Examen U3 ROBÓTICA

1. **Realizar una trayectoria rectilínea cúbica en el tiempo rest-to-rest entre dos puntos P, Q para ser ejecutada por el Origen O6, del efector final.**

Solución:

- Determinación de los coeficientes de los polinomios cúbicos para cada componente de la posición.

  Si se sabe que X, Y y Z, son una trayectoria cubica en función del tiempo, se puede deducir su posición en el tiempo inicial y el tiempo final del recorrido, así mismo se pueden deducir las velocidades en los mismos puntos, dando analíticamente los siguientes sistemas de ecuaciones.

$$\begin{bmatrix} axt_0^3 + bxt_0^2 + cxt_0 + dx \\ 3axt_0^2 + 2bxt_0 + cx \\ axt_f^3 + bxt_f^2 + cxt_f + dx \\ 3axt_f^2 + 2bxt_f + cx \end{bmatrix} = \begin{bmatrix} x_0 \\ v_{x0} \\ x_f \\ v_{xf} \end{bmatrix} \qquad \begin{bmatrix} ayt_0^3 + byt_0^2 + cyt_0 + dy \\ 3ayt_0^2 + 2byt_0 + cy \\ ayt_f^3 + byt_f^2 + cyt_f + dy \\ 3ayt_f^2 + 2byt_f + cy \end{bmatrix} = \begin{bmatrix} y_0 \\ v_{y0} \\ y_f \\ v_{yf} \end{bmatrix} \qquad \begin{bmatrix} azt_0^3 + bzt_0^2 + czt_0 + dz \\ 3azt_0^2 + 2bzt_0 + cz \\ azt_f^3 + bzt_f^2 + czt_f + dz \\ 3azt_f^2 + 2bzt_f + cz \end{bmatrix} = \begin{bmatrix} z_0 \\ v_{z0} \\ z_f \\ v_{zf} \end{bmatrix}$$

  Donde ai, bi, ci, di con *i* equivalente a x, y o z respectivamente, son los coeficientes que satisfacen los sistemas.

  Para conocer los valores de los coeficientes de la trayectoria se necesita conocer al menos los puntos $(x0, y0, z0)$ y (xf, yf, zf). Y para que las trayectorias cumplan con la condición *rest-to-rest* se asumen que las velocidades de partida y llegada en cualquier componente x, y & z sean igual a cero. ($Vx0 = Vy0 = Vz0 = Vxf = Vyf = Vzf = 0$).

  De los sistemas anteriores se obtiene la siguiente equivalencia . . .

$$\begin{bmatrix} t_0^3 & t_0^2 & t_0 & 1 \\ 3t_0^2 & 2t_0 & 1 & 0 \\ t_f^3 & t_f^2 & t_f & 1 \\ 3t_f^2 & 2t_f & 1 & 0 \end{bmatrix} \begin{bmatrix} ax \\ bx \\ cx \\ dx \end{bmatrix} = \begin{bmatrix} x_0 \\ v_{x0} \\ x_f \\ v_{xf} \end{bmatrix} \qquad \begin{bmatrix} t_0^3 & t_0^2 & t_0 & 1 \\ 3t_0^2 & 2t_0 & 1 & 0 \\ t_f^3 & t_f^2 & t_f & 1 \\ 3t_f^2 & 2t_f & 1 & 0 \end{bmatrix} \begin{bmatrix} ay \\ by \\ cy \\ dy \end{bmatrix} = \begin{bmatrix} y_0 \\ v_{y0} \\ y_f \\ v_{yf} \end{bmatrix} \qquad \begin{bmatrix} t_0^3 & t_0^2 & t_0 & 1 \\ 3t_0^2 & 2t_0 & 1 & 0 \\ t_f^3 & t_f^2 & t_f & 1 \\ 3t_f^2 & 2t_f & 1 & 0 \end{bmatrix} \begin{bmatrix} az \\ bz \\ cz \\ dz \end{bmatrix} = \begin{bmatrix} z_0 \\ v_{z0} \\ z_f \\ v_{zf} \end{bmatrix}$$

  Por lo tanto, los coeficientes son el resultado de los siguientes cálculos:

$$\begin{bmatrix} ax \\ bx \\ cx \\ dx \end{bmatrix} = \begin{pmatrix} t_0^3 & t_0^2 & t_0 & 1 \\ 3t_0^2 & 2t_0 & 1 & 0 \\ t_f^3 & t_f^2 & t_f & 1 \\ 3t_f^2 & 2t_f & 1 & 0 \end{pmatrix}^{-1} \cdot \begin{bmatrix} x_0 \\ v_{x0} \\ x_f \\ v_{xf} \end{bmatrix} \qquad \begin{bmatrix} ay \\ by \\ cy \\ dy \end{bmatrix} = \begin{pmatrix} t_0^3 & t_0^2 & t_0 & 1 \\ 3t_0^2 & 2t_0 & 1 & 0 \\ t_f^3 & t_f^2 & t_f & 1 \\ 3t_f^2 & 2t_f & 1 & 0 \end{pmatrix}^{-1} \cdot \begin{bmatrix} y_0 \\ v_{y0} \\ y_f \\ v_{yf} \end{bmatrix} \qquad \begin{bmatrix} az \\ bz \\ cz \\ dz \end{bmatrix} = \begin{pmatrix} t_0^3 & t_0^2 & t_0 & 1 \\ 3t_0^2 & 2t_0 & 1 & 0 \\ t_f^3 & t_f^2 & t_f & 1 \\ 3t_f^2 & 2t_f & 1 & 0 \end{pmatrix}^{-1} \cdot \begin{bmatrix} z_0 \\ v_{z0} \\ z_f \\ v_{zf} \end{bmatrix}$$

- Implementación en C++

  Función que define que se llevaran a cabo trayectorias cubicas en función del tiempo.

```
458  void Robot::Parametrica2(){
478
479      /// Parametrica para rectas con analisis de coeficientes
480      xO6 = ax*t*t*t + bx*t*t + cx*t + dx;
481      vxO6 = 3*ax*t*t + 2*bx*t + cx;
482      axO6 = 6*ax*t + 2*bx;
483
484      /*
485      vxO6 = 0;
486      axO6 = 0;
487      */
488
489      yO6 = ay*t*t*t + by*t*t + cy*t + dy;
490      vyO6 = 3*ay*t*t + 2*by*t + cy;
```

```
491       ayO6 = 6*ay*t + 2*by;
492
493       /*
494       vyO6 = 0;
495       ayO6 = 0;
496       */
497       zO6 = az*t*t*t + bz*t*t + cz*t + dz;
498       vzO6 = 3*az*t*t + 2*bz*t + cz;
499       azO6 = 6*az*t + 2*bz;
500
501       /*
502       vzO6 = 0;
503       azO6 = 0;
504       */
505       rO6.entry(0,0)=xO6;
506       rO6.entry(1,0)=yO6;
507       rO6.entry(2,0)=zO6;
508   }
509
```

Función para guardar y establecer el punto P$(x0, y0, z0)$ y punto Q(xf, yf, zf). Esta función permite guardar más de dos puntos (A,B, C, …) para una trayectoria compuesta si se quiere.

```
939   void Robot::guardarPosActual(){
940       bool stateExist = false;
941       float tPaP;
942       if(Trayecto.size()>0)
943           if(Trayecto[Trayecto.size()-1].x == xO6 && Trayecto[Trayecto.size()-1].y == yO6 &&
Trayecto[Trayecto.size()-1].z == zO6){
944               stateExist = true;
945           }
946       if(stateExist == false){
947           if(Trayecto.size()==0)
948               tPaP = 0;
949           else
950               tPaP = 3;
951
952           Trayecto.push_back(vector3d(xO6,yO6,zO6,tPaP));
953           cout << "============================" << endl;
954           cout << "||    Posicion Guardada   ||" << endl;
955           cout << "============================" << endl;
956       }
957       else{
958           cout << "========================================" << endl;
959           cout << "||    Ya se ha guardado esta posicion ||" << endl;
960           cout << "========================================" << endl;
961       }
962       cout << "-------------------------------------------------------------------------" << endl;
963   }
```

Función para resetear la pila de puntos guardados para la trayectoria.

```
934   void Robot::resetTrayectorias(){
935       Trayecto.clear();
936       cout << "Trayectoria Eliminada" << endl;
937       cout << "-------------------------------------------------------------------------" << endl;
938   }
```

Función que calcula los coeficientes entre dos puntos guardados (T1, T2)

```
964   void Robot::planTrayectoria(vector3d &T1, vector3d &T2){  // Funciones para planificar y dibujar
trayectorias
965       Matrix A(4,4);
966       t0=0; tf=T2.t;
967
968       x0 = T1.x;    xf = T2.x;     v0x = 0;     vfx = 0;
969       y0 = T1.y;    yf = T2.y;     v0y = 0;     vfy = 0;
970       z0 = T1.z;    zf = T2.z;     v0z = 0;     vfz = 0;
971
972       A.entry(0,0) = t0*t0*t0;     A.entry(0,1) = t0*t0;       A.entry(0,2) = t0;      A.entry(0,3) = 1;
973       A.entry(1,0) = 3*t0*t0;      A.entry(1,1) = 2*t0;        A.entry(1,2) = 1;       A.entry(1,3) = 0;
974       A.entry(2,0) = tf*tf*tf;     A.entry(2,1) = tf*tf;       A.entry(2,2) = tf;      A.entry(2,3) = 1;
975       A.entry(3,0) = 3*tf*tf;      A.entry(3,1) = 2*tf;        A.entry(3,2) = 1;       A.entry(3,3) = 0;
```

```
976
977     Matrix X(4,1), Cx(4,1);
978     Cx.entry(0,0) = x0;
979     Cx.entry(1,0) = v0x;
980     Cx.entry(2,0) = xf;
981     Cx.entry(3,0) = vfx;
982
983     Matrix Y(4,1), Cy(4,1);
984     Cy.entry(0,0) = y0;
985     Cy.entry(1,0) = v0y;
986     Cy.entry(2,0) = yf;
987     Cy.entry(3,0) = vfy;
988
989     Matrix Z(4,1), Cz(4,1);
990     Cz.entry(0,0) = z0;
991     Cz.entry(1,0) = v0z;
992     Cz.entry(2,0) = zf;
993     Cz.entry(3,0) = vfz;
994
995     X = A.inversa()*Cx;
996     ax=X.entry(0,0);
997     bx=X.entry(1,0);
998     cx=X.entry(2,0);
999     dx=X.entry(3,0);
1000
1001    cout << "====================================" << endl;
1002    cout << "||  Los coeficientes para X son  ||" << endl;
1003    cout << "====================================" << endl;
1004    X.mostrar();
1005
1006    Y = A.inversa()*Cy;
1007    ay=Y.entry(0,0);
1008    by=Y.entry(1,0);
1009    cy=Y.entry(2,0);
1010    dy=Y.entry(3,0);
1011
1012    cout << "====================================" << endl;
1013    cout << "||  Los coeficientes para Y son  ||" << endl;
1014    cout << "====================================" << endl;
1015    Y.mostrar();
1016
1017    Z = A.inversa()*Cz;
1018    az=Z.entry(0,0);
1019    bz=Z.entry(1,0);
1020    cz=Z.entry(2,0);
1021    dz=Z.entry(3,0);
1022
1023    cout << "====================================" << endl;
1024    cout << "||  Los coeficientes para Z son  ||" << endl;
1025    cout << "====================================" << endl;
1026    Z.mostrar();
1027
1028 }
```

Función que administra los coeficientes a utilizar entre dos puntos de la trayectoria y el cambio de puntos.

```
884 void Robot::mover(){
885
886     if(Trayecto.size()>=2 && pTrayec < Trayecto.size()-1 && trabajando == true){
887
888         planTrayectoria(Trayecto[pTrayec],Trayecto[pTrayec+1]);
889         CinematicaInversa2();
890         t =t+0.1;
891         if(t>tf) {pTrayec++; t=0;}
892     }else{
893         pTrayec = 0;
894         trabajando = false;
895     }
896
897 }
```

