha - c_3 s_2 s_\alpha\right)\right) \tag{18}$$

```
dpydθ2 = diff(py,s2)*c2 + diff(py,c2)*(-s2)
dpydθ2 = collect(dpydθ2,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$\ell_2 c_2 \left(-c_1 c_3 s_\alpha - c_3 c_\alpha s_1\right) + s_2 \left(-\ell_1 \left(c_1 s_\alpha + c_\alpha s_1\right) - \ell_2 \left(-c_1 s_3 s_\alpha - c_\alpha s_1 s_3\right)\right) \tag{19}$$

```
dpydθ3 = diff(py,s3)*c3 + diff(py,c3)*(-s3)
dpydθ3 = collect(dpydθ3,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$\ell_2 \left(c_3 \left(-c_1 c_2 s_\alpha - c_2 c_\alpha s_1\right) - s_3 \left(-c_1 s_2 s_\alpha - c_\alpha s_1 s_2\right)\right) \tag{20}$$

```
dpzdθ1 = diff(pz,s1)*c1 + diff(pz,c1)*(-s1)
dpzdθ1 = collect(dpzdθ1,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$0 \tag{21}$$

```
dpzdθ2 = diff(pz,s2)*c2 + diff(pz,c2)*(-s2)
dpzdθ2 = collect(dpzdθ2,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$\ell_2 c_3 s_2 + c_2 \left(-\ell_1 + \ell_2 s_3\right) \tag{22}$$

```
dpzdθ3 = diff(pz,s3)*c3 + diff(pz,c3)*(-s3)
dpzdθ3 = collect(dpzdθ3,[l1,l2,c1,s1,c2,s2,c3,s3])
```

$$\ell_2 \left(c_2 s_3 + c_3 s_2\right) \tag{23}$$

Appendix: Functions

```
function FindComponent(p)
   p = expand(p)
   p = subs(p,b1*b1 => 1)
   p = subs(p,b2*b2 => 1)
   p = subs(p,b3*b3 => 1)
   p = subs(p,b1*b2 => 0)
   p = subs(p,b2*b3 => 0)
   p = subs(p,b3*b1 => 0)
   p = subs(p,b2*b1 => 0)
   p = subs(p,b3*b2 => 0)
   p = subs(p,b1*b3 => 0)
   return p;
end;
```