```
> # proof of the mobility for the 1st family
>
> restart:with(LinearAlgebra):
>
> # entries of matrix R
> r11:=e0^2+e1^2-e2^2-e3^2:
> r12:=2*(e1*e2-e0*e3):
> r13:=2*(e1*e3+e0*e2):
> r21:=2*(e1*e2+e0*e3):
> r22:=e0^2-e1^2+e2^2-e3^2:
> r23:=2*(e2*e3-e0*e1):
> r31:=2*(e1*e3-e0*e2):
> r32:=2*(e2*e3+e0*e1):
> r33:=e0^2-e1^2-e2^2+e3^2:
>
> # translational part
> t1:=2*(f1*e0-f0*e1+f3*e2-f2*e3):
> t2:=2*(f2*e0-f0*e2+f1*e3-f3*e1):
> t3:=2*(f3*e0-f0*e3+f2*e1-f1*e2):
>
> # direct isometry: (AA,BB,CC) are the coordinates of the point
  (Y1,Y2,Y3) in the moving frame with respect to the fixed frame
> AA:=t1+r11*Y1+r12*Y2+r13*Y3:
> BB:=t2+r21*Y1+r22*Y2+r23*Y3:
> CC:=t3+r31*Y1+r32*Y2+r33*Y3:
>
> # computation of the sphere condition Lambda with base point
  (X1,X2,X3), platform point (Y1,Y2,Y3) and squard radius dd
> N:=e0^2+e1^2+e2^2+e3^2:
> Sphere:=AA^2+BB^2+CC^2-2*N*(AA*X1+BB*X2+CC*X3)+N^2*(X1^2+X2^2+X3
  ^2-dd):
> psi:=e0*f0+e1*f1+e2*f2+e3*f3:
> Lambda:=factor(Sphere + 4*psi^2)/N:nops(%);
```



```
> # parametrisation of the family
> B1:=0:C1:=0:
> A2:=mu1*A1:B2:=0:C2:=0:
> C3:=0:
> A4:=mu3*A3:B4:=mu3*B3:C4:=0:
> A6:=mu5*A5:B6:=mu5*B5:C6:=mu5*C5:
> a1:=-A2:b1:=-B2:c1:=-C2:
> a2:=-A1:b2:=-B1:c2:=-C1:
> a3:=-A4:b3:=-B4:c3:=-C4:
> a4:=-A3:b4:=-B3:c4:=-C3:
> a5:=-A6:b5:=-B6:c5:=-C6:
> a6:=-A5:b6:=-B5:c6:=-C5:
```

```
>
> # defining Lambda1, ..., Lambda6
> Lambda1:=simplify(subs(X1=A1,X2=B1,X3=C1,Y1=a1,Y2=b1,Y3=c1,dd=dd
  1,Lambda)):
> Lambda2:=simplify(subs(X1=A2,X2=B2,X3=C2,Y1=a2,Y2=b2,Y3=c2,dd=dd
  2,Lambda)):
> Lambda3:=simplify(subs(X1=A3,X2=B3,X3=C3,Y1=a3,Y2=b3,Y3=c3,dd=dd
  3,Lambda)):
> Lambda4:=simplify(subs(X1=A4,X2=B4,X3=C4,Y1=a4,Y2=b4,Y3=c4,dd=dd
  4,Lambda)):
> Lambda5:=simplify(subs(X1=A5,X2=B5,X3=C5,Y1=a5,Y2=b5,Y3=c5,dd=dd
  5,Lambda)):
> Lambda6:=simplify(subs(X1=A6,X2=B6,X3=C6,Y1=a6,Y2=b6,Y3=c6,dd=dd
  6,Lambda)):
>
> # defining Delata12,...