2. **Calcular las velocidades angulares relativas para cada grado de libertad del manipulador en función del tiempo.**

Solución:

- Utilizar el método expuesto en clase, utilizar software de computo simbólico para el cálculo de las derivadas totales de las matrices que resulten del análisis.
  Si se sabe que la posición relativa del punto del Origen O4 se puede encontrar con el punto del Origen O6, con la siguiente relación:

$$rO4 = rO6 - d\hat{k}$$

Con $d = cte$ y $\hat{k} = (0,0,1)$. Entonces se puede realizar el siguiente analisis que proporciona la velocidad lineal de los puntos.

$$\frac{d(rO4)}{dt} = vO6$$

Si se sabe que el cálculo de los coeficientes proporciona la Velocidad instantánea al llamar la función de paramétrica en el origen O6 para cada componente en x, y & z. Se tiene.

$$\frac{d(rO4)}{dt} = \begin{bmatrix} vx_{O6} \\ vy_{O6} \\ vy_{O6} \end{bmatrix}$$

Al calcular el rO4 en base a las transformaciones de rO4 = T01*T12*T23*RO4, se obtiene.

$$r_{o4} = \begin{bmatrix} x_2 \cos q_1 \cos q_2 - z_4 \cos q_1 \cos q_2 \sin q_3 - z_4 \cos q_1 \cos q_3 \sin q_2 \\ x_2 \cos q_2 \sin q_1 - z_4 \cos q_2 \sin q_1 \sin q_3 - z_4 \cos q_3 \sin q_1 \sin q_2 \\ z_1 - x_2 \sin q_2 - z_4 \cos q_2 \cos q_3 + z_4 \sin q_2 \sin q_3 \\ 1 \end{bmatrix}$$

Y que al derivar respecto al tiempo se obtiene el Jacobiano de rO4 por los componentes de las velocidades angulares de los eslabones del 1-3.

$JrO4=$

$$\begin{bmatrix} z_4 \cos q_2 \sin q_1 \sin q_3 - x_2 \cos q_2 \sin q_1 + z_4 \cos q_3 \sin q_1 \sin q_2 & z_4 \cos q_1 \sin q_2 \sin q_3 - z_4 \cos q_1 \cos q_2 \cos q_3 - x_2 \cos q_1 \sin q_2 & z_4 \cos q_1 \sin q_2 \sin q_3 - z_4 \cos q_1 \cos q_2 \cos q_3 \\ x_2 \cos q_1 \cos q_2 - z_4 \cos q_1 \cos q_2 \sin q_3 - z_4 \cos q_1 \cos q_3 \sin q_2 & z_4 \sin q_1 \sin q_2 \sin q_3 - z_4 \cos q_2 \cos q_3 \sin q_1 - x_2 \sin q_1 \sin q_2 & z_4 \sin q_1 \sin q_2 \sin q_3 - z_4 \cos q_2 \cos q_3 \sin q_1 \\ 0 & z_4 \cos q_2 \sin q_3 - x_2 \cos q_2 + z_4 \cos q_3 \sin q_2 & z_4 \cos q_2 \sin q_3 + z_4 \cos q_3 \sin q_2 \end{bmatrix}$$

El cual resulta multiplicado por el vector w13.

$$\begin{bmatrix} z_4 \cos q_2 \sin q_1 \sin q_3 - x_2 \cos q_2 \sin q_1 + z_4 \cos q_3 \sin q_1 \sin q_2 & z_4 \cos q_1 \sin q_2 \sin q_3 - z_4 \cos q_1 \cos q_2 \cos q_3 - x_2 \cos q_1 \sin q_2 & z_4 \cos q_1 \sin q_2 \sin q_3 - z_4 \cos q_1 \cos q_2 \cos q_3 & \dot{q}_1 \\ x_2 \cos q_1 \cos q_2 - z_4 \cos q_1 \cos q_2 \sin q_3 - z_4 \cos q_1 \cos q_3 \sin q_2 & z_4 \sin q_1 \sin q_2 \sin q_3 - z_4 \cos q_2 \cos q_3 \sin q_1 - x_2 \sin q_1 \sin q_2 & z_4 \sin q_1 \sin q_2 \sin q_3 - z_4 \cos q_2 \cos q_3 \sin q_1 & \dot{q}_2 \\ 0 & z_4 \cos q_2 \sin q_3 - x_2 \cos q_2 + z_4 \cos q_3 \sin q_2 & z_4 \cos q_2 \sin q_3 + z_4 \cos q_3 \sin q_2 & \dot{q}_3 \end{bmatrix}$$

Entonces se puede decir que:

$$(J_{rO4}) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \begin{pmatrix} vx_{O6} \\ vy_{O6} \\ vz_{O6} \end{pmatrix}$$

Al despejar el vector de velocidades angulares $< \dot{q}_1, \dot{q}_2, \dot{q}_3 >$ se tiene.

$$\dot{W}_{13} = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = (J_{TO4})^{-1} \begin{pmatrix} vx_{O6} \\ vy_{O6} \\ vz_{O6} \end{pmatrix}$$

Esta equivalencia ya se puede procesar en C++, puesto que se conocen los datos.
Para encontrar las velocidades angulares $< \dot{q}_4, \dot{q}_5, \dot{q}_6 >$ se toma como base.

$$\left({}^{0}_{3}R^{-1}\right){}^{0}_{6}R = {}^{3}_{6}R$$

Siendo R06 = R01*R12*R23*R34*R45*R56, haciendo alusión a R_{ij} como el producto de las rotaciones de z y x en una matriz 3x3. Y suponiendo que R06 contiene en el momento del cálculo puras constantes puesto que se conoce la posición de las articulaciones del robot, se tiene.

$${}^{0}_{6}R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

Entonces al encontrar R03

$$R03 = \begin{pmatrix} \cos q_1 & 0 & -\sin q_1 \\ \sin q_1 & 0 & \cos q_1 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} \cos q_2 & -\sin q_2 & 0 \\ \sin q_2 & \cos q_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos q_3 & 0 & -\sin q_3 \\ \sin q_3 & 0 & \cos q_3 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} \cos q_1 \cos q_2 \cos q_3 - \cos q_1 \sin q_2 \sin q_3 & \sin q_1 & -\cos q_1 \cos q_2 \sin q_3 - \cos q_1 \cos q_3 \sin q_2 \\ \cos q_2 \cos q_3 \sin q_1 - \sin q_1 \sin q_2 \sin q_3 & -\cos q_1 & -\cos q_2 \sin q_1 \sin q_3 - \cos q_3 \sin q_1 \sin q_2 \\ -\cos q_2 \sin q_3 - \cos q_3 \sin q_2 & 0 & \sin q_2 \sin q_3 - \cos q_2 \cos q_3 \end{pmatrix}$$

Se obtiene

$${}^{3}_{6}R = \begin{pmatrix} \cos q_1 \cos q_2 \cos q_3 - \cos q_1 \sin q_2 \sin q_3 & \sin q_1 & -\cos q_1 \cos q_2 \sin q_3 - \cos q_1 \cos q_3 \sin q_2 \\ \cos q_2 \cos q_3 \sin q_1 - \sin q_1 \sin q_2 \sin q_3 & -\cos q_1 & -\cos q_2 \sin q_1 \sin q_3 - \cos q_3 \sin q_1 \sin q_2 \\ -\cos q_2 \sin q_3 - \cos q_3 \sin q_2 & 0 & \sin q_2 \sin q_3 - \cos q_2 \cos q_3 \end{pmatrix}^{-1} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

$${}^{3}_{6}R = \begin{pmatrix} \frac{1}{2}\cos(q_1 - q_2 - q_3) + \frac{1}{2}\cos(q_1 + q_2 + q_3) & \frac{1}{2}\sin(q_1 - q_2 - q_3) + \frac{1}{2}\sin(q_1 + q_2 + q_3) & -\sin(q_2 + q_3) \\ \sin q_1 & -\cos q_1 & 0 \\ \frac{1}{2}\sin(q_1 - q_2 - q_3) - \frac{1}{2}\sin(q_1 + q_2 + q_3) & \frac{1}{2}\cos(q_1 + q_2 + q_3) - \frac{1}{2}\cos(q_1 - q_2 - q_3) & -\cos(q_2 + q_3) \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

Y si por otra parte R36 es el producto de R34*R45*R56.

$${}^{3}_{6}R = \begin{pmatrix} \cos q_4 & 0 & -\sin q_4 \\ \sin q_4 & 0 & \cos q_4 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} \cos q_5 & 0 & -\sin q_5 \\ \sin q_5 & 0 & \cos q_5 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \cos q_6 & -\sin q_6 & 0 \\ \sin q_6 & \cos q_6 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \sin q_4 \sin q_6 + \cos q_4 \cos q_5 \cos q_6 & \cos q_6 \sin q_4 - \cos q_4 \cos q_5 \sin q_6 & -\cos q_4 \sin q_5 \\ \cos q_5 \cos q_6 \sin q_4 - \cos q_4 \sin q_6 & -\cos q_4 \cos q_6 - \cos q_5 \sin q_4 \sin q_6 & -\sin q_4 \sin q_5 \\ -\cos q_6 \sin q_5 & \sin q_5 \sin q_6 & -\cos q_5 \end{pmatrix}$$

Se tiene

$$\begin{pmatrix} \frac{1}{2}\cos(q_1 - q_2 - q_3) + \frac{1}{2}\cos(q_1 + q_2 + q_3) & \frac{1}{2}\sin(q_1 - q_2 - q_3) + \frac{1}{2}\sin(q_1 + q_2 + q_3) & -\sin(q_2 + q_3) \\ \sin q_1 & -\cos q_1 & 0 \\ \frac{1}{2}\sin(q_1 - q_2 - q_3) - \frac{1}{2}\sin(q_1 + q_2 + q_3) & \frac{1}{2}\cos(q_1 + q_2 + q_3) - \frac{1}{2}\cos(q_1 - q_2 - q_3) & -\cos(q_2 + q_3) \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} \sin q_4 \sin q_6 + \cos q_4 \cos q_5 \cos q_6 & \cos q_6 \sin q_4 - \cos q_4 \cos q_5 \sin q_6 & -\cos q_4 \sin q_5 \\ \cos q_5 \cos q_6 \sin q_4 - \cos q_4 \sin q_6 & -\cos q_4 \cos q_6 - \cos q_5 \sin q_4 \sin q_6 & -\sin q_4 \sin q_5 \\ -\cos q_6 \sin q_5 & \sin q_5 \sin q_6 & -\cos q_5 \end{pmatrix}$$

Por algebra lineal se sabe que una matriz A_{n*m} es igual a una matriz B_{n*m}, si y sólo si las componentes de A son igual a las componentes de B. Dicho lo anterior se eligen 3 posiciones de las matrices de ambos lados y se igualan.

$$\begin{pmatrix} f1(q_1, q_2, q_3) \\ f2(q_1, q_2, q_3) \\ f3(q_1, q_2, q_3) \end{pmatrix} = \begin{pmatrix} g1(q_4, q_5, q_6) \\ g2(q_4, q_5, q_6) \\ g3(q_4, q_5, q_6) \end{pmatrix}$$

Una vez apiladas se les puede sacar el jacobiano el cual resulta en los siguiente.

$$J_F \dot{W}_{13} = J_G \dot{W}_{36}, \quad \dot{W}_{13} = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}, \dot{W}_{36} = \begin{pmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{pmatrix}$$

Al despejar se tiene

$$A_2\dot{W}_{13} = \dot{W}_{36} \quad con \quad A_2 = \left(J_G^{-1}\right)J_F$$

Por cuestiones de simplicidad, se calculó el Jacobiano de F y de G en Matlab. Para su comprobación se calculo también A2 y se introdujo en el código.

