Pyteron cod for forward

Kinematics

i) import numpy cump.

from runpy import array.

from sympy imposit symbols.

cos, rin, Pi, simplify, sgrt)

atan 2, pprint

from sympy, matrices import matrix.

broating symbols for Dr Parameter;

91,92,93, 94,95 = sumbols (91:6").
(angles)

di, d2, d3, d4, ds = Symbols ('d1:8')

link offets.

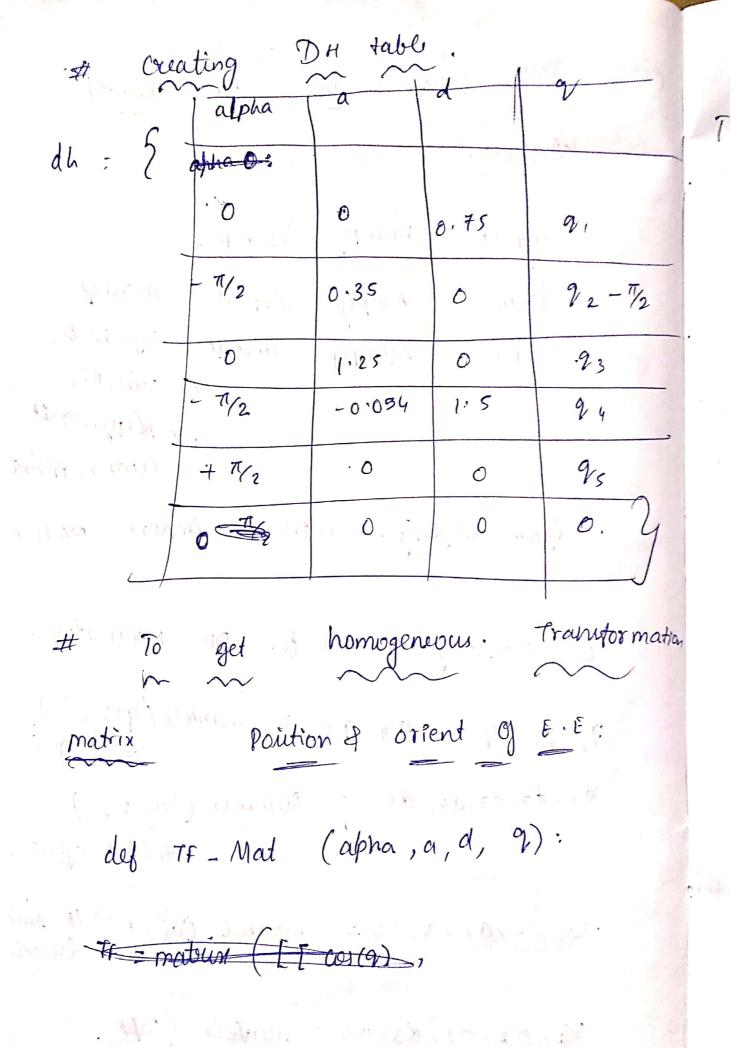
·ao, a1, a2, a3 a4 = symbols (°ao: +') # link length.

X1, X2, X2, X3, X4 = symbols ('apt

ap alpha o , alpha 1, opha 8, alpha 3, alpha 4

= symbols (apprao : 7').

Toint twist angle.
Scanned with CamScanner



```
det 7.
TF = modix([[cos(9) . ., -. lin(9), 0, .a],
   [L \sin(q)^*\cos(\alpha), \cos(q) + \cos(\alpha), -\sin(\alpha),
                                  - rin (x) xd
  [ win(q) * in(x), cos(q) * win(x), cos(x),
                                   COS (x)*d)
[0,0,0]
  retwin TF.
              DH Table (milestitate)
        for DH
  To -1 = TF-Mat (appa o, ao, d, qi). mily (du)
             e peritina y and I to chain
  ti-2 = 11
  T5 - 6 =
  # Transformation from
            effector: (F-k)
```

1 link-72. TO-2 = (To-1 * T1-2)4(Bau) link_ 0 to link-2, 70-3 = (10-2 ×72-3) #(Ball) link-0 to link-3. of the contract of the contract of the contract of To-4 -To-5 = TO-6 = (10= * TS-6) + (Base) lik_ Oto the link-b orientation aprection matrix. R- Ymatrix. R-y: matrix ([[os (-np.pi/2): 0, lin (-np.pi/2.) [o], 1., o, o], [- win (-np. pile.), 0, 201 (-np. pil2.),0) [0,0,0,1]] matrix ([[cos (np.pi), -un (np.pi), o, o)

Scanned with CamScanner