> Delta12:=simplify(Lambda1-Lambda2):
> Delta13:=simplify(Lambda1-Lambda3):
> Delta14:=simplify(Lambda1-Lambda4):
> Delta15:=simplify(Lambda1-Lambda5):
> Delta16:=simplify(Lambda1-Lambda6):
> Delta23:=simplify(Lambda1-Lambda3):
> Delta24:=simplify(Lambda1-Lambda4):
> Delta25:=simplify(Lambda1-Lambda5):
> Delta26:=simplify(Lambda1-Lambda6):
>
> # defining Sij, Tij, Uij, Vij, Wij
> S12:=simplify(select(has,Lambda1-Lambda2,f0)/f0):
> T12:=simplify(select(has,Lambda1-Lambda2,f1)/f1):
> U12:=simplify(select(has,Lambda1-Lambda2,f2)/f2):
> V12:=simplify(select(has,Lambda1-Lambda2,f3)/f3):
> W12:=simplify(Lambda1-Lambda2-S12*f0-T12*f1-U12*f2-V12*f3):
>
> S13:=simplify(select(has,Lambda1-Lambda3,f0)/f0):
> T13:=simplify(select(has,Lambda1-Lambda3,f1)/f1):
> U13:=simplify(select(has,Lambda1-Lambda3,f2)/f2):
> V13:=simplify(select(has,Lambda1-Lambda3,f3)/f3):
> W13:=simplify(Lambda1-Lambda3-S13*f0-T13*f1-U13*f2-V13*f3):
>
> S14:=simplify(select(has,Lambda1-Lambda4,f0)/f0):
> T14:=simplify(select(has,Lambda1-Lambda4,f1)/f1):
> U14:=simplify(select(has,Lambda1-Lambda4,f2)/f2):
> V14:=simplify(select(has,Lambda1-Lambda4,f3)/f3):
> W14:=simplify(Lambda1-Lambda4-S14*f0-T14*f1-U14*f2-V14*f3):
>
> S15:=simplify(select(has,Lambda1-Lambda5,f0)/f0):
```

```
> T15:=simplify(select(has,Lambda1-Lambda5,f1)/f1):
> U15:=simplify(select(has,Lambda1-Lambda5,f2)/f2):
> V15:=simplify(select(has,Lambda1-Lambda5,f3)/f3):
> W15:=simplify(Lambda1-Lambda5-S15*f0-T15*f1-U15*f2-V15*f3):
>
> S16:=simplify(select(has,Lambda1-Lambda6,f0)/f0):
> T16:=simplify(select(has,Lambda1-Lambda6,f1)/f1):
> U16:=simplify(select(has,Lambda1-Lambda6,f2)/f2):
> V16:=simplify(select(has,Lambda1-Lambda6,f3)/f3):
> W16:=simplify(Lambda1-Lambda6-S16*f0-T16*f1-U16*f2-V16*f3):
>
> S23:=simplify(select(has,Lambda2-Lambda3,f0)/f0):
> T23:=simplify(select(has,Lambda2-Lambda3,f1)/f1):
> U23:=simplify(select(has,Lambda2-Lambda3,f2)/f2):
> V23:=simplify(select(has,Lambda2-Lambda3,f3)/f3):
> W23:=simplify(Lambda2-Lambda3-S23*f0-T23*f1-U23*f2-V23*f3):
>
> S24:=simplify(select(has,Lambda2-Lambda4,f0)/f0):
> T24:=simplify(select(has,Lambda2-Lambda4,f1)/f1):
> U24:=simplify(select(has,Lambda2-Lambda4,f2)/f2):
> V24:=simplify(select(has,Lambda2-Lambda4,f3)/f3):
> W24:=simplify(Lambda2-Lambda4-S24*f0-T24*f1-U24*f2-V24*f3):
>
> S25:=simplify(select(has,Lambda2-Lambda5,f0)/f0):
> T25:=simplify(select(has,Lambda2-Lambda5,f1)/f1):
> U25:=simplify(select(has,Lambda2-Lambda5,f2)/f2):
> V25:=simplify(select(has,Lambda2-Lambda5,f3)/f3):
> W25:=simplify(Lambda2-Lambda5-S25*f0-T25*f1-U25*f2-V25*f3):
>
> S26:=simplify(select(has,Lambda2-Lambda6,f0)/f0):
> T26:=simplify(select(has,Lambda2-Lambda6,f1)/f1):
> U26:=simplify(select(has,Lambda2-Lambda6,f2)/f2):
> V26:=simplify(select(has,Lambda2-Lambda6,f3)/f3):
> W26:=simplify(Lambda2-Lambda6-S26*f0-T26*f1-U26*f2-V26*f3):
>
> # collecting Sij, Tij, Uij, Vij, Wij in a vector Hij
> H12:=<S12,T12,U12,V12,W12>:
> H13:=<S13,T13,U13,V13,W13>:
> H14:=<S14,T14,U14,V14,W14>:
> H15:=<S15,T15,U15,V15,W15>:
> H16:=<S16,T16,U16,V16,W16>:
> H23:=<S23,T23,U23,V23,W23>:
> H24:=<S24,T24,U24,V24,W24>:
> H25:=<S25,T25,U25,V25,W25>:
> H26:=<S26,T26,U26,V26,W26>:
>
```

```
> # the coefficients of psi are collected in H
> H:=<e0,e1,e2,e3,0>:
>
> # computing Omega1, ..., Omega6
> Omega1:=Determinant(<H23|H24|H25|H26|H>):
> Omega2:=Determinant(<H13|H14|H15|H16|H>):
> Omega3:=Determinant(<H12|H14|H15|H16|H>):
> Omega4:=Determinant(<H12|H13|H15|H16|H>):
> Omega5:=Determinant(<H12|H13|H14|H16|H>):
> Omega6:=Determinant(<H12|H13|H14|H15|H>):
>
> # computing G1, ..., G6
> G1:=factor(Omega1/N):
> G2:=factor(Omega2/N):
> G3:=factor(Omega3/N):
> G4:=factor(Omega4/N):
> G5:=factor(Omega5/N):
> G6:=factor(Omega6/N):
>
> # test
> simplify(G1-G2+G3-G4+G5-G6);
                                    0
> # specifying the leg lengths
> dd2:=dd1:dd4:=dd3:dd6:=dd5:
>
> # completing the proof
> G2n:=factor(numer(simplify(G2))):
> G3n:=factor(numer(simplify(G3))):
> G4n:=factor(numer(simplify(G4))):
> G5n:=factor(numer(simplify(G5))):
> G6n:=factor(numer(simplify(G6))):
>
> S:=gcd(gcd(gcd(G2n,G3n),gcd(G4n,G5n)),G6n):nops(%);
                                   650
>
```