>> Código Matlab

```
function A=Jacobian
syms t1 t2 t3 t4 t5 t6 x1 x2 x3 x4 x5 x6 z1 z2 z3 z4 z5 z6 q5
syms r00 r01 r02 r10 r11 r12 r20 r21 r22 t1v t2v t3v t4v t5v t6v
T01=[cos(t1) 0 -sin(t1) 0;sin(t1) 0 cos(t1) 0;0 -1 0 z1;0 0 0 1];
T12=[cos(t2) -sin(t2) 0 x2*cos(t2);sin(t2) cos(t2) 0 x2*sin(t2);0 0 1 0; 0 0 0 1];
T23=[cos(t3) 0 -sin(t3) 0;sin(t3) 0 cos(t3) 0; 0 -1 0 0;0 0 0 1];

T34=[cos(t4) 0 -sin(t4) 0;sin(t4) 0 cos(t4) 0;0 -1 0 z4;0 0 0 1];
T45=[cos(t5) 0 -sin(t5) 0;sin(t5) 0 cos(t5) 0;0 -1 0 0;0 0 0 1];
T56=[cos(t6) -sin(t6) 0 0;sin(t6) cos(t6) 0 0; 0 0 1 z6;0 0 0 1];

T02=simplify(T01*T12);
T03=simplify(T02*T23);
T13=simplify(T12*T23);

T04=simplify(T03*T34);
T14=simplify(T13*T34);
T24=simplify(T23*T34);

T05=simplify(T04*T45);
T15=simplify(T14*T45);
T25=simplify(T24*T45);
T35=simplify(T34*T45);

T06=simplify(T05*T56);
T16=simplify(T15*T56);
T26=simplify(T25*T56);
T36=simplify(T35*T56);

R36=T36(1:3,1:3);
R03=T03(1:3,1:3);
R03inv=simplify(inv(R03));
R06=[r00 r01 r02;r10 r11 r12;r20 r21 r22];
C=R03inv*R06;
f1=C(1,1);
g1=R36(1,1);
f2=C(3,1);
g2=R36(3,1);
f3=C(2,3);
g3=R36(2,3);

F=[f1;f2;f3];
G=[g1;g2;g3];

Jg=jacobian(G,[t4,t5,t6])
%ccode(Jg);
Jf=jacobian(F,[t1,t2,t3])
%ccode(Jf);
Jginv=simplify(inv(Jg));
A2=simplify(Jginv*Jf);
%ccode(A2);

End
```

- Implementación en C++

Implementación en Cinematica inversa 1 para las primeras Velocidades angulares contenidas en el $\dot{W}_{13}$

```
257  bool Robot::CinematicaInversa1() . . .
```

```
276
277         /// Analisis de Velocidades Angulares [Eslabones 1-3]
```

```
278
279         Velocidad13.entry(0,0)=vxO6;
280         Velocidad13.entry(1,0)=vyO6;
281         Velocidad13.entry(2,0)=vzO6;
282
283         VAngular13 = Jacobiano13(theta1,theta2,theta3).inversa()*Velocidad13;
284
285         omega1 = VAngular13.entry(0,0);
286         omega2 = VAngular13.entry(1,0);
287         omega3 = VAngular13.entry(2,0);
          . . . }
```

Implementación en Cinematica inversa 2 para las Velocidades angulares contenidas en el $\dot{W}_{36}$. Se presentan dos formas de calcularlo, la pimera con los A2 la cual ya ha sido sintetizada del cálculo de $(Jg^{-1})(Jf)$ y la otra con $Jg\ y\ Jf$ de manera explicita sin operación.

```
314 bool Robot::CinematicaInversa2(){ . . .

379     for(int fil = 0; fil < 3; fil++){
380         for(int col = 0; col < 3; col++){
381             R06.entry(fil, col) = T06.entry(fil, col);
382         }
383     }
384
385     Matrix _A2(3,3);
386     _A2 = A2(theta1, theta2, theta3, theta4, theta5, theta6, R06);
387
388     //Metodo 1
389     VAngular46 = _A2*VAngular13;
390
391     //Metodo 2
392     //VAngular46 = (Jg(theta4, theta5, theta6).inversa()*Jf(theta1, theta2, theta3, R06))*VAngular13;
393
394     omega4 = VAngular46.entry(0,0);
395     omega5 = VAngular46.entry(1,0);
396     omega6 = VAngular46.entry(2,0);
397
      . . . }
```

Jacobiano13 equivalente a J_{rO4}

```
743 Matrix Robot::Jacobiano13(float x, float y, float z){   // Funciones para Cinematica Inversa, Velocidades
y Aceleraciones Angulares [1-3] . . .
744     Matrix J(3,3);
745     J.entry(0,0)=(z4*cos(y)*sin(x)*sin(z))-1*(x2*cos(y)*sin(x))+(z4*cos(z)*sin(x)*sin(y));
746     J.entry(1,0)=(x2*cos(x)*cos(y))-1*(z4*cos(x)*cos(y)*sin(z))-1*(z4*cos(x)*cos(z)*sin(y));
747     J.entry(2,0)=0;
748
749     J.entry(0,1)=(z4*cos(x)*sin(y)*sin(z))-1*(z4*cos(x)*cos(y)*cos(z))-1*(x2*cos(x)*sin(y));
750     J.entry(1,1)=(z4*sin(x)*sin(y)*sin(z))-1*(z4*cos(y)*cos(z)*sin(x))-1*(x2*sin(x)*sin(y));
751     J.entry(2,1)=(z4*cos(y)*sin(z))-1*(x2*cos(y))+(z4*cos(z)*sin(y));
752
753     J.entry(0,2)=(z4*cos(x)*sin(y)*sin(z))-1*(z4*cos(x)*cos(y)*cos(z));
754     J.entry(1,2)=(z4*sin(x)*sin(y)*sin(z))-1*(z4*cos(y)*cos(z)*sin(x));
755     J.entry(2,2)=(z4*cos(y)*sin(z))+(z4*cos(z)*sin(y));
756     return J;
757 }
```

Jacobiano de G

```
809 Matrix Robot::Jg(float t4, float t5, float t6){
810     Matrix jg(3,3);
811
812     jg.entry(0,0) = cos(t4)*sin(t6)-cos(t5)*cos(t6)*sin(t4);
813     jg.entry(0,1) = -cos(t4)*cos(t6)*sin(t5);
814     jg.entry(0,2) = cos(t6)*sin(t4)-cos(t4)*cos(t5)*sin(t6);
815     jg.entry(1,0) = 0;
816     jg.entry(1,1) = -cos(t5)*cos(t6);
817     jg.entry(1,2) = sin(t5)*sin(t6);
```

```
818       jg.entry(2,0) = -cos(t4)*sin(t5);
819       jg.entry(2,1) = -cos(t5)*sin(t4);
820       jg.entry(2,2) = 0;
821
822       return jg;
823   }
```

Jacobiano de F

```
824   Matrix Robot::Jf(float t1, float t2, float t3, Matrix R){

825       Matrix jf(3,3);
826       jf.zero(3,3);
827       if(R.n == 3 && R.m == 3){
828
829       jf.entry(0,0) = R.entry(1,0)*cos(t2+t3)*cos(t1)-R.entry(0,0)*cos(t2+t3)*sin(t1);
830       jf.entry(0,1) = -R.entry(2,0)*cos(t2+t3)-R.entry(0,0)*sin(t2+t3)*cos(t1)-
                          R.entry(1,0)*sin(t2+t3)*sin(t1);
831       jf.entry(0,2) = -R.entry(2,0)*cos(t2+t3)-R.entry(0,0)*sin(t2+t3)*cos(t1)-
                          R.entry(1,0)*sin(t2+t3)*sin(t1);
832       jf.entry(1,0) = -R.entry(1,0)*sin(t2+t3)*cos(t1)+R.entry(0,0)*sin(t2+t3)*sin(t1);
833       jf.entry(1,1) = R.entry(2,0)*sin(t2+t3)-R.entry(0,0)*cos(t2+t3)*cos(t1)-
                          R.entry(1,0)*cos(t2+t3)*sin(t1);
834       jf.entry(1,2) = R.entry(2,0)*sin(t2+t3)-R.entry(0,0)*cos(t2+t3)*cos(t1)-
                          R.entry(1,0)*cos(t2+t3)*sin(t1);
835       jf.entry(2,0) = R.entry(0,2)*cos(t1)+R.entry(1,2)*sin(t1);
836       jf.entry(2,1) = 0;
837       jf.entry(2,2) = 0;
838
839       }
840       return jf;
841   }
842
```

Función equivalente a A2, la cual se encuentra sintetizada y sirve para encontrar las aceleraciones angulares.

```
773   Matrix Robot::A2(float t1, float t2, float t3, float t4, float t5, float t6, Matrix R){ // Funciones para
Cinematica Inversa, Velocidades y Aceleraciones Angulares [4-6]
 774       Matrix a2;
 775       a2.zero(3,3);
 776       if(R.n == 3 && R.m == 3){
 777
 778       a2.entry(0,0) = -
(R.entry(0,2)*cos(t1)*cos(t4)*cos(t6)*sin(t6)+R.entry(1,2)*cos(t4)*cos(t6)*sin(t1)*sin(t6)-
R.entry(0,2)*cos(t1)*cos(t5)*pow(cos(t6),2.0)*sin(t4)-R.entry(1,2)*cos(t5)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*cos(t6)*pow(sin(t4),2.0)+R.entry(0,0)*sin(t2+t3)*cos(t5)*cos(t6)*sin(t1)
*pow(sin(t4),2.0)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*sin(t4)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*pow(cos(t5),2.0)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*
sin(t5)*sin(t6))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)));
 779       a2.entry(0,1) = -
(cos(t5)*sin(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)));
 780       a2.entry(0,2) = -
(cos(t5)*sin(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)));
 781       a2.entry(1,0) = -
(R.entry(0,2)*cos(t1)*cos(t4)*pow(sin(t6),2.0)+R.entry(1,2)*cos(t4)*sin(t1)*pow(sin(t6),2.0)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t6)*sin(t1)*sin(t4)-R.entry(0,0)*cos(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)-
R.entry(0,2)*cos(t1)*cos(t5)*cos(t6)*sin(t4)*sin(t6)-R.entry(1,2)*cos(t5)*cos(t6)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t5
```

```
)*sin(t1)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t6)*sin(t4)+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4
)*sin(t5)*sin(t6))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6));
 782       a2.entry(1,1) =
(cos(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6));
 783       a2.entry(1,2) =
(cos(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6));
 784       a2.entry(2,0) = (R.entry(1,2)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t5),2.0)*cos(t6
)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t6)*sin(t1)-
R.entry(0,0)*sin(t2+t3)*pow(cos(t5),2.0)*cos(t6)*sin(t1)+R.entry(0,2)*cos(t1)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*
sin(t4)-R.entry(0,2)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t6)-
R.entry(1,2)*cos(t4)*cos(t5)*cos(t6)*sin(t1)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*cos(t6)*
sin(t1)*sin(t5)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*
sin(t4)*sin(t6))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)));
 785       a2.entry(2,1) = -((pow(cos(t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),
2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t4)*cos(t5
)*cos(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(pow(cos(t4
),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6));
 786       a2.entry(2,2) = -((pow(cos(t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),
2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t4)*cos(t5
)*cos(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(pow(cos(t4
),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6));
 787       }
 788       return a2;
 789   }
```

3. **Calcular las aceleraciones angulares relativas para cada grado de libertad del manipulador en función del tiempo.**

Solución:

- Utilizar el método expuesto en clase, utilizar software de computo simbólico para el cálculo de las derivadas totales de las matrices que resulten del análisis.
  Si se sabe que la posición relativa del punto del Origen O4 se puede encontrar con el punto del Origen O6, con la siguiente relación:

  Si del analisis anterior se sabe que:

$$(J_{TO4})\begin{pmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{pmatrix}=\begin{pmatrix}vx_{O6}\\ vy_{O6}\\ vy_{O6}\end{pmatrix} \qquad (J_{TO4})(\dot{W}_{13})=\begin{pmatrix}vx_{O6}\\ vy_{O6}\\ vz_{O6}\end{pmatrix},$$

  Entonces al conocer las aceleraciones lineales de la trayectoria se pueden encontrar las aceleraciones angulares para los primeros 3 eslabones.

$$\frac{d(J_{rO4})}{dt}(\dot{W}_{13}) + (J_{rO4})\frac{d(\dot{W}_{13})}{dt} = \begin{pmatrix} ax_{O6} \\ ay_{O6} \\ az_{O6} \end{pmatrix}, \quad \text{Si } \frac{d(\dot{W}_{13})}{dt} = \dot{\alpha}_{13}$$

Entonces

$$\frac{d(J_{rO4})}{dt}(\dot{W}_{13}) + (J_{rO4})(\dot{\alpha}_{13}) = \begin{pmatrix} ax_{O6} \\ ay_{O6} \\ az_{O6} \end{pmatrix}$$

Si se dice que $J_{rO4} = H_{3x3}$, entonces se asume que:

$$\frac{d(J_{rO4})}{dt} = \frac{d(H)}{dt} = \begin{bmatrix} \frac{d(H_{11})}{dt} & \frac{d(H_{12})}{dt} & \frac{d(H_{13})}{dt} \\ \frac{d(H_{21})}{dt} & \frac{d(H_{22})}{dt} & \frac{d(H_{23})}{dt} \\ \frac{d(H_{31})}{dt} & \frac{d(H_{32})}{dt} & \frac{d(H_{33})}{dt} \end{bmatrix}$$

Y si se sabe que

$$\frac{d(H_{ij})}{dt} = J(H_{ij}).(\dot{W}_{13})$$

Entonces

$$\frac{d(J_{rO4})}{dt} = \begin{bmatrix} J(H_{11}).(\dot{W}_{13}) & J(H_{12}).(\dot{W}_{13}) & J(H_{13}).(\dot{W}_{13}) \\ J(H_{21}).(\dot{W}_{13}) & J(H_{22}).(\dot{W}_{13}) & J(H_{23}).(\dot{W}_{13}) \\ J(H_{31}).(\dot{W}_{13}) & J(H_{32}).(\dot{W}_{13}) & J(H_{33}).(\dot{W}_{13}) \end{bmatrix}$$

Lo cual ya se puede calcular por medio de computo simbólico, en este caso Matlab.

>> Código Matlab para encontrar la derivada del Jacobiano13 o J_{rO4} respecto al tiempo

```
function J=Jacobian
syms x y z z1 z2 z3 z4 x2 dx dy dz

T01=[cos(x) 0 -sin(x) 0; sin(x) 0 cos(x) 0; 0 -1 0 z1; 0 0 0 1];

T12=[cos(y) -sin(y) 0 x2*cos(y);sin(y) cos(y) 0 x2*cos(y);0 0 1 0; 0 0 0 1];

T23=[cos(z) 0 -sin(z) 0; sin(z) 0 cos(z) 0; 0 -1 0 0; 0 0 0 1];
P = [0; 0; z4; 1];

R04= T01*T12*T23*P;

A= jacobian([R04],[x,y,z]);

qv=[dx;dy;dz];

dA00= ccode(jacobian(A(1,1),[x,y,z])*qv)
dA01= ccode(jacobian(A(1,2),[x,y,z])*qv);
dA02= ccode(jacobian(A(1,3),[x,y,z])*qv);

dA10= ccode(jacobian(A(2,1),[x,y,z])*qv);
dA11= ccode(jacobian(A(2,2),[x,y,z])*qv);
dA12= ccode(jacobian(A(2,3),[x,y,z])*qv);
```

```
dA20= ccode(jacobian(A(3,1),[x,y,z])*qv);
dA21= ccode(jacobian(A(3,2),[x,y,z])*qv);
dA22= ccode(jacobian(A(3,3),[x,y,z])*qv);

return
end
```

Así pues al despejar Alpha13 y sustituir el valor de los vectores y matrices en cada punto se logran encontrar las aceleraciones angulares de las primeras 3 articulaciones.

$$\dot{\alpha}_{13} = (J_{rO4})^{-1}\left(\begin{pmatrix} ax_{O6} \\ ay_{O6} \\ az_{O6} \end{pmatrix} - \frac{d(J_{rO4})}{dt}(\dot{W}_{13})\right) = \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix}$$

Para encontrar las aceleraciones de los siguientes grados de libertad se toma como base la relación antes mencionada para encontrar las velocidades angulares del 4to al 6to grado de libertad.

$$J_F\dot{W}_{13} = J_G\dot{W}_{36}, \qquad \dot{W}_{13} = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix}, \dot{W}_{36} = \begin{pmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{pmatrix}$$

$$A_2\dot{W}_{13} = \dot{W}_{36} \qquad con \qquad A_2 = \left(J_G^{-1}\right)J_F$$

Derivando respecto al tiempo de ambos lados se obtiene

$$A_2\left(\frac{d(\dot{W}_{13})}{dt}\right) + \frac{d(A_2)}{dt}(\dot{W}_{13}) = \frac{d(\dot{W}_{36})}{dt}$$

Si

$$\frac{d(\dot{W}_{13})}{dt} = \dot{\alpha}_{13}$$
$$\frac{d(\dot{W}_{36})}{dt} = \dot{\alpha}_{36}$$
$$\frac{d(A_2)}{dt} = \dot{A}_2$$

La ecuación se simplifica en

$$A_2\dot{\alpha}_{13} + \dot{A}_2\dot{W}_{13} = \dot{\alpha}_{36}$$

De la cual sólo hace falta conocer $\dot{A}_2$

Tomando a

$$A_2 = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Entonces se puede simplificar las columnas de la matriz en

$$\mathbf{a}_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix} \qquad \mathbf{a}_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix} \qquad \mathbf{a}_3 = \begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \end{pmatrix}$$

Entonces expandiendo se tiene que $\dot{A}_2$ es igual a

$$\frac{d}{dt}A_2 = \begin{pmatrix} \frac{d}{dt}\mathbf{a}_1 & \frac{d}{dt}\mathbf{a}_2 & \frac{d}{dt}\mathbf{a}_3 \end{pmatrix}$$

$$\frac{d}{dt}\mathbf{a}_1 = \begin{pmatrix} \frac{d(a_{11})}{dt} \\ \frac{d(a_{21})}{dt} \\ \frac{d(a_{31})}{dt} \end{pmatrix} \qquad \frac{d}{dt}\mathbf{a}_2 = \begin{pmatrix} \frac{d(a_{12})}{dt} \\ \frac{d(a_{22})}{dt} \\ \frac{d(a_{32})}{dt} \end{pmatrix} \qquad \frac{d}{dt}\mathbf{a}_3 = \begin{pmatrix} \frac{d(a_{13})}{dt} \\ \frac{d(a_{23})}{dt} \\ \frac{d(a_{33})}{dt} \end{pmatrix}$$

Si anteriormente se ha mencionado el uso del jacobiano para componentes de una matriz se puede deducir que

$$\frac{da_{ij}}{dt} = \frac{\partial a_{ij}}{\partial q_1}\dot{q}_1 + \frac{\partial a_{ij}}{\partial q_2}\dot{q}_2 + \frac{\partial a_{ij}}{\partial q_3}\dot{q}_3 + \frac{\partial a_{ij}}{\partial q_4}\dot{q}_4 + \frac{\partial a_{ij}}{\partial q_5}\dot{q}_5 + \frac{\partial a_{ij}}{\partial q_6}\dot{q}_6$$

Que es equivalente a esta notación

$$\frac{da_{ij}}{dt} = J(a_{ij}) \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{pmatrix}$$

Con ello se puede decir que $\dot{A}_2$ utilizando Jacobianos es igual a:

$$\frac{d}{dt}A_2 = \begin{pmatrix} J_{\mathbf{a}_1}\dot{\mathbf{q}} & J_{\mathbf{a}_2}\dot{\mathbf{q}} & J_{\mathbf{a}_3}\dot{\mathbf{q}} \end{pmatrix}$$

El cual se puede encontrar con la ayuda de Matlab, retomando el ejercicio para encontrar las velocidades angulares de los grados de libertad 3-6, del ejercicio 2.

>> Complementación código Matlab

```
A2=simplify(Jginv*Jf);
%ccode(A2);

a1=[A2(1,1); A2(2,1); A2(3,1)];
a2=[A2(1,2); A2(2,2); A2(3,2)];
a3=[A2(1,3); A2(2,3); A2(3,3)];

Ja1=jacobian(a1,[t1,t2,t3,t4,t5,t6]);
Ja2=jacobian(a2,[t1,t2,t3,t4,t5,t6]);
Ja3=jacobian(a3,[t1,t2,t3,t4,t5,t6]);

qv=[t1v;t2v;t3v;t4v;t5v;t6v];
dA2 =[Ja1*qv,Ja2*qv,Ja3*qv]
ccode(dA2)
```

Con lo cual ya sólo se necesitaría sustituir dA2 en para encontrar $\dot{\alpha}_{36}$

$$A_2\dot{\alpha}_{13} + \dot{A}_2\dot{W}_{13} = \dot{\alpha}_{36}$$

- Implementación en C++

Implementación en Cinematica inversa 1 para las primeras Aceleraciones angulares contenidas en el vector $\dot{\alpha}_{13}$

```
257  bool Robot::CinematicaInversa1(){ . . .
 288
 289          /// Analisis de Aceleraciones Angulares [Eslabones 1-3]
 290
 291          Aceleracion13.entry(0,0)=axO6;
 292          Aceleracion13.entry(1,0)=ayO6;
 293          Aceleracion13.entry(2,0)=azO6;
 294
 295          Matrix dA13(3,3);
 296          dA13 = dJacobian13(theta1, theta2, theta3, omega1, omega2, omega3);
 297          AAngular13 = (Jacobiano13(theta1, theta2, theta3).inversa())*(Aceleracion13 - dA13*VAngular13);
 298
 299          alpha1 = AAngular13.entry(0,0);
 300          alpha2 = AAngular13.entry(1,0);
 301          alpha3 = AAngular13.entry(2,0);
 302
 303          return true;
 304      }
.  .  .}
```

Implementación en Cinematica inversa 2 para las siguientes Aceleraciones angulares contenidas en el vector $\dot{\alpha}_{36}$

```
 314  bool Robot::CinematicaInversa2(){ . . .
 398      /// Analisis de Aceleraciones Angulares [Eslabones 4-6]
 399      Matrix DA2(3,3);
 400      DA2 = dA2(theta1, theta2, theta3, theta4, theta5, theta6, omega1, omega2, omega3, omega4, omega5,
               omega6, R06);
 401
 402      AAngular46 = _A2*AAngular13 + DA2*VAngular13;
 403      alpha4 = AAngular46.entry(0,0);
 404      alpha5 = AAngular46.entry(1,0);
 405      alpha6 = AAngular46.entry(2,0);
        . . . }
```

Función que contiene la matriz resultante del código en Matlab para $\frac{d(J_{RO4})}{dt}$

```
758  Matrix Robot::dJacobian13(float x, float y, float z, float dx, float dy, float dz){
 759      Matrix dJ(3,3);
```

```
 760       dJ.entry(0,0) = -dy*(z4*(sin(x)*sin(y)*sin(z)-cos(y)*cos(z)*sin(x))-
x2*sin(x)*sin(y))+dx*(z4*(cos(x)*cos(y)*sin(z)+cos(x)*cos(z)*sin(y))-x2*cos(x)*cos(y))-
dz*z4*(sin(x)*sin(y)*sin(z)-cos(y)*cos(z)*sin(x));
 761       dJ.entry(0,1) = -dx*(z4*(sin(x)*sin(y)*sin(z)-cos(y)*cos(z)*sin(x))-
x2*sin(x)*sin(y))+dy*(z4*(cos(x)*cos(y)*sin(z)+cos(x)*cos(z)*sin(y))-
x2*cos(x)*cos(y))+dz*z4*(cos(x)*cos(y)*sin(z)+cos(x)*cos(z)*sin(y));
 762       dJ.entry(0,2) =
dy*z4*(cos(x)*cos(y)*sin(z)+cos(x)*cos(z)*sin(y))+dz*z4*(cos(x)*cos(y)*sin(z)+cos(x)*cos(z)*sin(y))-
dx*z4*(sin(x)*sin(y)*sin(z)-cos(y)*cos(z)*sin(x));
 763
 764       dJ.entry(1,0) = dx*(z4*(cos(y)*sin(x)*sin(z)+cos(z)*sin(x)*sin(y))-
x2*cos(y)*sin(x))+dy*(z4*(cos(x)*sin(y)*sin(z)-cos(x)*cos(y)*cos(z))-
x2*cos(x)*sin(y))+dz*z4*(cos(x)*sin(y)*sin(z)-cos(x)*cos(y)*cos(z));
 765       dJ.entry(1,1) = dy*(z4*(cos(y)*sin(x)*sin(z)+cos(z)*sin(x)*sin(y))-
x2*cos(y)*sin(x))+dx*(z4*(cos(x)*sin(y)*sin(z)-cos(x)*cos(y)*cos(z))-
x2*cos(x)*sin(y))+dz*z4*(cos(y)*sin(x)*sin(z)+cos(z)*sin(x)*sin(y));
 766       dJ.entry(1,2) = dx*z4*(cos(x)*sin(y)*sin(z)-
cos(x)*cos(y)*cos(z))+dy*z4*(cos(y)*sin(x)*sin(z)+cos(z)*sin(x)*sin(y))+dz*z4*(cos(y)*sin(x)*sin(z)+cos(z)*sin(x
)*sin(y));
 767
 768       dJ.entry(2,0) = 0;
 769       dJ.entry(2,1) = dy*(z4*(cos(y)*cos(z)-sin(y)*sin(z))+x2*cos(y))+dz*z4*(cos(y)*cos(z)-sin(y)*sin(z));
 770       dJ.entry(2,2) = dy*z4*(cos(y)*cos(z)-sin(y)*sin(z))+dz*z4*(cos(y)*cos(z)-sin(y)*sin(z));
 771       return dJ;
 772   }
```

Función que contiene la matriz resultante del código de Matlab para $\dot{A}_2$.

```
790   Matrix Robot::dA2(float t1, float t2, float t3, float t4, float t5, float t6, float t1v, float t2v, float
t3v, float t4v, float t5v, float t6v, Matrix R){
 791       Matrix da2;
 792       da2.identity(3);
 793       if(R.n == 3 && R.m == 3){
 794
 795       da2.entry(0,0) =
t4v*((R.entry(0,2)*cos(t1)*cos(t6)*sin(t4)*sin(t6)+R.entry(1,2)*cos(t6)*sin(t1)*sin(t4)*sin(t6)+R.entry(0,2)*cos
(t1)*cos(t4)*cos(t5)*pow(cos(t6),2.0)+R.entry(1,2)*cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t1)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*pow(cos(t5),2.0)*sin(t6)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.
0)*pow(cos(t5),2.0)*sin(t1)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t5),2.0)*pow(sin(t4),2.0)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*pow(cos(t5),2.0)*sin(t1)*pow(sin(t4),2.0)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4
)*cos(t5)*cos(t6)*sin(t4)*2.0+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t5)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*cos(t6)*sin(t1)*sin(t4)*2.0-
R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t5)*sin(t6))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0)*(R.entry(0,2)*cos(t1)*cos(t4)*cos(t6)*sin(t6)+R.entry(1,2)*cos(t4)*cos(t6)*sin(t1)*sin(t
6)-R.entry(0,2)*cos(t1)*cos(t5)*pow(cos(t6),2.0)*sin(t4)-R.entry(1,2)*cos(t5)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*cos(t6)*pow(sin(t4),2.0)+R.entry(0,0)*sin(t2+t3)*cos(t5)*cos(t6)*sin(t1)
*pow(sin(t4),2.0)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*sin(t4)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*pow(cos(t5),2.0)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*
sin(t5)*sin(t6)))/sin(t5))-t6v*((R.entry(0,2)*cos(t1)*cos(t4)*pow(cos(t6),2.0)-
R.entry(0,2)*cos(t1)*cos(t4)*pow(sin(t6),2.0)+R.entry(1,2)*cos(t4)*pow(cos(t6),2.0)*sin(t1)-
R.entry(1,2)*cos(t4)*sin(t1)*pow(sin(t6),2.0)+R.entry(0,2)*cos(t1)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*2.0+R.entry(1
,2)*cos(t5)*cos(t6)*sin(t1)*sin(t4)*sin(t6)*2.0+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*pow(sin(t4),2.0)*sin(t6)
-
R.entry(0,0)*sin(t2+t3)*cos(t5)*sin(t1)*pow(sin(t4),2.0)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*pow(cos
(t5),2.0)*cos(t6)*sin(t4)-R.entry(0,0)*sin(t2+t3)*cos(t4)*pow(cos(t5),2.0)*cos(t6)*sin(t1)*sin(t4)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t5)*cos(t6)*sin(t4)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t5)*cos(t6)*sin(t1)*
sin(t4)*sin(t5))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0)*(R.entry(0,2)*co
s(t1)*cos(t4)*cos(t6)*sin(t6)+R.entry(1,2)*cos(t4)*cos(t6)*sin(t1)*sin(t6)-
R.entry(0,2)*cos(t1)*cos(t5)*pow(cos(t6),2.0)*sin(t4)-R.entry(1,2)*cos(t5)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
```

```
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*cos(t6)*pow(sin(t4),2.0)+R.entry(0,0)*sin(t2+t3)*cos(t5)*cos(t6)*sin(t1)
*pow(sin(t4),2.0)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*sin(t4)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*pow(cos(t5),2.0)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*
sin(t5)*sin(t6)))/sin(t5))-
t5v*((R.entry(0,2)*cos(t1)*pow(cos(t6),2.0)*sin(t4)*sin(t5)+R.entry(1,2)*pow(cos(t6),2.0)*sin(t1)*sin(t4)*sin(t5
)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*pow(cos(t5),2.0)*sin(t4)*sin(t6)+R.entry(0,0)*cos(t2+t3)*pow(cos(t5),2.0)*sin(t1
)*sin(t4)*sin(t6)+R.entry(1,0)*cos(t2+t3)*cos(t1)*sin(t4)*pow(sin(t5),2.0)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t
1)*cos(t6)*pow(sin(t4),2.0)*sin(t5)-R.entry(0,0)*cos(t2+t3)*sin(t1)*sin(t4)*pow(sin(t5),2.0)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t6)*sin(t1)*pow(sin(t4),2.0)*sin(t5)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t5)*sin(t6)*2.0+R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(
t5)*sin(t1)*sin(t4)*sin(t5)*sin(t6)*2.0)/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+((cos(t4)*sin(t
4)*sin(t5)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.
0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(R.entry(0,2
)*cos(t1)*cos(t4)*cos(t6)*sin(t6)+R.entry(1,2)*cos(t4)*cos(t6)*sin(t1)*sin(t6)-
R.entry(0,2)*cos(t1)*cos(t5)*pow(cos(t6),2.0)*sin(t4)-R.entry(1,2)*cos(t5)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*cos(t6)*pow(sin(t4),2.0)+R.entry(0,0)*sin(t2+t3)*cos(t5)*cos(t6)*sin(t1)
*pow(sin(t4),2.0)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*sin(t4)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*pow(cos(t5),2.0)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*
sin(t5)*sin(t6)))/sin(t5)-
(cos(t5)*1.0/pow(sin(t5),2.0)*(R.entry(0,2)*cos(t1)*cos(t4)*cos(t6)*sin(t6)+R.entry(1,2)*cos(t4)*cos(t6)*sin(t1)
*sin(t6)-R.entry(0,2)*cos(t1)*cos(t5)*pow(cos(t6),2.0)*sin(t4)-
R.entry(1,2)*cos(t5)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*cos(t6)*pow(sin(t4),2.0)+R.entry(0,0)*sin(t2+t3)*cos(t5)*cos(t6)*sin(t1)
*pow(sin(t4),2.0)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*sin(t4)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*pow(cos(t5),2.0)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*
sin(t5)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(t1v*(R.entry(1,2)*cos(t1)*cos(t4)*cos(t6)*sin(t6)-R.entry(0,2)*cos(t4)*cos(t6)*sin(t1)*sin(t6)-
R.entry(1,2)*cos(t1)*cos(t5)*pow(cos(t6),2.0)*sin(t4)+R.entry(0,2)*cos(t5)*pow(cos(t6),2.0)*sin(t1)*sin(t4)+R.en
try(0,0)*sin(t2+t3)*cos(t1)*cos(t5)*cos(t6)*pow(sin(t4),2.0)+R.entry(1,0)*sin(t2+t3)*cos(t5)*cos(t6)*sin(t1)*pow
(sin(t4),2.0)-R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t4)*pow(cos(t5),2.0)*sin(t1)*sin(t4)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t5)
*sin(t4)*sin(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*sin(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4)
,2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(t2v*(R.entry(1
,0)*cos(t2+t3)*cos(t1)*cos(t5)*cos(t6)*pow(sin(t4),2.0)-
R.entry(0,0)*cos(t2+t3)*cos(t5)*cos(t6)*sin(t1)*pow(sin(t4),2.0)+R.entry(0,0)*sin(t2+t3)*cos(t5)*sin(t1)*sin(t4)
*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*sin(t4)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t4)*pow(cos
(t5),2.0)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(t3v*(R.entry(1
,0)*cos(t2+t3)*cos(t1)*cos(t5)*cos(t6)*pow(sin(t4),2.0)-
R.entry(0,0)*cos(t2+t3)*cos(t5)*cos(t6)*sin(t1)*pow(sin(t4),2.0)+R.entry(0,0)*sin(t2+t3)*cos(t5)*sin(t1)*sin(t4)
*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*sin(t4)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t4)*pow(cos
(t5),2.0)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)));
 796      da2.entry(0,1) = -
t6v*((R.entry(0,2)*cos(t1)*cos(t4)*cos(t6)*sin(t6)*2.0+R.entry(1,2)*cos(t4)*cos(t6)*sin(t1)*sin(t6)*2.0-
R.entry(0,2)*cos(t1)*cos(t5)*pow(cos(t6),2.0)*sin(t4)+R.entry(0,2)*cos(t1)*cos(t5)*sin(t4)*pow(sin(t6),2.0)-
R.entry(1,2)*cos(t5)*pow(cos(t6),2.0)*sin(t1)*sin(t4)+R.entry(1,2)*cos(t5)*sin(t1)*sin(t4)*pow(sin(t6),2.0)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*sin(t4)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t4)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t5)*cos(t6)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t5
)*cos(t6)*sin(t1)+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t6)*sin(t5)-
R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t6)*sin(t1)*sin(t5))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0)*(R.entry(0,2)*co
s(t1)*cos(t4)*pow(sin(t6),2.0)+R.entry(1,2)*cos(t4)*sin(t1)*pow(sin(t6),2.0)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t6)*sin(t1)*sin(t4)-R.entry(0,0)*cos(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)-
```

```
R.entry(0,2)*cos(t1)*cos(t5)*cos(t6)*sin(t4)*sin(t6)-R.entry(1,2)*cos(t5)*cos(t6)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t5
)*sin(t1)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t6)*sin(t4)+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4
)*sin(t5)*sin(t6)))+t4v*((R.entry(0,2)*cos(t1)*sin(t4)*pow(sin(t6),2.0)+R.entry(1,2)*sin(t1)*sin(t4)*pow(sin(t6)
,2.0)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t6)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t6)*sin(t1
)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t6)*pow(sin(t4),2.0)-
R.entry(0,0)*sin(t2+t3)*cos(t6)*sin(t1)*pow(sin(t4),2.0)+R.entry(1,0)*cos(t2+t3)*cos(t1)*sin(t4)*sin(t5)*sin(t6)
+R.entry(0,2)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*sin(t1)*sin(t4)*sin(t5)*sin(t6)+R.entry(1,2)*cos(t4)*cos(t5)*cos(t6)*sin(t1)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t6)*2.0+R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(
t1)*sin(t4)*sin(t6)*2.0)/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0)*(R.entry(0,2)*cos(t1)*cos(t4)*pow(sin(t6),2.0)+R.entry(1,2)*cos(t4)*sin(t1)*pow(sin(t6),
2.0)-R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t6)*sin(t1)*sin(t4)-
R.entry(0,0)*cos(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)-R.entry(0,2)*cos(t1)*cos(t5)*cos(t6)*sin(t4)*sin(t6)-
R.entry(1,2)*cos(t5)*cos(t6)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t5
)*sin(t1)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t6)*sin(t4)+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4
)*sin(t5)*sin(t6)))+(t1v*(-
R.entry(1,2)*cos(t1)*cos(t4)*pow(sin(t6),2.0)+R.entry(0,2)*cos(t4)*sin(t1)*pow(sin(t6),2.0)+R.entry(1,0)*sin(t2+
t3)*cos(t4)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)+R.entry(1,2)*cos(t1)
*cos(t5)*cos(t6)*sin(t4)*sin(t6)-R.entry(0,2)*cos(t5)*cos(t6)*sin(t1)*sin(t4)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t5)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t5)*sin(t1)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t6)
*sin(t4)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*sin(t5)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(t2v*(R.entry(1,
0)*sin(t2+t3)*cos(t1)*cos(t4)*sin(t5)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t5)
*sin(t6)-R.entry(0,0)*cos(t2+t3)*pow(cos(t4),2.0)*cos(t5)*sin(t1)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t6)*sin(t4)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t6)*sin(t1)*sin(t4))
)/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(t3v*(R.entry(1,
0)*sin(t2+t3)*cos(t1)*cos(t4)*sin(t5)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t5)
*sin(t6)-R.entry(0,0)*cos(t2+t3)*pow(cos(t4),2.0)*cos(t5)*sin(t1)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t4)*cos(t6)*sin(t1)*sin(t4))
)/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6));
 797       da2.entry(0,2) =
t4v*((R.entry(0,2)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*pow(cos(t6),2.0)+R.entry(1,2)*cos(t4)*pow(cos(t5),2.0)*pow(c
os(t6),2.0)*sin(t1)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t6)*sin(t1)*sin(t4)*2.0+R.entry(0,2)*cos(t1)*cos(t5)*cos(t6)*sin(t4)*sin(t6)
+R.entry(1,2)*cos(t5)*cos(t6)*sin(t1)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t5
)*sin(t1)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*pow(sin(t4),2.0)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t5)*sin(t1)*pow(sin(t4),2.0)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t6)
*sin(t4)*2.0+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t5)*cos(t6)*sin(t4)*sin(t5)-
R.entry(0,0)*cos(t2+t3)*cos(t5)*cos(t6)*sin(t1)*sin(t4)*sin(t5))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(1.0/pow(pow(cos
s(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
```

```
6)*sin(t4)*sin(t6)*2.0)*(R.entry(1,2)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t5),2.0)*cos(t6
)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t6)*sin(t1)-
R.entry(0,0)*sin(t2+t3)*pow(cos(t5),2.0)*cos(t6)*sin(t1)+R.entry(0,2)*cos(t1)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*
sin(t4)-R.entry(0,2)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t6)-
R.entry(1,2)*cos(t4)*cos(t5)*cos(t6)*sin(t1)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*cos(t6)*
sin(t1)*sin(t5)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*
sin(t4)*sin(t6)))/sin(t5))-t6v*((-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t5),2.0)*sin(t6
)+R.entry(0,2)*cos(t1)*cos(t4)*cos(t5)*pow(cos(t6),2.0)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*sin(t1)*sin(t6)
-R.entry(0,0)*sin(t2+t3)*pow(cos(t5),2.0)*sin(t1)*sin(t6)-
R.entry(0,2)*cos(t1)*cos(t4)*cos(t5)*pow(sin(t6),2.0)+R.entry(1,2)*cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t1)-
R.entry(1,2)*cos(t4)*cos(t5)*sin(t1)*pow(sin(t6),2.0)+R.entry(0,2)*cos(t1)*pow(cos(t5),2.0)*cos(t6)*sin(t4)*sin(
t6)*2.0+R.entry(1,2)*pow(cos(t5),2.0)*cos(t6)*sin(t1)*sin(t4)*sin(t6)*2.0+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4
)*cos(t5)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t5)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*cos(t6)*sin(t1)*sin(t4)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t1)*
sin(t5)*sin(t6))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(1.0/pow(pow(co
s(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0)*(R.entry(1,2)*po
w(cos(t5),2.0)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t5),2.0)*cos(t6
)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t6)*sin(t1)-
R.entry(0,0)*sin(t2+t3)*pow(cos(t5),2.0)*cos(t6)*sin(t1)+R.entry(0,2)*cos(t1)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*
sin(t4)-R.entry(0,2)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t6)-
R.entry(1,2)*cos(t4)*cos(t5)*cos(t6)*sin(t1)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*cos(t6)*
sin(t1)*sin(t5)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*
sin(t4)*sin(t6)))/sin(t5))+t5v*((R.entry(0,0)*sin(t2+t3)*cos(t5)*cos(t6)*sin(t1)*sin(t5)*2.0+R.entry(0,2)*cos(t1
)*cos(t4)*cos(t6)*sin(t5)*sin(t6)+R.entry(1,2)*cos(t4)*cos(t6)*sin(t1)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*pow(cos(t5),2.0)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t4)*pow(cos(t5),2.0
)*cos(t6)*sin(t1)+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t6)*pow(sin(t5),2.0)-
R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t6)*sin(t1)*pow(sin(t5),2.0)-
R.entry(0,2)*cos(t1)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
R.entry(1,2)*cos(t5)*pow(cos(t6),2.0)*sin(t1)*sin(t4)*sin(t5)*2.0-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t5)*cos(t6)*sin(t5)*2.0-
R.entry(0,0)*sin(t2+t3)*cos(t4)*sin(t1)*sin(t4)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*sin(t4)*
sin(t5)*sin(t6))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+((cos(t4)*sin(t
4)*sin(t5)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.
0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(R.entry(1,2
)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t6)+R.entry(1,0)*sin(t2+t3)*cos(t1)*pow(cos(t5),2.0)*cos(t6
)+R.entry(0,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t6)*sin(t1)-
R.entry(0,0)*sin(t2+t3)*pow(cos(t5),2.0)*cos(t6)*sin(t1)+R.entry(0,2)*cos(t1)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*
sin(t4)-R.entry(0,2)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t6)-
R.entry(1,2)*cos(t4)*cos(t5)*cos(t6)*sin(t1)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*cos(t6)*
sin(t1)*sin(t5)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*
sin(t4)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(t1v*(-
R.entry(0,2)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*sin(t1)*sin(t4)+R.entry(0,0)*sin(t2+t3)*cos(t1)*pow(cos(t4),2.0)*
cos(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*pow(cos(t5),2.0)*cos(t6)+R.entry(1,0)*sin(t2+t3)*pow(cos(t4),2.0)*cos(t6)*sin(t1
)-
```

```
R.entry(1,0)*sin(t2+t3)*pow(cos(t5),2.0)*cos(t6)*sin(t1)+R.entry(1,2)*cos(t1)*pow(cos(t5),2.0)*pow(cos(t6),2.0)*
sin(t4)-
R.entry(1,2)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t6)+R.entry(0,2)*cos(t4)*cos(t5)*cos(t6)*sin(t1)*sin(t6)+R.entr
y(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t5)+R.entry(1,0)*cos(t2+t3)*cos(t4)*cos(t5)*cos(t6)*sin(t1
)*sin(t5)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5
)*sin(t1)*sin(t4)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(t2v*(R.entry(1,0)*cos(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*pow(cos(t5),2.0)*cos(t6)-
R.entry(0,0)*cos(t2+t3)*pow(cos(t4),2.0)*cos(t6)*sin(t1)+R.entry(0,0)*cos(t2+t3)*pow(cos(t5),2.0)*cos(t6)*sin(t1
)+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t5)-
R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t4)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*cos(t6)*
sin(t1)*sin(t5)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(t3v*(R.entry(1,0)*cos(t2+t3)*cos(t1)*pow(cos(t4),2.0)*cos(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*pow(cos(t5),2.0)*cos(t6)-
R.entry(0,0)*cos(t2+t3)*pow(cos(t4),2.0)*cos(t6)*sin(t1)+R.entry(0,0)*cos(t2+t3)*pow(cos(t5),2.0)*cos(t6)*sin(t1
)+R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t4)*sin(t6)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)*sin(t5)-
R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t4)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t4)*cos(t5)*cos(t6)*
sin(t1)*sin(t5)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)));
 798       da2.entry(1,0)  =
t5v*((sin(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(pow(cos(t5),2.0
)*sin(t4)*1.0/pow(sin(t5),2.0)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
(cos(t5)*sin(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t5)*sin(t6)+R.entr
y(1,0)*sin(t2+t3)*cos(t5)*sin(t1)*sin(t6)+R.entry(2,0)*sin(t2+t3)*cos(t4)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*sin(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(cos(t5)*sin(t4)*(cos(t4)*sin(t4)*sin(t5)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.
0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(R.entry(2,0
)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/sin(t5))-
t4v*((cos(t4)*cos(t5)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(cos(t5)*sin(t4)*(-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t6)*sin(t1)-
R.entry(2,0)*sin(t2+t3)*cos(t5)*sin(t4)*sin(t6)+R.entry(1,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*sin(t6)+R.entry(
0,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t5)*sin(t4
)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
```

```
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/sin(t5))-t6v*((cos(t5)*sin(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t5)-
R.entry(2,0)*sin(t2+t3)*sin(t4)*sin(t6)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*sin(t4)*sin(t6)+R.entry(0,0)*sin
(t2+t3)*cos(t1)*cos(t6)*sin(t5)+R.entry(1,0)*cos(t2+t3)*sin(t1)*sin(t4)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*
sin(t1)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)+R.entry(1,0)*cos(t2+t3)*cos(t4)*cos(t5)*
cos(t6)*sin(t1)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(cos(t5)*sin(t4)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0)*(R.entry(2,0)*si
n(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/sin(t5))-(t2v*cos(t5)*sin(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*sin(t5)*sin(t6)-
R.entry(2,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t6)*sin(t4)+R.entry(0,0)*cos
(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*sin(t1)*
sin(t5)*sin(t6)-R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(t3v*cos(t5)*sin(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t4)-R.entry(2,0)*sin(t2+t3)*sin(t5)*sin(t6)-
R.entry(2,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t6)*sin(t4)+R.entry(0,0)*cos
(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*sin(t1)*
sin(t5)*sin(t6)-R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(t1v*cos(t5)*si
n(t4)*(R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(0,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*sin(t1)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6))
)/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)));
 799       da2.entry(1,1) =
t5v*((cos(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t5)*sin(t6)+R.entry(1
,0)*sin(t2+t3)*cos(t5)*sin(t1)*sin(t6)+R.entry(2,0)*sin(t2+t3)*cos(t4)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*sin(t5)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+cos(t4)*(cos(t4)
*sin(t4)*sin(t5)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.
0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(R.entry(2,0
)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))+t6v*((cos(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t5)-
R.entry(2,0)*sin(t2+t3)*sin(t4)*sin(t6)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*sin(t4)*sin(t6)+R.entry(0,0)*sin
(t2+t3)*cos(t1)*cos(t6)*sin(t5)+R.entry(1,0)*cos(t2+t3)*sin(t1)*sin(t4)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*
sin(t1)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)+R.entry(1,0)*cos(t2+t3)*cos(t4)*cos(t5)*
cos(t6)*sin(t1)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
cos(t4)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0)*(R.entry(2,0)*si
n(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))-
t4v*((sin(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
```

```
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(cos(t4)*(-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t6)*sin(t1)-
R.entry(2,0)*sin(t2+t3)*cos(t5)*sin(t4)*sin(t6)+R.entry(1,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*sin(t6)+R.entry(
0,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
cos(t4)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))+(t2v*cos(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*sin(t5)*sin(t6)-
R.entry(2,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t6)*sin(t4)+R.entry(0,0)*cos
(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*sin(t1)*
sin(t5)*sin(t6)-R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(t3v*cos(t4)*(R.
entry(2,0)*cos(t2+t3)*cos(t6)*sin(t4)-R.entry(2,0)*sin(t2+t3)*sin(t5)*sin(t6)-
R.entry(2,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t6)*sin(t4)+R.entry(0,0)*cos
(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*sin(t1)*
sin(t5)*sin(t6)-R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
(t1v*cos(t4)*(R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(0,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*sin(t1)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6))
)/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6));
 800       da2.entry(1,2) = -t5v*(((cos(t5)*cos(t6)*sin(t5)*2.0-cos(t4)*sin(t4)*sin(t5)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),
2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+((pow(cos(t4),2
.0)*cos(t6)-pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1))*(cos(t4)*sin(t4)*sin(t5
)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos
(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0))/sin(t5)-
(cos(t5)*1.0/pow(sin(t5),2.0)*(pow(cos(t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(pow(cos(t4),2.0)*cos(
t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(cos(t4)*cos(t6)
*sin(t5)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(pow(cos(t4)
,2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
cos(t4)*cos(t5)*cos(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1)
)*(cos(t4)*sin(t4)*sin(t5)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+
cos(t4)*cos(t5)*sin(t4),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0))-
t1v*(((R.entry(1,0)*cos(t2+t3)*cos(t1)-R.entry(0,0)*cos(t2+t3)*sin(t1))*(pow(cos(t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(cos(t4)*cos(t5)*cos(t6)*(R.entry(1,0)*sin(t2+t3)*cos(t1)-
R.entry(0,0)*sin(t2+t3)*sin(t1)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+t2v*(((pow(cos(
t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t
1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t4)*cos(t5
)*cos(t6)*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(pow(cos(t4),2.0)*cos(
t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+t3v*(((pow(cos(
```

```
t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t
1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t4)*cos(t5
)*cos(t6)*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(pow(cos(t4),2.0)*cos(
t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-t6v*(((-
pow(cos(t4),2.0)*sin(t6)+pow(cos(t5),2.0)*sin(t6)+cos(t4)*cos(t5)*cos(t6)*sin(t4))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),
2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t4)*cos(t5
)*sin(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(pow(cos(t4
),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
((pow(cos(t4),2.0)*cos(t6)-pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1))*1.0/pow(pow(cos(t4),2.0
)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0))/sin(t5)+cos(t4)
*cos(t5)*cos(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1))*1.0/p
ow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0))+t4v*(((-
pow(cos(t4),2.0)*cos(t5)*sin(t6)+cos(t5)*pow(sin(t4),2.0)*sin(t6)+cos(t4)*cos(t6)*sin(t4)*2.0)*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),
2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
((pow(cos(t4),2.0)*cos(t6)-pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1))*1.0/pow(pow(cos(t4),2.0
)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0))/sin(t5)-
(cos(t5)*cos(t6)*sin(t4)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1
)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+cos(t4)*cos(t5)*
cos(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1))*1.0/pow(pow(co
s(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0));
 801      da2.entry(2,0) =
t5v*((sin(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(pow(cos(t5),2.0
)*sin(t4)*1.0/pow(sin(t5),2.0)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
(cos(t5)*sin(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t5)*sin(t6)+R.entr
y(1,0)*sin(t2+t3)*cos(t5)*sin(t1)*sin(t6)+R.entry(2,0)*sin(t2+t3)*cos(t4)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*sin(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(cos(t5)*sin(t4)*(cos(t4)*sin(t4)*sin(t5)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.
0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(R.entry(2,0
)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
```

```
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/sin(t5))-
t4v*((cos(t4)*cos(t5)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(cos(t5)*sin(t4)*(-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t6)*sin(t1)-
R.entry(2,0)*sin(t2+t3)*cos(t5)*sin(t4)*sin(t6)+R.entry(1,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*sin(t6)+R.entry(
0,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t5)*sin(t4
)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/sin(t5))-t6v*((cos(t5)*sin(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t5)-
R.entry(2,0)*sin(t2+t3)*sin(t4)*sin(t6)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*sin(t4)*sin(t6)+R.entry(0,0)*sin
(t2+t3)*cos(t1)*cos(t6)*sin(t5)+R.entry(1,0)*cos(t2+t3)*sin(t1)*sin(t4)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*
sin(t1)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)+R.entry(1,0)*cos(t2+t3)*cos(t4)*cos(t5)*
cos(t6)*sin(t1)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(cos(t5)*sin(t4)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0)*(R.entry(2,0)*si
n(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/sin(t5))-(t2v*cos(t5)*sin(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*sin(t5)*sin(t6)-
R.entry(2,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t6)*sin(t4)+R.entry(0,0)*cos
(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*sin(t1)*
sin(t5)*sin(t6)-R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(t3v*cos(t5)*sin(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t4)-R.entry(2,0)*sin(t2+t3)*sin(t5)*sin(t6)-
R.entry(2,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t6)*sin(t4)+R.entry(0,0)*cos
(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*sin(t1)*
sin(t5)*sin(t6)-R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(t1v*cos(t5)*si
n(t4)*(R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(0,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*sin(t1)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6))
)/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)));
 802       da2.entry(2,1) =
t5v*((cos(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t5)*sin(t6)+R.entry(1
,0)*sin(t2+t3)*cos(t5)*sin(t1)*sin(t6)+R.entry(2,0)*sin(t2+t3)*cos(t4)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t4)*sin(t1)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*sin(t5)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+cos(t4)*(cos(t4)
*sin(t4)*sin(t5)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.
0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(R.entry(2,0
)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
```

```
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))+t6v*((cos(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t5)-
R.entry(2,0)*sin(t2+t3)*sin(t4)*sin(t6)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*sin(t4)*sin(t6)+R.entry(0,0)*sin
(t2+t3)*cos(t1)*cos(t6)*sin(t5)+R.entry(1,0)*cos(t2+t3)*sin(t1)*sin(t4)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*
sin(t1)*sin(t5)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*cos(t6)+R.entry(1,0)*cos(t2+t3)*cos(t4)*cos(t5)*
cos(t6)*sin(t1)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
cos(t4)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0)*(R.entry(2,0)*si
n(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))-
t4v*((sin(t4)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(cos(t4)*(-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t6)*sin(t1)-
R.entry(2,0)*sin(t2+t3)*cos(t5)*sin(t4)*sin(t6)+R.entry(1,0)*cos(t2+t3)*cos(t5)*sin(t1)*sin(t4)*sin(t6)+R.entry(
0,0)*cos(t2+t3)*cos(t1)*cos(t5)*sin(t4)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
cos(t4)*1.0/pow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0)*(R.entry(2,0)*sin(t2+t3)*cos(t6)*sin(t4)+R.entry(2,0)*cos(t2+t3)*sin(t5)*sin(t6)-
R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(1,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin
(t2+t3)*sin(t1)*sin(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(1,0)*cos(t2+t3)*
cos(t4)*cos(t5)*sin(t1)*sin(t6)))+(t2v*cos(t4)*(R.entry(2,0)*cos(t2+t3)*cos(t6)*sin(t4)-
R.entry(2,0)*sin(t2+t3)*sin(t5)*sin(t6)-
R.entry(2,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t6)*sin(t4)+R.entry(0,0)*cos
(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*sin(t1)*
sin(t5)*sin(t6)-R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(t3v*cos(t4)*(R.
entry(2,0)*cos(t2+t3)*cos(t6)*sin(t4)-R.entry(2,0)*sin(t2+t3)*sin(t5)*sin(t6)-
R.entry(2,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t6)*sin(t4)+R.entry(0,0)*cos
(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(1,0)*sin(t2+t3)*cos(t6)*sin(t1)*sin(t4)+R.entry(1,0)*cos(t2+t3)*sin(t1)*
sin(t5)*sin(t6)-R.entry(1,0)*sin(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6)-
R.entry(0,0)*sin(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
(t1v*cos(t4)*(R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t6)*sin(t4)-R.entry(0,0)*cos(t2+t3)*cos(t6)*sin(t1)*sin(t4)-
R.entry(1,0)*sin(t2+t3)*cos(t1)*sin(t5)*sin(t6)+R.entry(0,0)*sin(t2+t3)*sin(t1)*sin(t5)*sin(t6)-
R.entry(1,0)*cos(t2+t3)*cos(t1)*cos(t4)*cos(t5)*sin(t6)+R.entry(0,0)*cos(t2+t3)*cos(t4)*cos(t5)*sin(t1)*sin(t6))
)/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6));
 803       da2.entry(2,2) = -t5v*(((cos(t5)*cos(t6)*sin(t5)*2.0-cos(t4)*sin(t4)*sin(t5)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),
2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+((pow(cos(t4),2
.0)*cos(t6)-pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1))*(cos(t4)*sin(t4)*sin(t5
)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.
0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0))/sin(t5)-
(cos(t5)*1.0/pow(sin(t5),2.0)*(pow(cos(t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(pow(cos(t4),2.0)*cos(
t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+(cos(t4)*cos(t6)
*sin(t5)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(pow(cos(t4)
```

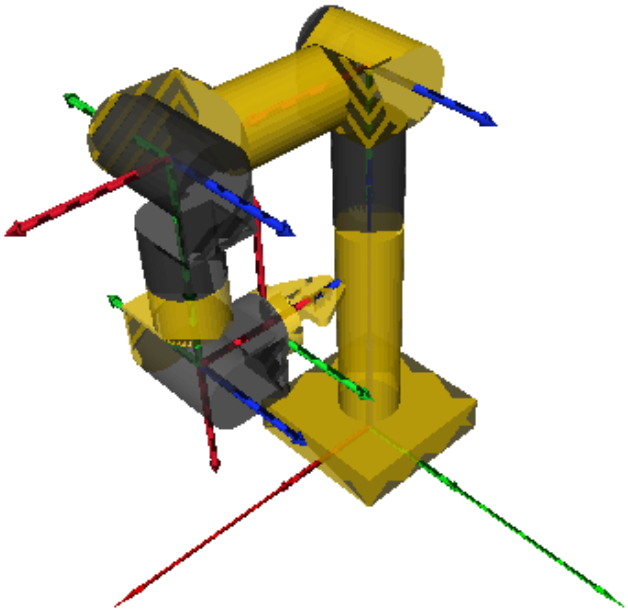
```
,2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
cos(t4)*cos(t5)*cos(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1)
)*(cos(t4)*sin(t4)*sin(t5)-cos(t4)*pow(cos(t6),2.0)*sin(t4)*sin(t5)*2.0-
cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0+pow(cos(t4),2.0)*cos(t5)*cos(t6)*sin(t5)*sin(t6)*2.0)*1.0/pow(pow(cos(t4),2.
0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0))-
t1v*(((R.entry(1,0)*cos(t2+t3)*cos(t1)-R.entry(0,0)*cos(t2+t3)*sin(t1))*(pow(cos(t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
(cos(t4)*cos(t5)*cos(t6)*(R.entry(1,0)*sin(t2+t3)*cos(t1)-
R.entry(0,0)*sin(t2+t3)*sin(t1)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+t2v*(((pow(cos(
t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t
1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t4)*cos(t5
)*cos(t6)*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(pow(cos(t4),2.0)*cos(
t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+t3v*(((pow(cos(
t4),2.0)*cos(t6)-
pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t
1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),2.0)*cos(t6)*sin(t6)-
pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t4)*cos(t5
)*cos(t6)*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(pow(cos(t4),2.0)*cos(
t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-t6v*(((-
pow(cos(t4),2.0)*sin(t6)+pow(cos(t5),2.0)*sin(t6)+cos(t4)*cos(t5)*cos(t6)*sin(t4))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),
2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))+(cos(t4)*cos(t5
)*sin(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1)))/(pow(cos(t4
),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))-
((pow(cos(t4),2.0)*cos(t6)-pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1))*1.0/pow(pow(cos(t4),2.0
)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-
pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0))/sin(t5)+cos(t4)
*cos(t5)*cos(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1))*1.0/p
ow(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(pow(cos(t4)
,2.0)*pow(cos(t6),2.0)-pow(cos(t5),2.0)*pow(cos(t6),2.0)-
pow(cos(t4),2.0)*pow(sin(t6),2.0)+pow(cos(t5),2.0)*pow(sin(t6),2.0)+pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(cos(t6
),2.0)-pow(cos(t4),2.0)*pow(cos(t5),2.0)*pow(sin(t6),2.0)+cos(t4)*cos(t5)*cos(t6)*sin(t4)*sin(t6)*4.0))+t4v*(((-
pow(cos(t4),2.0)*cos(t5)*sin(t6)+cos(t5)*pow(sin(t4),2.0)*sin(t6)+cos(t4)*cos(t6)*sin(t4)*2.0)*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1)))/(sin(t5)*(pow(cos(t4),
2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6)))-
((pow(cos(t4),2.0)*cos(t6)-pow(cos(t5),2.0)*cos(t6)+cos(t4)*cos(t5)*sin(t4)*sin(t6))*(-
R.entry(2,0)*sin(t2+t3)+R.entry(0,0)*cos(t2+t3)*cos(t1)+R.entry(1,0)*cos(t2+t3)*sin(t1))*1.0/pow(pow(cos(t4),2.0
)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0))/sin(t5)-
(cos(t5)*cos(t6)*sin(t4)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1
)))/(pow(cos(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6))+cos(t4)*cos(t5)*
cos(t6)*(R.entry(2,0)*cos(t2+t3)+R.entry(0,0)*sin(t2+t3)*cos(t1)+R.entry(1,0)*sin(t2+t3)*sin(t1))*1.0/pow(pow(co
s(t4),2.0)*cos(t6)*sin(t6)-pow(cos(t5),2.0)*cos(t6)*sin(t6)+cos(t4)*cos(t5)*sin(t4)-
cos(t4)*cos(t5)*pow(cos(t6),2.0)*sin(t4)*2.0+pow(cos(t4),2.0)*pow(cos(t5),2.0)*cos(t6)*sin(t6),2.0)*(-
pow(cos(t4),2.0)*cos(t5)+cos(t5)*pow(sin(t4),2.0)+pow(cos(t4),2.0)*cos(t5)*pow(cos(t6),2.0)*2.0-
cos(t5)*pow(cos(t6),2.0)*pow(sin(t4),2.0)*2.0+cos(t4)*cos(t6)*sin(t4)*sin(t6)*2.0+cos(t4)*pow(cos(t5),2.0)*cos(t
6)*sin(t4)*sin(t6)*2.0));
805       }
 807       return da2;
 808   }
```

6. **Realizar una visualización tridimensional en tiempo real en C++ y la API OpenGL. Posción de Home**

```
"C:\Users\User Admi\Desktop\Robot Posiciones - c
------------------------------------
============================
||      Posicion rO6      ||
============================
192
-1.38533e-005
321
============================
||   Angulos con C.I.     ||
============================
theta1 : 0
theta2 : 0
theta3 : 0
theta4 : -3.14159
theta5 : 4.71239
theta6 : 3.14159
================================
||    Velocidades Angulares     ||
================================
ohmega1 : 0
ohmega2 : 0
ohmega3 : 0
ohmega4 : 0
ohmega5 : 0
ohmega6 : 0
================================
||   Aceleraciones Angulares     ||
================================
alpha1 : 0
alpha2 : 0
alpha3 : 0
alpha4 : 0
alpha5 : 0
alpha6 : 0
```

TECLA G

```
----------------------------
============================
||    Posicion Guardada   ||
============================
----------------------------
```

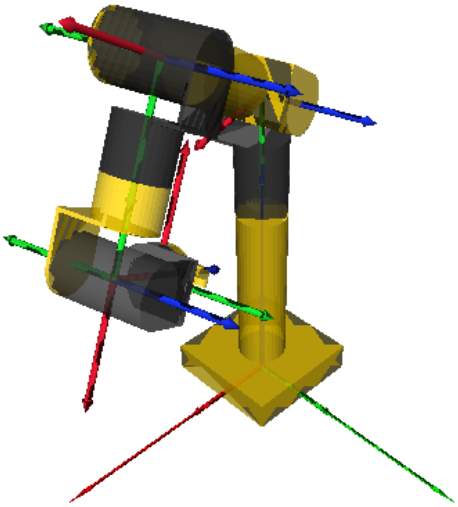


Movimiento de sus articulaciones

Articulación	Tecla
1	“ 1 ” , “ 2 ”
2	“ 3 ” , “ 4 ”
3	“ 5 ” , “ 6 ”
4	“ 7 ” , “ 8 ”
5	“ 9 ” , “0 ”
6	“ O ” , “ P ”

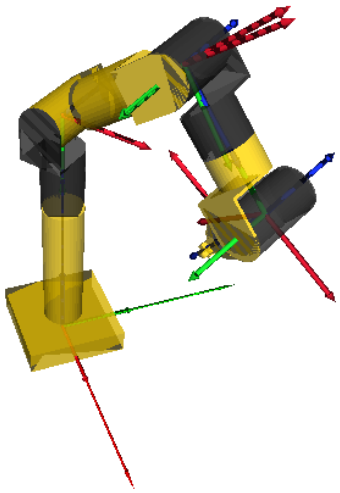
TECLA G

```
----------------------------
============================
||    Posicion Guardada   ||
============================
----------------------------
```



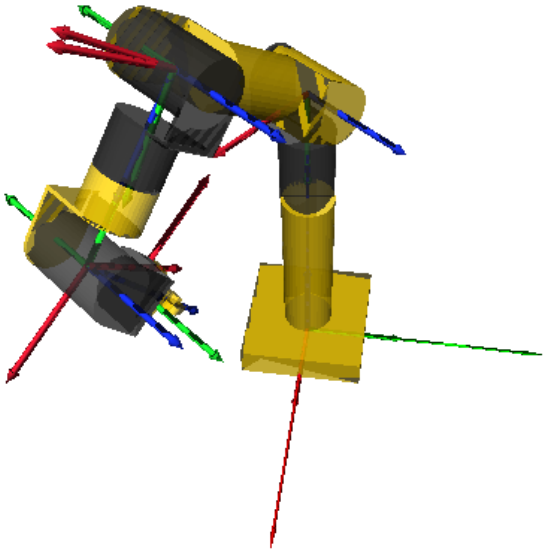
TECLA G

```
----------------------------
============================
||    Posicion Guardada   ||
============================
----------------------------
```



TECLA G

```
----------------------------
============================
||    Posicion Guardada   ||
============================
----------------------------
```



TECLA T

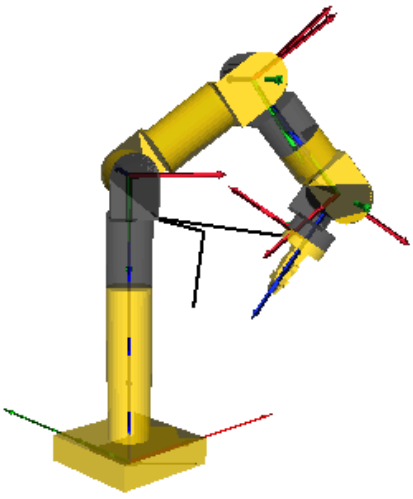
```
==================================
||      Trayectoria a seguir    ||
==================================
[seg: 0] a      (192,-1.38533e-005,321)->
[seg: 3] a      (328.572,289.095,370.74)->
[seg: 3] a      (375.727,-224.42,370.74)->
[seg: 3] a      (202.327,-224.442,304.717)

Tiempo estimado de trayectoria [seg]: 9
```

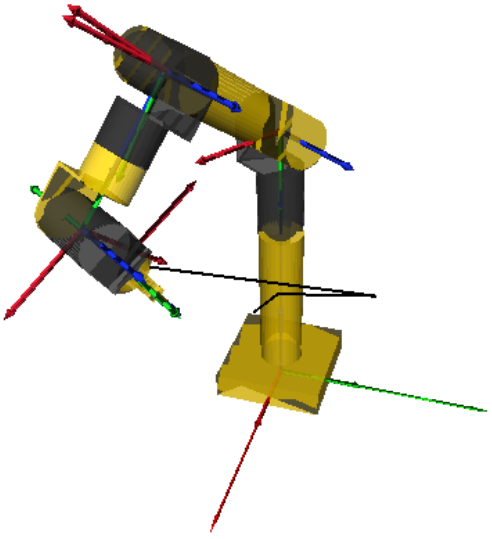
TECLA M

Coeficientes para la trayectoria cubica de (304.346,84.2302,464.271)->(286.426,250.26,481.946) y el analisis de sus velocidades angulares y aceleraciones angulares en un punto aleatorio del recorrido.

```
(304.346,84.2302,464.271)->(286.426,250.26,481.946)
==================================
||  Los coeficientes para X son  ||
==================================


1.32738

-5.97319

0

304.346

==================================
||  Los coeficientes para Y son  ||
==================================


-12.2985

55.3432

0

84.2302

==================================
||  Los coeficientes para Z son  ||
==================================


-1.30927

5.89172

0

464.271
```

```
============================
||      Posicion rO6      ||
============================
304.346
84.2302
464.271
============================
||   Angulos con C.I.     ||
============================
theta1 : 0.138796
theta2 : -0.739441
theta3 : 0.266181
theta4 : -2.65506
theta5 : 4.09977
theta6 : 3.5348
==================================
||    Velocidades Angulares     ||
==================================
ohmega1 : 9.13558e-008
ohmega2 : -2.14575e-007
ohmega3 : -1.7919e-007
ohmega4 : -7.39528e-008
ohmega5 : 3.28603e-007
ohmega6 : 2.70023e-007
==================================
||   Aceleraciones Angulares     ||
==================================
alpha1 : -0.127725
alpha2 : 0.164686
alpha3 : 0.280226
alpha4 : 0.0686877
alpha5 : -0.366065
alpha6 : -0.317164
```





TECLA M

Coeficientes para la trayectoria cubica de (286.426,250.26,481.946)->(315.029,-205.156,479.092) y un punto instantáneo en el corrimiento, con sus velocidades y aceleraciones angulares

```
(286.426,250.26,481.946)->(315.029,-205.156,479.092)
==================================
||  Los coeficientes para X son  ||
==================================


-2.11875

9.53436

0

286.426

==================================
||  Los coeficientes para Y son  ||
==================================


33.7345

-151.805

0

250.26

==================================
||  Los coeficientes para Z son  ||
==================================


0.211434

-0.951457

0

481.946
```

```
============================
||      Posicion rO6      ||
============================
286.426
250.26
481.946
============================
||   Angulos con C.I.     ||
============================
theta1 : 0.573457
theta2 : -0.716172
theta3 : 0.0919845
theta4 : -2.47246
theta5 : 4.10024
theta6 : 3.86484
==================================
||    Velocidades Angulares     ||
==================================
ohmega1 : 1.518e-007
ohmega2 : 1.36205e-007
ohmega3 : -1.56715e-007
ohmega4 : 6.53857e-008
ohmega5 : -3.8945e-008
ohmega6 : 1.00571e-007
==================================
||   Aceleraciones Angulares     ||
==================================
alpha1 : -0.250931
alpha2 : -0.0658449
alpha3 : 0.222744
alpha4 : -0.0545176
alpha5 : -0.0320944
alpha6 : -0.25947
```